ENGG7302: Advanced Computational Techniques in Engineering
Assignment 3

Overview

There are 9 questions in this assignment which count towards 10% of your final mark for ENGG7302. It aims to develop your skills in programming in Matlab, to deepen your understanding of Stochastic processes and to explore its applications. Your solutions to this assignment will be Matlab program listings and relevant output, together with any other mathematical derivations, notes or explanations that aids understanding. In marking the assignment, the following criteria will be applied:

- correctness of the programs,

- clear and concise documentation, in the form of comments in the code, as to the approach being used, and

- relevance of the results in verifying the correctness of the program and in illustrating the solution.

Some consideration will also be given to the efficiency of the solution (does the program make reasonable use of time and memory?). Although electronic submission of your code is not automatically required, the marker may, at his/her discretion, ask you to demonstrate your code in order to verify its correctness.
Due date: 5pm, Thursday, 10th November 2011.
Where to submit: through the Faculty of EAIT (Hawken Building 50) assignment chute. Your submission requires an assignment cover sheet. This is available from `http://www.eng.uq.edu.au/courses.asp`

Computational algorithms for data filtering

Introduction

In most communication systems, may it be wired or wireless, the data transmitted over some medium is usually corrupted by interference or crosstalk. Such systems mostly take the form $\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{z}$, where $\mathbf{H}$ is the channel matrix containing the intereference or crosstalk information of a medium, $\mathbf{y}$ is the received signal vector at the end user, $\mathbf{x}$ is the vector of transmitted data and $\mathbf{z}$ is the noise vector, generally, additive white Gaussian noise (AWGN). Crosstalk has been a source of severe performance degration in these systems. Competitive research is being done to invent state of art interference filtering techniques to reproduce an estimate of $\mathbf{x}$ from the received $\mathbf{y}$ at the receiver.

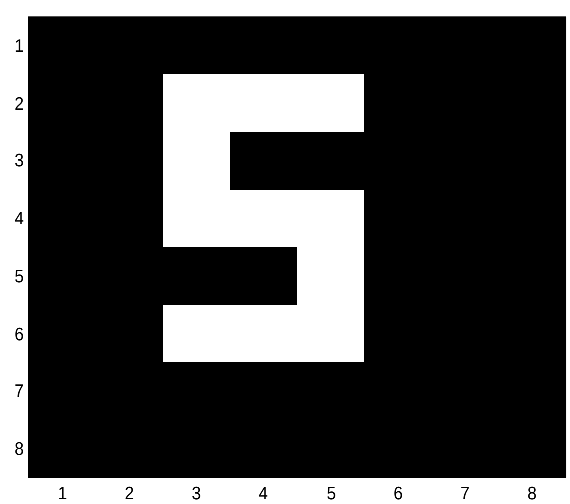
The goodness of such a filtering technique is measured by an error curve. An error curve is a plot of signal to noise ratio (SNR, on the x-axis) vs. the error rate between the transmitted and the filtered signal (on the y-axis). Such a plot is displayed on a log scale using a semilog, semilogy to be precise, function in Matlab. In this assignment, we aim to design a filter using an adaptive filtering algorithm and later, study/plot its error curve for a noisy character recognition using Matlab.

Procedure to follow

We begin by displaying a character ($\mathbf{X}$) which we intend to transmit over the channel $\mathbf{H}$. For instance a character 5 is created using the following matrix.

$$\mathbf{X} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

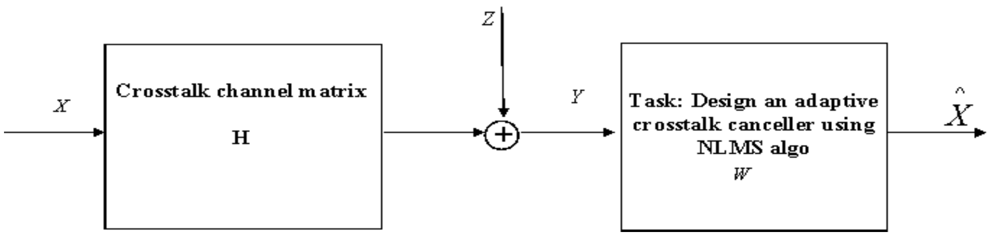
The image is displayed by using the command
>>imagesc(X)
>> colormap('gray') to look like



