EECS 844 – Spring 2026

Exam 4 Cover page*

Each student is expected to complete the exam individually using only course notes, the book, and technical literature, and without aid from outside sources.

Aside from the most general conversation of the exam material, I assert that I have neither provided help nor accepted help from another student in completing this exam. As such, the work herein is mine and mine alone.

____________________________________ __________

Signature Date

____________________________________ __________

Name (printed) Student ID #

*** Attach as cover page to completed exam.**

EECS 844 Exam 4
(Due Mon. May 15 by noon electronic only, no hardcopy)

Data sets can be found at http://www.ittc.ku.edu/~sdblunt/EECS844_Exam4

Provide: Complete and concise answers to all questions
Matlab code with solutions as appropriate
All solution material (including discussion and figures) for a given problem **together** (*i.e.* don't put all the plots or code at the end)
Email final Matlab code to me in a zip file (all together in 1 email)

** All data time sequences are column vectors with **increasing** time index as one traverses down the vector. (you will need to properly orient the data into "snapshots")

1. Data set P1.mat contains the signal $\mathbf{x}$ and the desired signal $\mathbf{d}$ resulting from passing the input through an unknown system. Estimate the Wiener filter $\mathbf{w}_o$ for a filter length of $M = 30$.

   Next apply the Normalized LMS algorithm with a step-size of $\tilde{\mu} = 1/2$ and leakage factor of $\delta = 0.02$. Plot the squared error (in dB) versus iteration *n* and state what you observe (may need to zoom in near the beginning). Also plot the squared deviation (in dB) defined as $\|\mathbf{w}(n) - \mathbf{w}_o\|^2$ versus iteration *n* and likewise state what you observe. Initialize the NLMS filter to all zeroes.

2. Using data set P1.mat, compute the Karhunen-Loeve Transform (KLT) for the input to the adaptive filter. Use the resulting transformation to implement the Transform-Domain LMS algorithm, using $\tilde{\mu} = 0.5/M$ and $\delta = 0.02$. Plot both the squared error (in dB) and squared deviation (in dB) versus iteration *n*. What is observed relative to the results of Problem 1 in terms of convergence speed and steady state performance?

3. Using data set P1.mat, implement Recursive Least-Squares and plot both the squared error (in dB) and squared deviation (in dB) versus iteration. Set $\mathbf{P}(0) = \mathbf{I}$ and use each of the following "forgetting factors": $\lambda = 1$, $\lambda = 0.9999$, $\lambda = 0.999$ and $\lambda = 0.99$, (just show one squared error and one standard deviation plot with all 4 cases on each). What is observed relative to the results of Problems 1 and 2 in terms of convergence speed for the case that yields the same steady state performance? What is the impact of different forgetting factors?

4. Data set P4.mat contains a data record of time-domain samples. For the cases of $K = 1$, 4, 8, and 16 segments plot the Bartlett estimate of power spectral density. Comment on what you observe. *Note: For problems 4 & 5, implement yourself, not with the MATLAB function (though you can use that to check your results).*

5. Repeat Problem 4 using the Yule-Walker method for AR models of order $p = 2, 3, 4$, and 5. How do the results differ from those obtained using the Bartlett method? *Hint: use the 'freqz' command to evaluate the frequency response of each AR model (with 'whole' modifier).*

6. Data set P6.mat contains two sets of spatial data from a length $M = 12$ element uniform linear array for $L = 50$ time samples, where 4 signals are known to be present. For data matrix **Xuc** the signals are uncorrelated and for data matrix **Xc** they are coherent (completely correlated), but they are otherwise arriving from the same directions. Implement MUSIC (yourself, not the built-in function) for the dimensionality of the noise-only subspace set to $m = 1, 2, \ldots, 12$, plotting the resulting pseudo-spectrum for all cases. Given that you know the number of signals present, discuss what you observe as you change the size of the noise-only subspace being used. *Note: the uncorrelated case can be viewed as a "ground truth" for the correlated case.*

7. Repeat Prob. 6 using a modification of the MUSIC algorithm in which each eigenvector is weighted by the inverse square-root of its associated eigenvalue.

8. Compare the results in Prof. 7 to what is obtained using MVDR and explain the relationship mathematically (i.e. show math in your explanation).

9. Repeat Problem 6 again using spatial smoothing as defined in Appendix A and a sub-array size of $\tilde{M} = 8$, where the dimensionality of the noise-only subspace can now be $m = 1, 2, \ldots, 8$. How are these results the same or different from those in Problem 6?

Appendix A – Spatial Smoothing

Given the $M \times L$ spatial data matrix **X** for a M-element uniform linear array, spatial smoothing can be achieved by averaging the covariance matrices formed from multiple sub-arrays of the data. Defining the components of the data matrix as

$$\mathbf{X} = \begin{bmatrix} x_1(0) & x_1(1) & \cdots & x_1(L-1) \\ x_2(0) & x_2(1) & \cdots & x_2(L-1) \\ \vdots & \vdots & & \vdots \\ x_M(0) & x_M(1) & \cdots & x_M(L-1) \end{bmatrix},$$

and a sub-array size of $\tilde{M}$, then $K = M - \tilde{M} + 1$ sub-arrayed data matrices can be formed where the k^{th} sub-array of the data is the $\tilde{M} \times L$ matrix

$$\mathbf{X}_k = \begin{bmatrix} x_k(0) & x_k(1) & \cdots & x_k(L-1) \\ x_{k+1}(0) & x_{k+1}(1) & \cdots & x_{k+1}(L-1) \\ \vdots & \vdots & & \vdots \\ x_{k+\tilde{M}-1}(0) & x_{k+\tilde{M}-1}(1) & \cdots & x_{k+\tilde{M}-1}(L-1) \end{bmatrix}.$$

Thus the spatially-smoothed covariance matrix (of size $\tilde{M} \times \tilde{M}$) is formed as

$$\mathbf{R}_{\mathrm{SS}} = \left(\frac{1}{LK} \right) \sum_{k=1}^{K} \mathbf{X}_k \, \mathbf{X}_k^H \; .$$