EGT3
ENGINEERING TRIPOS PART IIB

Wednesday 20 April 2016 9.30 to 11

Module 4F7

DIGITAL FILTERS AND SPECTRUM ESTIMATION - WORKED SOLUTIONS

Answer not more than ***three*** *questions.*

All questions carry the same number of marks.

The ***approximate*** *percentage of marks allocated to each part of a question is indicated in the right margin.*

Write your candidate number ***not*** *your name on the cover sheet.*

STATIONERY REQUIREMENTS
Single-sided script paper

SPECIAL REQUIREMENTS TO BE SUPPLIED FOR THIS EXAM
CUED approved calculator allowed

10 minutes reading time is allowed for this paper.

You may not start to read the questions printed on the subsequent pages of this question paper until instructed to do so.

1 Examiner's comment: A popular and straightforward question, well-answered by most candidates.

(a) Describe briefly the principles behind the *nonparametric* power spectral estimation method. Your discussion should include the correlogram, periodogram and possible improvement strategies.

[30%]

Solution:

[This is a little more detailed than required in the exam]

- The basic principle is generally to estimate the autocorrelation function R_{XX} and then take Fourier transforms - **Correlogram and Periodogram** methods.
- Further improvements can be made if we perform various types of *smoothing* or *averaging* - **Bartlett, Blackman-Tukey, Welch** methods

Correlogram and Periodogram Estimates

- These classical techniques are based on the principle of obtaining estimates of the auto-correlation function R_{XX} of the random process and then taking the Discrete time Fourier transform:

$$S_X(e^{j\omega T}) = \sum_{k=-\infty}^{\infty} R_{XX}[k]\, e^{-jk\omega T}$$

- If the process is WSS and ergodic, we can estimate R_{XX} assuming a correlation ergodic signal:

$$R_{XX}[k] \approx \frac{1}{2N+1} \sum_{-N}^{+N} x_n x_{n+k}$$

- There are several ways to proceed when the number of data points is finite; we consider the consequences of two of these.

Assume that N data points are available from a single sample function from a WSS process; then two possible estimates of the autocorrelation function are:

(i) Sample autocorrelation function (biased estimate):

$$\boxed{\hat{R}_{XX}[k] = \frac{1}{N}\sum_{n=0}^{N-1-k} x_n x_{n+k} \qquad 0 \le k < N} \tag{1}$$

(ii) Sample autocorrelation function (unbiased estimate):

$$\boxed{\hat{R}_{XX}[k] = \frac{1}{N-k}\sum_{n=0}^{N-1-k} x_n x_{n+k} \qquad 0 \le k < N} \tag{2}$$

•Intuitively, 1. is biased since we divide the summation by N rather than $N-k$, the number of terms in the summation.

•Note that the form of the upper limit ensures that only samples $x_n, \quad 0 \le n \le N-1$ appear in the summations.

•Note that the autocorrelation is an even function so that estimates for negative k are given by:

$$\hat{R}_{XX}[-k] = \hat{R}_{XX}[k]$$

•In fact the biased form has better properties (see later) and is generally used for spectrum estimation.

•Now, assume that $R_{XX}[k] = 0$ for $|k| > L$, where L is some chosen constant, typically with $L << N$.

•The **Correlogram** estimate for the power spectrum is obtained by taking the DTFT of the sample autocorrelation function, $\hat{R}_{XX}[k]$:

$$\boxed{\hat{S}_X(e^{j\omega T}) = \sum_{k=-L}^{L} \hat{R}_{XX}[k]\, e^{-jk\omega T}, \qquad L < N}$$

•Typically used with $L << N$.

•However, if the maximum correlation lag is taken to be:

$$L = N - 1$$

then the resulting estimate is:

$$\hat{S}_X(e^{j\omega T}) = \sum_{k=-(N-1)}^{N-1} \hat{R}_{XX}[k]\, e^{-jk\omega T} \tag{3}$$

•When the biased form (2.) is used for $\hat{R}_{XX}$, this can be rewritten in terms of the DTFT of $\{x_0, x_1, ..., x_{N-1}\}$:

$$\boxed{\begin{aligned} \hat{S}_X(e^{j\omega T}) &= \frac{1}{N}|X_w(e^{j\omega T})|^2 \\ X_w(e^{j\omega T}) &= \sum_{n=0}^{N-1} x_n\, e^{-jn\omega T} \end{aligned}} \tag{4}$$

which is known as the **Periodogram**.

Improving the Spectral Estimate

•The periodogram is a useful tool, but its variability is very high.

•We will consider several common methods to improve the performance, based on averaging, smoothing and windowing.

