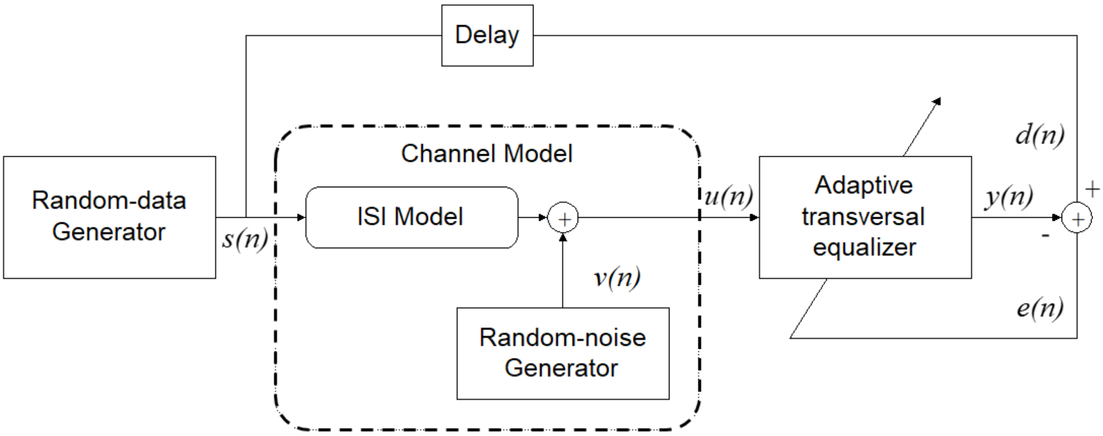
1061 Advanced VLSI HW1

Computer Experiment on Adaptive Equalization

TA Information
電二 232 實驗室
鄧傑方 jeff@access.ee.ntu.edu.tw



- The data are all **real valued**.
- Random-data generator generates a *Bernoulli sequence s(n)* with *s(n)*=±1, and the random variable *s(n)* has <u>zero mean and unit variance</u>.
- The impulse response of the channel (ISI model) is described by the raised cosine function (See Figure 1):

$$h_n = \begin{cases} \frac{1}{2}\left[1+\cos\left(\frac{2\pi}{W}(n-2)\right)\right], & n = 1,2,3 \\ 0, & \text{otherwise} \end{cases}$$

where the parameter W controls the amount of amplitude distortion produced by the channel. Basically, <u>the distortion increases with W</u>.

- The random-noise sequence *v(n)* has zero mean and variance $\sigma_v{}^2 = 0.001$ (Be sure to use the Gaussian noise.)
- The equalizer has M=11 taps.
- The delay stage D is set to be 7. The reason is that the channel has an impulse response h_n that is symmetric about time n=2, and the equalizer is symmetric about time n=5. Hence, the channel input *s(n)* is delayed by 2+5=7 samples to provide the desired response for the equalizer.
- The simulation results must be averaged over 200 independent trials of experiment (i.e., to obtain the ensemble average of the learning curve.)

◆ All the initial guess of weights (filter coefficient) are set to be zero, *i.e.*, w_i=0 for i=0,1,…,M-1.

◆ Figure 3 shows the typical example of the learning curve. The X-axis is the number of iterations, n, and the Y-axis is the ensemble-averaged square error, $E\{e^2(n)\}$.

