

**Review Problems for Exam 3**

1. **LMS Algorithm:** The convergence and stability of the LMS algorithm is intimately related to the eigenvalues of the input covariance matrix. As a rough rule of thumb, steepest descent and LMS are stable as long as the step size \( \mu < \frac{2}{\lambda_{\text{max}}} \), where \( \lambda_{\text{max}} \) is the largest eigenvalue of the covariance matrix.

Consider the following problem. We would like to use the LMS algorithm to adaptively tune a filter to match the acoustics of an auditorium. Specifically, the input to the algorithm is a sampled voice signal \( x[n] \) (recorded directly at the microphone) and \( y[n] \) is the sound measured in the middle of the auditorium. To account for echoes and noise in the auditorium, we assume that \( y[n] \) is related to \( x[n] \) by

\[
y[n] = h[0]x[n] + h[1]x[n - 1] + w[n],
\]

where \( \{h[0], h[1]\} \) is the (FIR) impulse response of the room and \( w[n] \) is an additive white noise. The voice signal \( x[n] \) is modeled as a random process described by the following (single-pole) difference equation:

\[
x[n] = \alpha x[n - 1] + u[n],
\]

where \( u[n] \) is a white noise process of power \( \sigma_u^2 \), and \( 0 < \alpha < 1 \) models the effect of the vocal tract shaping the noise to generate speech.

What is the largest allowable step-size for the LMS algorithm in this case? How does \( \alpha \) affect the step-size?
2. **Image Filtering:** Suppose we wish to convolve an \( N \times N \) image with an \( M \times M \) 2-d filter, \( M < N \).

(a) Approximately how many operations are required to directly compute the 2-d convolution?

(b) Suppose that we are going to use the 2-d FFT instead. Accounting for the necessary zero-padding to obtain the regular convolution from the 2-d circular convolution, approximately how many operations does the FFT-based approach require? Is the FFT-based computation more or less efficient than the direct convolution?

(c) Now assume that the filter has the special property of *separability*. That is, the 2-d filter is simply the convolution of two 1-d filters (one filter operating in the row direction and the other operating in the column direction). In this case, the 2-d convolution can be computed by first applying the row filter to each row of the image, then applying the column filter to each column of the row-filtering result. What is the computational complexity in this case?
3. Wiener Filtering:

Consider the simple problem of estimating a signal $s[n]$ observed in noise $w[n]$. That is, we observe $x[n] = s[n] + w[n]$ and want to design a filter $h[n]$ that minimizes the MSE $E[(s[n] - h[n] * x[n])^2]$. Recall that in this case the optimal Wiener filter is expressed in the frequency domain as

$$H(\omega) = \frac{S_{ss}(\omega)}{S_{ss}(\omega) + S_{ww}(\omega)},$$

where $S_{ss}(\omega)$ is the power spectral density of the signal $s[n]$ and $S_{ww}(\omega)$ is the power spectral density of the noise $w[n]$.

Derive an expression for the mean-square error of the optimal Wiener filter at each frequency. That is, determine $E[\|S(\omega) - H(\omega)X(\omega)\|^2]$, where $S(\omega)$ and $X(\omega)$ are the DTFTs of $s[n]$ and $x[n]$, respectively. **HINT:** If $y[n]$ is a stationary random process, then the power spectral density $S_{yy}(\omega) = E[Y(\omega)Y^*(\omega)]$. 
4. IIR Adaptive Filtering

In class we considered FIR adaptive filters of the form

\[ \hat{y}[n] = \sum_{k=0}^{p-1} w_n[k] x[n - k]. \]

IIR filter provide more flexibility and are preferable in cases where the adaptive filter requires a long memory (impulse response duration). Consider the IIR adaptive filter

\[ \hat{y}[n] = \sum_{k=1}^{q} v_n[m] y[n - m] + \sum_{k=0}^{p-1} w_n[k] x[n - k], \]

with adaptive weights \( \{v_n[m]\} \) and \( \{w_n[k]\} \).

a. Derive the LMS update equations for this filter.

b. Let \( q = 1 \) and \( p = 2 \). Assume that the input \( x[n] \) is Gaussian white noise with \( \sigma^2 = 1 \) and that \( E[y^2[n]] = 2 \). What step size range would you recommend for this filter?
5. Denoising — Warning: this one is pretty challenging (tougher than exam questions)

Let $\theta_0, \ldots, \theta_{N-1}$ denote the coefficients resulting from an orthogonal transform (e.g., DFT or DWT) of a finite duration input signal $s[n]$, $n = 0, \ldots, N - 1$. We observe a noisy version of the signal

$$x[n] = s[n] + w[n],$$

where $w[n], n = 0, \ldots, N - 1$, is a Gaussian white noise process with variance $\sigma^2 = 1$. The orthogonal transform coefficients of $x[n]$ are

$$\omega_i = \theta_i + w_i, \ i = 0, \ldots, N - 1,$$

where $w_i$ are the transform coefficients of the noise process $w[n]$.

Suppose the signal coefficients $\theta_i$ are either large in magnitude or zero. Specifically, each coefficient is independently and identically distributed according to

$$\theta_i = \begin{cases} +4 & \text{with probability } \frac{1}{4} \\ -4 & \text{with probability } \frac{1}{4} \\ 0 & \text{with probability } \frac{1}{2} \end{cases}$$

a. Determine $E[\theta_i^2]$?

b. In class we derived the optimal Wiener filter for each coefficient:

$$\hat{\theta}_i = \frac{E[\theta_i^2]}{E[\theta_i^2] + \sigma^2} \omega_i.$$

Show that the MSE $E[(\theta_i - \hat{\theta}_i)^2] \approx 0.89.$
c. Consider the thresholding estimator

\[ \tilde{\theta}_i = \begin{cases} 
\omega_i & \text{if } |\omega_i| > 2 \\
0 & \text{if } |\omega_i| \leq 2
\end{cases} \]

Note that with probability 1/2 we have \( \omega_i = \pm 4 + w_i \), and with probability 1/2 we have \( \omega_i = w_i \). In the first case \( \omega_i = \pm 2 + w_i \), it's easy to see (think about the Gaussian density function) that the probability that \( |\omega_i| > 2 \) is equal to 0.95. In the second case \( \omega_i = w_i \), the condition \( |\omega_i| > 2 \) is equivalent to the condition that the noise is outside of a \( \pm 2\sigma \) (two standard deviations) range, which happens with probability of approximately 0.05. (You may recall that a Gaussian random variable falls within two standard deviations about 95% of the time). Show that the MSE \( E[(\theta_i - \tilde{\theta}_i)^2] \approx 0.60 \).