The Bartlett Procedure

•Earlier in this section it was observed that

$$S_X(\omega) = \lim_{D\to\infty} \frac{1}{2D} E\{|X_D(\omega)|^2\}$$

•It would seem natural to try and improve the spectrum estimate by performing some averaging in order to mimic the ensemble average above.

•Let the data sequence x_n be of length $N_s = KN$ and segment this sequence into K subsequences of length N:

$$x_n^{(k)} = x_{n+kN} \qquad 0 \le n \le N-1 \qquad 0 \le k \le K-1$$

•Calculate the periodogram for each frame, denoted by $\hat{S}_X^{(k)}(e^{j\omega T})$, $k = 0, 1, 2, ..., K-1$.

•The Bartlett estimate is then given by:

$$\boxed{\hat{S}_X^B(e^{j\omega T}) = \frac{1}{K} \sum_{k=0}^{K-1} \hat{S}_X^{(k)}(e^{j\omega T})} \tag{5}$$

 (cont.

- If the data subsequences are uncorrelated with one another the Bartlett procedure reduces the variance by a factor of K, by less if they are correlated.
- Bartlett allows a trade-off between frequency resolution ($\propto N$) and variance of the estimate ($\propto 1/K$).
- Reduction in variance is at the expense of requiring more data for the same resolution.

The Blackman-Tukey Procedure

- The Blackman-Tukey method applies a window function of length $2L+1$ to the estimated autocorrelation function:

$$\boxed{\hat{S}_X^{BT}(e^{j\omega T}) = \sum_{-L}^{L} w_l \hat{R}_{XX}[l] \exp(-j\omega T)} \tag{6}$$

  where $L < N$ and w_l is any suitable window function, e.g. Hamming, Hanning, Bartlett,...
- We have already analysed a similar case, see page 67. It is clear that the resulting spectrum can be written as a frequency domain convolution:

$$\hat{S}_X^{BT}(e^{j\omega T}) = \frac{1}{2\pi} W(e^{j\omega t}) * \hat{S}_X(e^{j\omega t})$$

  where $W(.)$ is the DTFT of the window function and $\hat{S}_X(.)$ is the Periodogram.
- The B-T method can reduce the variance of the periodogram estimate at the expense of some frequency resolution. A special case is the correlogram considered earlier

The Welch Procedure

- The Welch procedure performs averaging over frames as in the Bartlett method
- However, the periodograms are *modified* to incorporate a window function on the data:

$$\boxed{\hat{S}'^{(k)}(e^{j\omega T}) = \frac{1}{N} \left| \sum_{n=0}^{N-1} w_n x_n^{(k)} e^{-j\omega nT} \right|^2}$$

  with $1/N \sum_{n=0}^{N-1} w_n^2 = 1$.
- As for the Bartlett method, averaging is then performed over K frames:

$$\boxed{\hat{S}_X^{W}(e^{j\omega T}) = \frac{1}{K} \sum_{k=0}^{K-1} \hat{S}_X'^{(k)}(e^{j\omega T})} \tag{7}$$

•The expected value of this spectral estimate can be shown to be:

$$E[\hat{S}_X^W(e^{j\omega T})] = \frac{1}{2\pi}V(e^{j\omega T}) * S_X(e^{j\omega T})$$

where $W(e^{j\omega T})$ is the DTFT of the window and $V(e^{j\omega T}) = \frac{1}{N}|W(e^{j\omega T})|^2$.

•When the segments are non-overlapping the variance is approximately that of the Bartlett estimate.

(b) It is proposed to estimate the power spectrum of a wide-sense stationary random process by first multiplying the data x_n with a window function w_n having length N, i.e. $w_n = 0$ for $n < 0$ and $n > N-1$, so that

$$x_n^w = w_n x_n\,.$$

The autocorrelation function is then estimated as

$$\hat{R}_{XX}[|k|] = \begin{cases} \frac{1}{N}\sum_{n=0}^{N-1} x_n^w x_{n+|k|}^w, & k = -N+1, \ldots -1, 0, 1, \ldots, N-1\,, \\ 0, & \text{otherwise.} \end{cases}$$

(i) Show that the expected value of the autocorrelation function estimate is given by

$$E[\hat{R}_{XX}[|k|]] = R_{XX}[k]\frac{1}{N}\sum_{n=0}^{N-1-|k|} w_n w_{n+|k|}$$

where $R_{XX}[k]$ is the true autocorrelation function for the process, and hence and hence explain whether this estimator is biased or not.. [30%]

Solution: For positive k:

$$\begin{aligned}
E[\hat{R}_{XX}[k]] &= E[\frac{1}{N}\sum_{n=0}^{N-1} x_n^w x_{n+k}^w] \\
&= E[\frac{1}{N}\sum_{n=0}^{N-1-k} (w_n x_n)(w_{n+k} x_{n+k})] \\
&= \frac{1}{N}\sum_{n=0}^{N-1-k} w_n w_{n+k} E[x_n x_{n+k}] \\
&= \frac{1}{N}\sum_{n=0}^{N-1-k} w_n w_{n+k} R_{XX}[k] \\
&= R_{XX}[k]\frac{1}{N}\sum_{n=0}^{N-1-k} (w_n w_{n+k})
\end{aligned}$$

Then obtain an expression for negative k by substituting $|k|$ for k.

