EECS 844 – Fall 2016
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 Thursday Dec. 15 by noon turn in at EECS Dept. office)

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 $\mathbf{w}_\text{o}$ for a filter length of $M = 60$.

   Next apply the Normalized LMS algorithm with a step-size of $\tilde{\mu} = 1/2$ and leakage factor of $\delta = 0.03$. 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 $\left\| \mathbf{w}(n) - \mathbf{w}_\text{o} \right\|^2$ 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 $\tilde{\mu} = 0.125/M$ and $\delta = 0.03$. 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": $\lambda = 0.99$, $\lambda = 0.998$ and $\lambda = 0.9995$, (just show one squared error and one standard deviation plot with all 3 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 $K = 1$, 6, 12, 24, and 48 segments plot the Bartlett estimate of power spectral density. Comment on what you observe. *Note: For problems 4 & 5, implement yourself, not the MATLAB function*.

5. Repeat Problem 4 using the Yule-Walker method for AR models of order p = 4, 6, and 8. 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.*

6. Data set P6.mat contains four sets of spatial data denoted as **X1**, **X2**, **X3** and **X4**, each collected from a length M = 25 element uniform linear array for N = 50 time samples. We wish to determine the number of sources present in each of the data sets so that we can apply the MUSIC algorithm to determine their respective directions of arrival. Using the paper by Chen, *et al* entitled "A comparative study of model selection criteria for the number of signals," implement the *Bayesian Information Criterion (BIC)* via equations (15) and (18) to <u>estimate the model order for each data set</u>. What assumptions are being made about the data that could impact the number of sources that are estimated?

7. Repeat Problem 6 using <u>forward-backward averaging</u> as defined in Appendix A. How are these results the same or different from those in Problem 6? Are there benefits and/or drawbacks to FB averaging? (and if so, what are they?)

8. Repeat Problem 6 again using <u>spatial smoothing</u> as defined in Appendix B and a sub-array size of $\tilde{M} = 12$. How are these results the same or different from those in Problems 6 & 7? Based on your observations, are there benefits and/or drawbacks to spatial smoothing? (and if so, what are they?)

9. The four data sets in Problem 6 correspond to the four cases listed below (with the <u>number and direction of the incident signals being the same</u> across all four cases). From what you have observed in the previous problems, determine which data set corresponds to each case (there is one of each). Discuss how you make each determination.

   a) ideal array (no calibration errors) & temporally uncorrelated signals
   b) ideal array (no calibration errors) & temporally correlated signals
   c) realistic array (calibration errors present) & temporally uncorrelated signals
   d) realistic array (calibration errors present) & temporally correlated signals

10. Using the four data sets in P6.mat and the estimated model orders and correlation matrices from Problems 6, 7, and 8 <u>implement</u> (yourself) the MUSIC algorithm and plot the pseudo-spectrum (for each implementation plot the results from the 4 data sets together for a total of 3 plots). Comment on what you observe for each case. Presuming the peaks must be easily discernible, how many peaks does MUSIC reveal for each case?

Appendix A – Forward-Backward Averaging

One way to form the forward-backward averaged covariance matrix given the $M \times L$ data matrix **X** for a uniform linear array is

$$\mathbf{R}_{\mathrm{FB}} = \frac{1}{2L}(\mathbf{X}\mathbf{X}^H + \mathbf{J}\mathbf{X}^*\mathbf{X}^T\mathbf{J})$$

where $\mathbf{J}$ is the $M \times M$ reflection matrix (looks like a reversed identity matrix) defined as

$$\mathbf{J} = \begin{bmatrix} 0 & \cdots & 0 & 1 \\ \vdots & \iddots & 1 & 0 \\ 0 & \iddots & \iddots & \vdots \\ 1 & 0 & \cdots & 0 \end{bmatrix}.$$

Appendix B – 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 m^{th} sub-array of the data is the $\tilde{M} \times L$ matrix

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

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

$$\mathbf{R}_{\mathrm{SS}} = \left(\frac{1}{LK}\right)\sum_{m=1}^{K} \mathbf{X}_m \mathbf{X}_m^H .$$