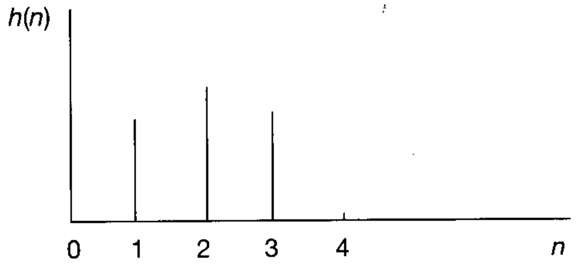


Figure 1

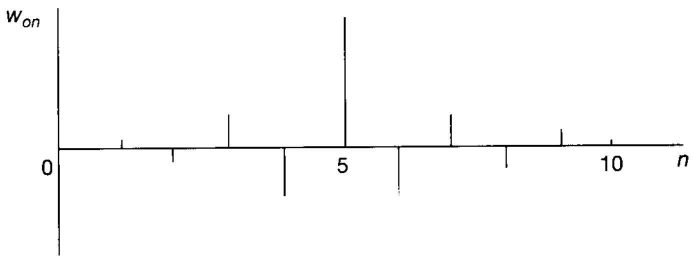


Figure 2

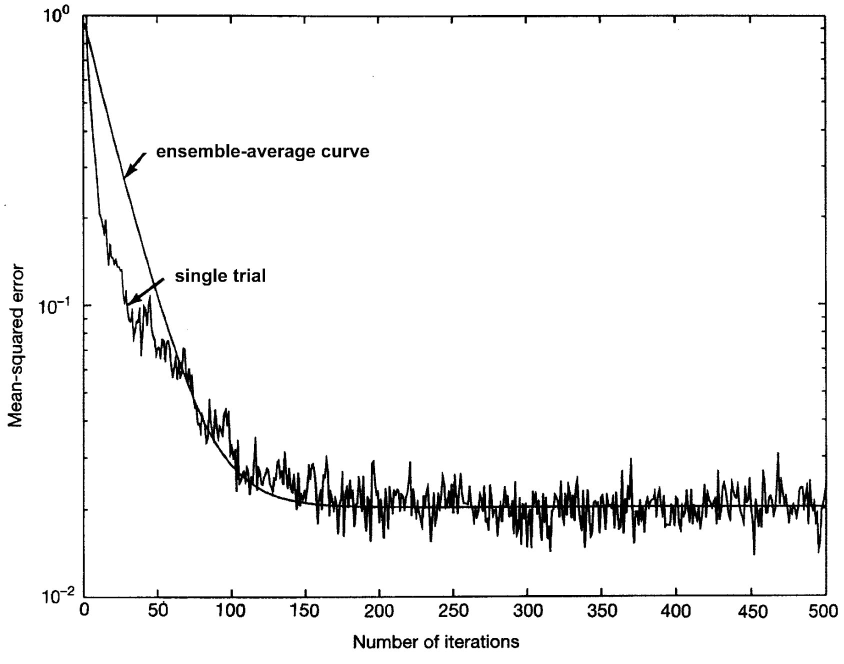


Figure 3

Homework requirement:

1. Assume that 1000 transmitted data and 1000 desired signal are collected. Write a C/Matlab program to solve the **Wiener-Hopf equation**. That is, compute the *correlation matrix* and *cross-correlation matrix*. Then solve for the optimal coefficients of the 11-tap equalizer. Repeat the computation for three different channels (W=2.9, W=3.1, and W=3.3.)
2. Repeat the above experiment using LMS algorithm
   (a) Plot the learning curve of E{$e^2(n)$} based on the following conditions, respectively:
      (i) W=2.9, μ=0.04, 0.02 and 0.01 (in one figure)
      (ii) μ=0.02, W=2.9, W=3.1 and W=3.3 (in one figure)
   (b) Plot the learning curves of the filter coefficients; *i.e.*, show the curves of the weights, w_i, versus number of iterations, n, for all i (let W=2.9, μ=0.02). Compare your converged coefficients with the Wiener-Hoft results.
3. Repeat the above LMS experiment (Problem 2) using "Normalized LMS algorithm (NLMS)." Compare its performance (e.g., maladjustment (see definition of Haykin's book) and convergence speed) with the conventional LMS algorithm in Problem 2(a)(ii) by plotting both of them in one figure.
   (NLMS: Use the 2nd equation in http://www.ee.ucr.edu/~yhua/ee211/Note_3.pdf)

Bonus:

Derive the maximum value of μ, and plot the learning curves to verify the value you find.

Deadline: 2017/10/8 23:59:59

1. The file name of the report should be **r06943xxx_hw1_report.pdf**
2. The simulation code should be upload
3. An easy readme.txt file should be provided to describe how to use your code
4. Put the <u>report</u>, <u>codes</u> and <u>readme</u> in the same folder, which is named **r06943xxx _hw1**
5. Compress the folder to **r06943xxx_hw1.rar/ r06943xxx_hw1.zip**
6. Submit the .rar/.zip to Ceiba before deadline