It is biased in general for $k < N$, since the window correlation summation will not be constant with k, and of course biased for larger k, since we set those estimates to zero.

(ii) The power spectrum estimate $\hat{S}_X(e^{j\theta})$ is obtained by taking the DTFT of the estimated autocorrelation function $\hat{R}_{XX}[k]$.
Show that the expected value of the corresponding power spectrum estimate is:

$$E[\hat{S}_X(e^{j\theta})] = \frac{1}{2\pi N} S_X(e^{j\theta}) * |W(e^{j\theta})|^2$$

where $S_X(e^{j\theta})$ is the true power spectrum of the random process, $W(e^{j\theta})$ is the DTFT of the window function w_n, and $*$ denotes the convolution operator. [25%]

Solution:
We have from lectures on the periodogram that

$$E[\hat{S}_X(e^{j\omega T})] = E[DTFT\{\hat{R}_{XX}[k]\}] = DTFT\{E[\hat{R}_{XX}[k]]\}$$

Then, note that

$$\sum_{n=0}^{N-1-k} (w_n w_{n+k}) = \{w_n\} * \{w_{-n}\}$$

whose DTFT is:

$$W(e^{j\omega T}) W^*(e^{j\omega T}) = |W(e^{j\omega T})|^2$$

But, this term is multiplied (in time) with $R_{XX}[k]$. Hence overall the DTFT is:

$$E[\hat{S}_X(e^{j\omega T})] = \frac{1}{2\pi N} S_X(e^{j\omega T}) * |W(e^{j\omega T})|^2,$$

as required

(iii) Explain the advantages and disadvatages of this method for power spectral estimation in comparison with the standard periodogram estimator. [15%]

Solution:
The method convolves the periodogram with the window function magnitude squared. With appropriate choice of window this will smooth out the randomness in the periodogram without losing too much spectral detail. The estimate is guaranteed positive-valued, which is good, though of course there is a trade-off in some loss of spectral detail through the convolution.

2 Examiner's comment: Attempted by all candidates with good results in general. Some confusion about how to handle the less standard part (b) for some candidates. Many expressed h0 and h1 in terms of the autocorrelation function, which gained some, but not full, credit. This route sometimes led to a correct answer to part (ii).

(a) Describe the autoregressive moving average (ARMA) class of signal model, explaining how to obtain the power spectrum of an ARMA process and any advantages of such an approach compared to nonparametric approaches. [25%]

Answer: Bookwork, taken from:

ARMA Models A quite general representation is the autoregressive moving-average (ARMA) model:

•The ARMA(P,Q) model difference equation representation is:

$$\boxed{x_n = -\sum_{p=1}^{P} a_p x_{n-p} + \sum_{q=0}^{Q} b_q w_{n-q}} \tag{8}$$

where:

a_p are the AR parameters,

b_q are the MA parameters

and $\{W_n\}$ is a zero-mean stationary white noise process with unit variance, $\sigma_w^2 = 1$.

•Clearly the ARMA model is a pole-zero IIR filter-based model with transfer function

$$H(z) = \frac{B(z)}{A(z)}$$

where:

$$A(z) = 1 + \sum_{p=1}^{P} a_p z^{-p}, \quad B(z) = \sum_{q=0}^{Q} b_q z^{-q}$$

•Unless otherwise stated we will always assume that the filter is stable, i.e. the poles (solutions of $A(z) = 0$) all lie *within* the unit circle (we say in this case that $A(z)$ is *minimum phase*). Otherwise the autocorrelation function is undefined and the process is technically *non-stationary*.

•Hence the power spectrum of the ARMA process is:

$$S_X(e^{j\omega T}) = \frac{|B(e^{j\omega T})|^2}{|A(e^{j\omega T})|^2}$$

Thus, estimate the parameters a and b from the data, then plug into spectral density formula.

The ARMA model is quite a flexible and general way to model a stationary random process:

- The poles model well the *peaks* in the spectrum (sharper peaks implies poles closer to the unit circle)
- The zeros model troughs in the spectrum
- Complex spectra can be approximated well by large model orders P and Q

Can give improved variance of estimation; however, may be highly biased and inaccurate when an ARMA model is inappropriate for the data. Also, quite expensive to compute parameters accurately.

