EE640 STOCHASTIC SYSTEMS
SPRING 2008
COMPUTER PROJECT 1

PART C: DETECTION and DISCRIMINATION (updated 4-14-09)

1. Fisher Discriminant: Perform a fisher discriminant measure\(^1\) (Rayleigh quotient) on each of the target/clutter vector pairs:

\[ t_{li}, c_{li} \quad i = 1,2,3 \]

\[ J_i = \frac{(\mu_{li} - \mu_{ci})^2}{\sigma_{li}^2 + \sigma_{ci}^2} \quad i = 1,2,3 \]  
\[ (C-1) \]

where \( \sigma_{li}^2 = \frac{1}{N-1} \sum_{m=1}^{N} (t_{li}[m] - \mu_{li})^2 \)

\[ \sigma_{ci}^2 = \frac{1}{N-1} \sum_{m=1}^{N} (c_{li}[m] - \mu_{ci})^2 \]

2. Mathematically and in words describe a MLR test for 3 correlated random variables. Assume equal covariances and reduce to the linear form. Using vector pairs (target/clutter pair \( t_{li}, c_{li} \)) from problem 1, with the highest \( J_i \) value first, form a MLR discriminator for one, two and three r.v.s.

a. Use the average of the target and clutter covariance matrices to get a common Covariance Matrix. For each of these discriminators, plot the target data response and clutter data response\(^2\) on the same graph. Estimate minimum probability of error from graphs and plot the three MPE values. Assume \( P(\text{target}) = P(\text{clutter}) = 1/2 \).

b. Repeat process in 2(a) for the independent data vectors but don't make any assumptions about the Covariance matrices being diagonal (i.e., keep in quadratic form). Only do the 3 variable case.

3. Form a M=64 long detection filter from a segment of a noise sequence and correlate with the original sequence. Plot correlation and note peak location. At peak location in output correlation determine the performance measures\(^3\), Signal-to-Noise Ratio, \( \text{SNR} = \text{peak}^2 / \text{variance} \) and also Peak-to-Average Correlation Energy, \( \text{PACE} = \text{peak}^2 / (\text{average squared value of output correlation}) \). The correlation is given in terms of convolution as:

\[(C-2)\]
\[ y[n] = x[n] \otimes h^*[n] \]

where in the frequency domain this is

\[ Y[k] = X[k]H^*[k] \]  \hspace{1cm} (C-3)

The entire process is implemented in MATLAB by \( y = \text{ifft}(\text{fft}(x) \cdot \text{conj(fft(h)))} \). Solve this problem for the following two cases:

The signal sequence \((x[n])\) is \( \text{binary} \) and the filter is

\[ h[n] = \begin{cases} b_{\text{binary}}[n + m_1] & \text{for } n = 1, 2, \ldots, M \\ 0 & \text{for } n = (M + 1), (M + 2), \ldots, N \end{cases} \]  \hspace{1cm} (C-4)

where \( m_1 \) is chosen arbitrarily between 1 and \( N-M \).

b. Repeat 3a for \((x[n])\) is \( \text{density int.} \).

4. ENCODING INFORMATION INTO THE COLOR NOISE OF A RGB IMAGE

Your objective is to encode and decode a pattern hidden in the color noise of an RGB image. Display the images of each step and describe with a few words their dimensions and contents. Assume the host image is a 24 bit per pixel BMP formatted image with dimensions \( M \times N \) pixels. You will input and separate this image into three \( M \times N \) 8 bit component images. You can use the following MATLAB code to do this

\[
A_{\text{bmp}} = \text{double(imread('example.bmp'));} % load example.bmp image \\
A_r = A_{\text{bmp}}(:,:,1); \\
A_g = A_{\text{bmp}}(:,:,2); \\
A_b = A_{\text{bmp}}(:,:,3);
\]

where the matrix \( A = \{ A_r, A_g, A_b \} \)

Define a set of 6 signature tiles. For each Red, Green and Blue image, there is a target tile and a clutter tile. We will refer to these tile pairs as \( \{ T_r, C_r \}, \{ T_g, C_g \} \) and \( \{ T_b, C_b \} \). The dimensions of the tiles are \( M_t \times N_t \) where \( M \) and \( N \) are integer multiples of \( M_t \) and \( N_t \), respectively. These tiles are uniform noise with their pixel ranges from \(-d\) to \(+d\), zero mean and all pixels are iid. The message image \( B \) is binary (you can use MS paint to create this and then input using imread) and of dimensions \( (M/M_t) \times (N/N_t) \).

Create the noise images \( B_r, B_g \) and \( B_b \) by using a Kronecker multiplication, “\( \otimes \)”, such that

\[
B_r = B \otimes T_r + (1-B) \otimes C_r
\]
\[ B_g = B \otimes T_g + (1-B) \otimes C_g \]
\[ B_b = B \otimes T_b + (1-B) \otimes C_b \]

Add the noise to the host image components (be sure to clip or limit sum values between 0 and 255) such that

\[ E_r = A_r + B_r \]
\[ E_g = A_g + B_g \]
\[ E_b = A_b + B_b \]

You should choose “d” to minimize the apparent noise corruption of \( A \). Try \( d=5 \) for starters but see what the minimum value should approximately be.

Combine \( E_r, E_g, \) and \( E_b \) to form an RGB image. You can use imwrite to store it and email me a copy of \( A \) and \( E \) to look at via email or get a color print out and include in your project.

Now you need to decode the image.

Input \( E \) and separate into \( E_r, E_g, \) and \( E_b \). Correlate with the appropriate signature filters. Indicate your filter design considerations. Show individual correlation and the sum of all three correlation planes. You may have to threshold and binarize these to bring out the peak locations (you can use negative images to save ink) and also plot final correlation in 3-D mesh so the correlation plane topology can be seen.

**SUGGESTED PROJECT LAYOUT (1)**

(Note: this is just a suggested layout, it does not guarantee an A grade)

1. Title Page
2. Introduction and description of project. (Turn in with Part C)
3. Project 1A. Discussion of Synthesis (no more than 1 page).
4. Project 1B. Discussion of Analysis (no more than 1 page)
5. Project 1C. Discussion of Detection and Discrimination
   Explain and present Fisher ratios.
   Derive MLR tests for 1,2 and 3 variables.
   Plot results for 1 variable. Plot results for 2 variables.
   Plot results for 3 variables.
   Plot out and describe problem 1C (3).
   Plot out and describe details of problem 1C (4).
6. Conclusions and References.

Appendix: Source Code.
References


APPENDIX

RGB ENCODING

% rgbencode.m by LGH 4-23-02
% encodes example.bmp using messageee640.bmp
% stores encoded image into encode.bmp
% retrieves encode.bmp and decodes messageee640.
% parameter d controls range of noise signature from -d to d
d=40;
%
% The number of rows and columns of messageee640.bmp must be integer
% multiples of the row and column sizes of example.bmp
% The example.bmp is a 24 bit BMP format
% The messageee640.bmp is an 8 bit BMP format with just 2 values, 0 and 1 for
% each pixel
%
% Fig. 1 Red component in B&W
% Fig. 2 Green component
% Fig. 3 Blue component
% Fig. 4 complimented binarized messageee640.bmp
% Fig. 5 Tiled Red signature image containing message
% Fig. 6 Red image + tiled Red signature image
% Fig. 7 Original image stored in example.bmp
% Fig. 8 Encoded image stored in encode.bmp
% Fig. 9 Decoded message image
% Fig. 10 Binarized version of Fig. 9 with threshold between max and min