Using the same procedure, create the same or perhaps a different character as per your choice which you intend to transmit as $\mathbf{X}$. For simplicity, lets assume $\mathbf{X}$ is a square matrix of the size $M \times M$. Now create a $M \times M$ random channel matrix, $\mathbf{H}$, using the 0.1*randn function which contains the channel interference information. Create a vector σ^2, from say 0 to 1 in steps of 0.01 (or smaller such that the SNR is nicely spread out) which will be our noise amplitude. For each noise amplitude create a noise matrix $\mathbf{Z}$ (of the size of the input signal) again using the 'randn' function in Matlab. Ultimately, create $\mathbf{Y}$ using

$$\mathbf{Y} = \mathbf{H}\mathbf{X} + \sigma^2\mathbf{Z}$$

This is our received data $\mathbf{Y}$ corrupted with interference effects, and using this we intend to design a filter $\mathbf{W}$ (for each noise amplitude) which would filter the crosstalk/interference effects to yield an estimate of $\mathbf{X}$. The following diagram would help you clear your task in this assignment.



Filtering algorithm

To design a filter, we will be following the Normalised Least Mean Squares algorithm. This is an adaptive FIR filter given by $\mathbf{W}$. It follows a learning rule and estimates the filter. One possible way of estimating such a filter recursively (in a *for* loop) is

$$\mathbf{W}^{(\text{new})} = \mathbf{W}^{(\text{old})} + \mu \frac{\mathbf{E}\mathbf{Y}^T}{\mathbf{Y}.\mathbf{Y} + \epsilon} \tag{1}$$

$$\hat{\mathbf{X}} = \mathbf{W}\mathbf{Y} \tag{2}$$

$$\mathbf{E} = \mathbf{X} - \hat{\mathbf{X}} \tag{3}$$

In the second equation we are applying the estimated filter to our received signal $\mathbf{Y}$ to yield a close estimate of that transmitted ($\mathbf{X}$) which we call $\hat{\mathbf{X}}$. μ is some user-selectable constant in the range of $0 < \mu < 1$ and set ϵ to 0.01. This filter requires some amount of training samples so probably, the initial estimate of the filter may not be the best estimate. Make sure you place the above equations in a recursive loop running for say $m = 1000$ (or preferably more) iterations. Also check how the error (equation 3) changes according to time.

Once the estimation of the filter is complete (after *for* loop) get the filtered image using the final estimate of $\mathbf{W}$ and performing a decision like

$$\hat{\mathbf{X}} = \lfloor \mathbf{W}\mathbf{Y} \rceil \tag{4}$$

where $\lfloor . \rceil$ is a decision operation such as rounding. Use (4) and now compare the filtered image with that transmitted (using symerr function) and get the average error for each noise amplitude. Note: There are several ways to implement this algorithm (depending on how you 'view' the data in equation 1 and the way you want to construct $\mathbf{W}$). There are number of references in the library (and on the internet) which could be of help.

[1] Haykin S. *Adaptive filter theory*, Prentice Hall, New Jersey, 1996.
[2] Farhang-Boroujeny B. *Adaptive filters: theory and applications*, Wiley, New York, 1998.

Please ensure you properly understand the algorithm before you answer the following questions.

Questions:

(1) Using Matlab, implement the NLMS filter to clean (filter) our noisy data $\mathbf{Y}$.
Hint: See that μ is properly set and kept constant for all noise amplitudes. Display the filtered image (equation (4)) using the imagesc function and compare it with that transmitted (original) for various noise amplitudes.

(2) How does μ depend on filter estimation?

(3) For the value of μ set in your code, estimate the amount of iterations the filter takes to reach least error/ converge.
Hint: You can use a plot function to see the convergence of the error curve in equation (3) and how the filter coefficients behave as time varies.

(4) Using symerr function, tabulate the average error in the filtered image ($\hat{\mathbf{X}}$) by comparing with the transmitted $\mathbf{X}$ for each noise amplitude.

(5) Using the semilogy function, plot the error curve against the SNR. Since our transmitted signal amplitude is 1, SNR is given by $10\log_{10}(\frac{1}{\sigma^2})$. For better Symbol error rate run your whole code including the recursion for large number of iterations.

(6) What can you conclude from the error curve? How effective has your filter been at various noise levels?

(7) Using the xcorr function in Matlab, check (or plot) the correlation between the transmitted data and the filtered data at various (but not all) noise amplitudes.

(8) What can you conclude from the correlation plots at these noise amplitudes?

(9) Write a general difference equation using one input column (of $\mathbf{Y}$) and one row of the filter ($\mathbf{W}$) to estimate the corresponding element(s) of $\hat{\mathbf{X}}$.

If you need further clarification on this assignment send me an email: m.gujrathi@uq.edu.au