(b) An ARMA(P,Q) model has the following digital filtering equation:

$$x_n = -\sum_{p=1}^{P} a_p x_{n-p} + \sum_{q=0}^{Q} b_q w_{n-q}.$$

where $\{w_n\}$ is zero mean white noise with unity variance, and the filter is assumed stable.

(i) Explain carefully why it is not necessary to include a variance parameter (not necessarily equal to unity) for the white noise process $\{w_n\}$ in the above ARMA formulation. [10%]

Answer:

This is not necessary, since any scaling of the noise process by a standard deviation parameter can be absorbed into the values of the b coefficients (not the as since we have to have 'a_0' equal to 1.)

(ii) Show that the ARMA model autocorrelation function obeys the following difference equation:

$$R_{XX}[r] + \sum_{p=1}^{P} a_p R_{XX}[r-p] = \sum_{q=0}^{Q} b_q h_{q-r}$$

where h_r is a particular function of the ARMA systems that should be carefully defined. Explain why the term $\sum_{q=0}^{Q} b_q h_{q-r}$ must always be zero for $r > Q$.

[30%]

Answer:

Autocorrelation function for ARMA Model The autocorrelation function $R_{XX}[r]$ for the output x_n of the ARMA model is:

$$R_{XX}[r] = E[x_n x_{n+r}]$$

Substituting for x_{n+r} from equation 8 gives:

$$R_{XX}[r] = E\left[x_n \left\{ -\sum_{p=1}^{P} a_p x_{n+r-p} + \sum_{q=0}^{Q} b_q w_{n+r-q} \right\}\right]$$

$$= -\sum_{p=1}^{P} a_p E[x_n x_{n+r-p}] + \sum_{q=0}^{Q} b_q E[x_n w_{n+r-q}]$$

The white noise process $\{W_n\}$ is wide-sense stationary so that $\{X_n\}$ is also wide-sense stationary provided the the ARMA filter is stable. Therefore:

$$\boxed{R_{XX}[r] = -\sum_{p=1}^{P} a_p R_{XX}[r-p] + \sum_{q=0}^{Q} b_q R_{XW}[r-q]} \tag{9}$$

Note that the auto-correlation and cross-correlation satisfy the same ARMA system difference equation as x_n and w_n.

The cross-correlation term $R_{XW}[.]$ can be obtained as follows. Let the system impulse response be h_n, then:

$$x_n = \sum_{m=\infty}^{\infty} h_m w_{n-m}$$

Therefore,

$$E[x_n w_{n+k}] = E[w_{n+k} \sum_{m=\infty}^{\infty} h_m w_{n-m}]$$

$$R_{XW}[k] = \sum_{m=-\infty}^{\infty} h_m E[w_{n+k} w_{n-m}]$$

Now the noise is a zero-mean stationary white process so that:

$$E[w_{n+k} w_{n-m}] = \begin{cases} \sigma_W^2 & \text{if } m = -k \\ 0 & \text{otherwise} \end{cases}$$

and $\sigma_W^2 = 1$ without loss of generality. Hence,

$$R_{XW}[k] = h_{-k}$$

Substituting this expression for $R_{XW}[k]$ into equation 9 gives the *Yule-Walker Equation* for an ARMA process,

$$\boxed{R_{XX}[r] = -\sum_{p=1}^{P} a_p R_{XX}[r-p] + \sum_{q=0}^{Q} b_q h_{q-r}} \tag{10}$$

Since the system is causal, equation 10 may be rewritten as:

$$\boxed{R_{XX}[r] = -\sum_{p=1}^{P} a_p R_{XX}[r-p] + c_r} \tag{11}$$

where:

$$c_r = \begin{cases} \sum_{q=r}^{Q} b_q h_{q-r} & \text{if } r \leq Q \\ 0 & \text{if } r > Q \end{cases} \tag{12}$$

(c) An ARMA(1,1) model is to be estimated from autocorrelation data.

(i) Express the first two terms h_0 and h_1 from the ARMA(1,1) model in terms of the coeefficients $\{a_p\}$ and $\{b_q\}$. [10%]

Answer:

h_n is the impulse response of the filter. Hence we may drive the filter directly with a digital impulse δ_n to determine h_0 and h_1:

$$h_0 = -a_1 h_{-1} + b_0\delta_0 + b_1\delta_{-1} = 0 + b_0 + 0 = b_0$$

since δ_{-1} and h_{-1} are zero (causal system).

$$h_1 = -a_1 h_0 + b_0\delta_1 + b_1\delta_0 = -a_1 b_0 + b_1$$

since $\delta_1 = 0$.

(ii) Some values of the autocorrelation function for an ARMA(1,1) process are given by

$$R_{XX}[0] = 1,\ R_{XX}[1] = -0.4,\ R_{XX}[2] = 0.2,\ R_{XX}[3] = -0.1\,.$$

Use the result of part (b)(ii) and your expressions for h_0 and h_1 to determine the coefficients of the corresponding ARMA(1,1) model. You are given that b_0 equals 2. [20%]

Answer:

Write out the autocorrelation equations for $r = 0, 1, 2, 3$:

$$R_{XX}[0] = -a_1 R_X X[-1] + c_0$$
$$R_{XX}[1] = -a_1 R_X X[0] + c_1$$
$$R_{XX}[2] = -a_1 R_X X[1]$$
$$R_{XX}[3] = -a_1 R_X X[2]$$

since c_2 and c_3 are zero. Solving the $r = 2$ or 3 case, we get:

$$a_1 = 0.5$$

Then, solving for b, we first calculate c_0 and c_1:

$$c_0 = b_0 h_0 + b_1 h_1 = b_0^2 + b_1(-a_1 b_0 + b_1)$$

$$c_1 = b_1 h_0 = b_0 b_1$$

But we know a_1, so

$$c_0 = 1 + 0.5 * -0.4 = 0.8$$

, since $R_{XX}[-1] = R_{XX}[1]$. and

$$c_1 = -0.4 + 0.5 * 1 = 0.1$$

Thus we have

$$b_0 b_1 = 0.1, \; b_0^2 + b_1(-a_1 b_0 + b_1) = 0.8$$

With the given $b_0 = 2$ we obtain from just the first expression that $b_1 = 0.05$.

3 Examiner's comment: Popular question, with high marks in general

(a) In the standard adaptive filtering problem we have an input signal $\{u(n)\}_{n=0}^{\infty}$, a reference signal $\{d(n)\}_{n=0}^{\infty}$, and a Finite Impulse Response (FIR) filter $\{h_m\}_{m=0}^{M-1}$ of length M. Describe the setup of the general adaptive adaptive filter, including the error criterion/cost function vector notation, and illustrate it by a simple block diagram. Also explain briefly the main conceptual difference between a Wiener filter and an adaptive filter implementation. [15%]

(b) Name the four basic classes of application within the framework of part (a) and describe any three of them with the aid of block diagrams. Give one practical example for each class of applications. [15%]

(c) One of the most popular adaptive filtering algorithms is the Least-Mean-Square (LMS) algorithm. Explain the main ideas behind the LMS algorithm and give the coefficient update equation. How is it obtained from the cost function in part (a)? (No detailed derivation is required.) [10%]

(d) The Normalised LMS (NMLS) algorithm is closely related to the LMS algorithm.

(i) Give the coefficient update equation of the NLMS algorithm. What is the advantage of the NLMS algorithm over the LMS? [15%]

(ii) Describe how the NLMS coefficient update equation be interpreted as a projection mechanism. Give a geometrical illustration of this projection. [15%]

(e) The idea of interpreting the NLMS coefficient update as a projection operation (as in (d)(ii)) can be generalised to yield a whole class of improved adaptation algorithms.

(i) Explain how the NLMS projection idea can be generalised to improve performance. Give a graphical illustration of the projections involved. [15%]

(ii) Give the coefficient update equation for the resulting algorithm. [15%]

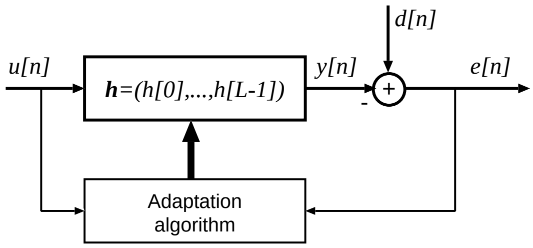
SOLUTION:

(a) Length-M FIR System:

$$e(n) = y(n) - d(n) = \sum_{m=0}^{M-1} h_m u(n-m) - d(n) = \mathbf{h}^T \mathbf{u}(n) - d(n),$$

where

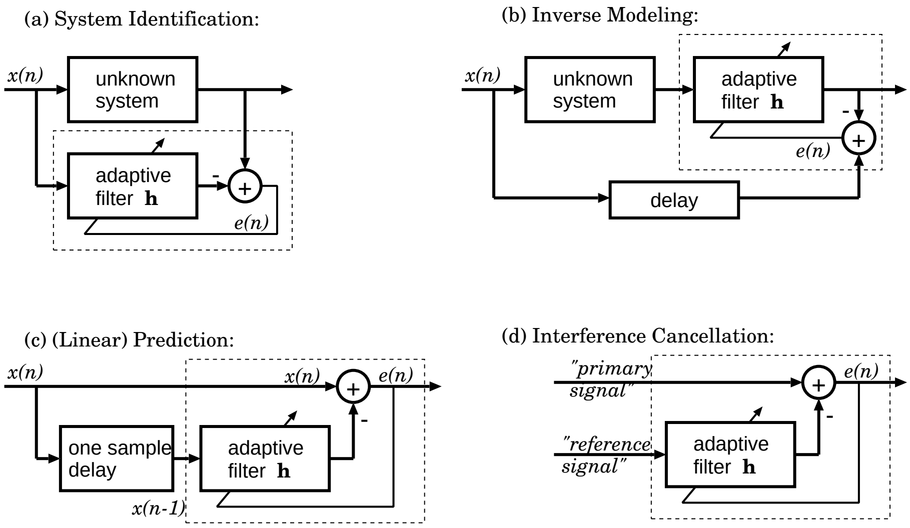
$$\begin{aligned}\mathbf{h} &= [h_0, h_1, \ldots, h_{M-1}]^T, \\ \mathbf{u}(n) &= [u(n), u(n-1), \ldots, u(n-M+1)]^T.\end{aligned}$$



Cost function to be minimized w.r.t. $\mathbf{h}$:
mean squared error $J(\mathbf{h}) = E\{e^2\}$.

- •Wiener filter: J is minimized under stationarity assumptions
- •Adaptive solution: nonstationary/time-varying environments are allowed.

(b) Four basic classes of adaptive filter applications.



Examples: (a) echo cancellation, (b) equalizer, dereverberation, (c) linear predictive coding for speech signals, (d) acoustic noise cancellation,

(c) LMS:

$$\mathbf{h}(n+1) = \mathbf{h}(n) + \mu \mathbf{u}(n) e(n).$$

The LMS update can be obtained by a stochastic approximation of the steepest descent algorithm, based on the cost function J (see above), i.e.,

$$\mathbf{h}(n+1) = \mathbf{h}(n) - \frac{\mu}{2} \nabla J(\mathbf{h}(n)).$$

 (cont.

In the stochastic approximation, the expectation in the update is replaced by the instantaneous value.

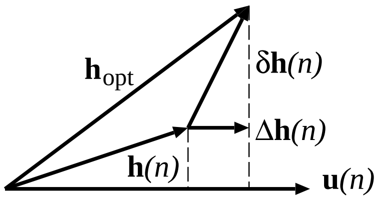
(d) (i) NLMS:

$$\mathbf{h}(n+1) = \mathbf{h}(n) + \mu \frac{\mathbf{u}(n)}{\|\mathbf{u}\|^2 + \delta} e(n).$$

In the original LMS algorithm, the limits of the stepsize μ for stable convergence depend on the input signal power. Specifically, $0 \le \mu < \frac{2}{ME\{u^2(n)\}}$ for LMS. The normalization in NLMS removes this power dependence of the stepsize, i.e., $0 \le \mu < 2$, making it more suitable in many applications with nonstationary signals.

(ii) Current misalignment: $\delta\mathbf{h}(n) = \mathbf{h}_{\text{opt}} - \mathbf{h}(n) \Rightarrow e(n) = \mathbf{u}^T(n)\delta\mathbf{h}(n)$
Current NLMS adjustment (for $\mu = 1$): $\Delta\mathbf{h}(n) = \mathbf{h}(n+1) - \mathbf{h}(n)$,

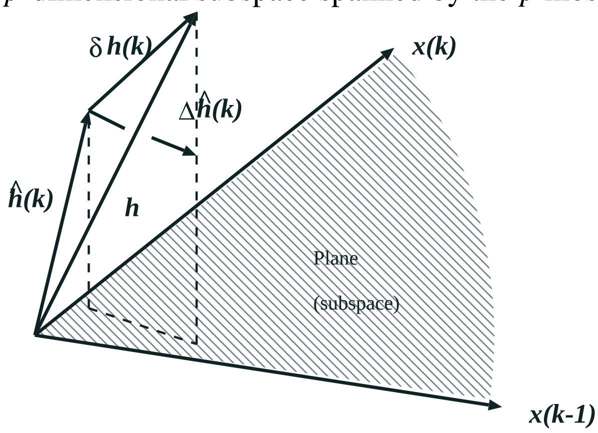
$$\Delta\mathbf{h}(n) = \frac{\mathbf{u}^T(n)\delta\mathbf{h}(n)}{\|\mathbf{u}(n)\|^2}\mathbf{u}(n)$$



The adjustment vector $\Delta\mathbf{h}(n)$ is the projection of the current misalignment $\delta\mathbf{h}(n)$ onto the current input signal vector $\mathbf{u}(n)$.

(e) Affine Projection Algorithm (APA):

(i) The update $\Delta\mathbf{h}(n)$ is the projection of the current misalignment $\delta\mathbf{h}(n)$ onto a p-dimensional subspace spanned by the p most recent input signal vectors.



(ii)

$$
\begin{aligned}
\mathbf{e}(n) &= \mathbf{y}(n) - \mathbf{X}^T(n)\hat{\mathbf{h}}(n) \\
\hat{\mathbf{h}}(n+1) &= \hat{\mathbf{h}}(n) + \mu \mathbf{X}(n)[\mathbf{X}^T(n)\mathbf{X}(n) + \delta \mathbf{I}]^{-1}\mathbf{e}(n),
\end{aligned}
$$

where $\mathbf{e}(k)$ is an error vector of order p,

$$\mathbf{e}(n) = [e_1(n), e_2(n), \ldots, e_p(n)]^T$$

and

$$\mathbf{y}(n) = [y(n), y(n-1), \ldots, y(n-p+1)]^T,$$

$$\mathbf{X}(n) = [\mathbf{u}(n), \mathbf{u}(n-1), \ldots, \mathbf{u}(n-p+1)].$$

The update for the new coefficient vector $\mathbf{h}(n+1)$ follows from the objective to cancel the error of the latest p time instances, i.e.,

$$
\begin{aligned}
\mathbf{u}^T(n)\hat{\mathbf{h}}(n+1) &= d(n) \\
\mathbf{u}^T(n-1)\hat{\mathbf{h}}(n+1) &= d(n-1) \\
&\vdots \\
\mathbf{u}^T(n-p+1)\hat{\mathbf{h}}(n+1) &= d(n-p+1).
\end{aligned}
$$

4 Examiner's comment: The least popular question, but again well handled by most.

Consider the following recursive algorithm:

$$\mathbf{h}(n) = \mathbf{h}(n-1) + \mu \tilde{\mathbf{R}}^{-1}(\mathbf{p} - \mathbf{R}\mathbf{h}(n-1)),$$

where $\mathbf{R}$ and $\mathbf{p}$ are a definite positive matrix (input correlation matrix) and a vector (crosscorrelation vector between input signal and reference) of appropriate dimensions, respectively. The definite positive matrix $\tilde{\mathbf{R}}$ is assumed to be an approximation or estimate of the true correlation matrix $\mathbf{R}$. Moreover, it is assumed that $\tilde{\mathbf{R}}$ can be expressed as $\tilde{\mathbf{R}} = \mathbf{Q}\tilde{\Lambda}\mathbf{Q}^T$, where the matrix $\mathbf{Q}$ is an orthonormal matrix and contains the eigenvectors of the original correlation matrix $\mathbf{R}$, and $\tilde{\Lambda}$ is approximated as a diagonal matrix.

(a) Assuming the coefficient vector $\mathbf{h}(n)$ of the algorithm converges towards a limit $\mathbf{h}_{\text{opt}}$, find an expression for $\mathbf{h}_{\text{opt}}$. [10%]

(b) (i) Based on the eigenvalue decomposition (modal decomposition) of $\mathbf{R}$ and the above expression decomposition of $\tilde{\mathbf{R}}$, obtain a recursion for the misalignment $\mathbf{h}(n) - \mathbf{h}_{\text{opt}}$ in the corresponding eigendomain, and find the limits for the choice of the stepsize μ ensuring convergence of the algorithm whatever initial vector $\mathbf{h}(0)$ is chosen. [30%]

(ii) Discuss the extreme cases $\tilde{\mathbf{R}} = \mathbf{R}$ and $\tilde{\mathbf{R}} = \mathbf{I}$. Distinguish in this discussion between the use of a common stepsize for all modes, and modal stepsizes in which different step sizes may be chosen for each mode. [15%]

(iii) In the case $\tilde{\mathbf{R}} = \mathbf{I}$ and a single stepsize for all modes, express the range of stepsize in terms of a signal variance rather than eigenvalues. [15%]

(c) In practical applications, the quantities $\mathbf{R}$ and $\mathbf{p}$ are typically not known in advance. Moreover, they can be time-varying.

(i) How can the above recursive algorithm be approximated to obtain practical algorithms such as the LMS algorithm? State the relation explicitly using equations. [10%]

(ii) How is the matrix $\tilde{\mathbf{R}}$ defined for the LMS algorithm? [10%]

(iii) How should the matrix $\tilde{\mathbf{R}}$ be defined in order to obtain an RLS-like algorithm? Note that in this case, it will be required to handle nonstationary environments. [10%]

SOLUTION:

We consider

$$\mathbf{h}(n) = \mathbf{h}(n-1) + \mu \tilde{\mathbf{R}}^{-1}(\mathbf{p} - \mathbf{R}\mathbf{h}(n-1)).$$

(a) Limit: $\mathbf{h}(n) = \mathbf{h}(n-1)$
$\Rightarrow \mathbf{p} - \mathbf{R}\mathbf{h}(n-1) = \mathbf{0}$
$\Rightarrow \mathbf{h}_{\text{opt}} = \mathbf{R}^{-1}\mathbf{p}$ (= Wiener solution)

(b) (i) Misalignment:

$$\begin{aligned}(\mathbf{h}(n) - \mathbf{h}_{\text{opt}}) &= (\mathbf{h}(n-1) - \mathbf{h}_{\text{opt}}) - \mu \tilde{\mathbf{R}}^{-1}\mathbf{R}(\mathbf{h}(n-1) - \mathbf{h}_{\text{opt}}) \\ &= (\mathbf{I} - \mu \tilde{\mathbf{R}}^{-1}\mathbf{R})(\mathbf{h}(n-1) - \mathbf{h}_{\text{opt}}).\end{aligned}$$

Let $\mathbf{R} = \mathbf{Q}\Lambda\mathbf{Q}^T$, $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$, $\tilde{\mathbf{R}} = \mathbf{Q}\tilde{\Lambda}\mathbf{Q}^T$, and
$\mathbf{v}(n) = \mathbf{Q}^T(\mathbf{h}(n) - \mathbf{h}_{\text{opt}})$.
For the misalignment we obtain:

$$\begin{aligned}\mathbf{v}(n) &= \mathbf{Q}^T(\mathbf{I} - \mu \tilde{\mathbf{R}}^{-1}\mathbf{R})(\mathbf{h}(n-1) - \mathbf{h}_{\text{opt}}) \\ &= (\mathbf{I} - \mu \tilde{\Lambda}^{-1}\Lambda)\mathbf{v}(n-1).\end{aligned}$$

The k-th component of this misalignment vector reads

$$v_k(n) = \left(1 - \mu \frac{\lambda_k}{\tilde{\lambda}_k}\right) v_k(n-1).$$

Condition for stability:

$$\left|1 - \mu \frac{\lambda_k}{\tilde{\lambda}_k}\right| < 1,$$

i.e.,

$$0 \le \mu \le 2\frac{\tilde{\lambda}_k}{\lambda_k}.$$

(ii) •Case $\tilde{\mathbf{R}} = \mathbf{R}$: We obtain $0 \le \mu \le 2$
(valid for all modes and, thus, also for the common step size)

•Case $\tilde{\mathbf{R}} = \mathbf{I}$: We obtain $0 \le \mu \le \frac{2}{\lambda_k}$.
In total, for common step size: $0 \le \mu \le \frac{2}{\lambda_{\max}}$,
where $\lambda_{\max}$ denotes the largest eigenvalue of $\mathbf{R}$.

(iii)

$$\begin{aligned}\lambda_{\max} < \sum_{k=1}^{M} \lambda_k &= \text{tr}\{\Lambda\} = \text{tr}\{\Lambda\mathbf{Q}^T\mathbf{Q}\} = \text{tr}\{\mathbf{Q}\Lambda\mathbf{Q}^T\} \\ &= \text{tr}\{\mathbf{R}\} = M \cdot E\{u^2(n)\}.\end{aligned}$$

 (cont.

Hence,

$$0 \leq \mu < \frac{2}{M \cdot E\{u^2(n)\}}.$$

(c) Stochastic approximation of the update:

$$\begin{aligned}
\mathbf{p} - \mathbf{R}\mathbf{h} &= E\{\mathbf{u}(n)d(n)\} - E\{\mathbf{u}(n)\mathbf{u}^T(n)\mathbf{h}\} \\
&= E\{\mathbf{u}(n)(d(n) - \mathbf{u}^T(n)\mathbf{h})\} \\
&= E\{\mathbf{u}(n)e(n)\} \\
&\approx \mathbf{u}(n)e(n).
\end{aligned}$$

Hence, in practice, $\mathbf{p} - \mathbf{R}\mathbf{h}$ is replaced by $\mathbf{u}(n)e(n)$.
LMS: $\tilde{\mathbf{R}} = \mathbf{I}$
RLS: $\tilde{\mathbf{R}}(n) = \lambda\tilde{\mathbf{R}}(n-1) + \mathbf{u}(n)\mathbf{u}^T(n)$, where λ denotes a forgetting factor ($0 < \lambda < 1$).
The forgetting factor allows us to handle nonstationary environments.

END OF PAPER

THIS PAGE IS BLANK