EECS 152B – CSE 135B

Winter 2018 Lab Assignment 3M

Due: Fri. March 9

The objective of this project is to investigate the performance of an adaptive equalizer for data transmission over a multipath channel that causes "inter-symbol interference" (ISI). The basic configuration of the system to be simulated is shown in the figure on the next page. Note that we have not included carrier modulation and demodulation, however all processing involves complex arithmetic operations (complex symbols as input, complex channel impulse response, etc). There are five basic modules that must be incorporated into the simulation:

1. The data generator module is used to create a sequence of complex valued information symbols $s[n]$. For this experiment we will assume QPSK symbols, or in other words that $s[n]$ must be drawn from the set $\{a+ja, a-ja, -a+ja, -a-ja\}$, where a represents the signal amplitude that is chosen according to a given signal-to-noise ratio (SNR). Assuming that the noise has unit power, then $\mathrm{SNR} = 20\log_{10}(\sqrt{2}a)$.

2. The channel filter module is an FIR filter with impulse response $c[n]$ that simulates the channel distortion.

3. The noise generator module is used to generate additive noise that is present in any digital communication system. For the noise, assume unit-power, complex Gaussian noise. In Matlab, this can be done using the command: `(1/sqrt(2))*(randn(1,N)+j*randn(1,N))`, which generates a vector of N noise samples.

4. The adaptive equalizer module is a length $M+1$ FIR filter $h[n]$ whose coefficients are adjusted using either the LMS or the normalized-LMS algorithm. (Remember: use the complex version of the filter!)

5. The decision device module takes the output of the equalizer and quantizes it to one of the four possible transmitted symbols, based on whichever is closest. This module is necessary when your equalizer is operating in decision-directed mode, where the detected symbols are used to train the filter.

The effectiveness of the equalizer in suppressing the ISI introduced by the channel may be seen by plotting the input and output of the equalizer in a two-dimensional (real-imaginary) display. If you run the full experiment a total of P times, then you will have P different values of the equalizer input and output for each time sample n, where $n = 0, \cdots, N-1$. If you plot these P points for each n, hopefully you will see that the output of the equalizer eventually converges to four tightly clustered clouds around the QPSK symbol values. You should also plot the error $e[n]$ as a function of n, averaged over the P experiments. This will give you a picture of the rate at which the equalizer is converging.

Your main Matlab simulation should have the following as input variables: a vector $\mathbf{c}$ containing the channel impulse response, the SNR in dB, the step size μ for the fixed-step LMS equalizer, the length M of the equalizer filter $\mathbf{h}$, the number of training samples T used to initially train the filter, the number of total samples N per experiment, and the total number P of experiments used to plot the results and compute the average error.

To test the functionality of your code, use the following test case: $\mathbf{c} = [1\ 0.2\ -0.4\ 0.3\ 0.2\ -0.1]^T$, $\mathrm{SNR} = 25\mathrm{dB}$, $\mu = 0.01$, $M = 10$, $N = 300$, $T = 30$, $P = 100$. You should get results similar to those shown in class. For your report, investigate the effect of different channels (including complex channels), different SNR values, step sizes, and filter lengths. Comment on the relative performance of the LMS and normalized-LMS algorithms, touching on convergence rate, stability and the final value of the error. Recall that channel filters whose first tap has the largest value will likely lead to the best performance. Channels whose strongest tap is not the first require the addition of the delay (shown in the figure below) to ensure that the error signal is being computed by comparing the transmitted symbol from the output of the filter that should correspond to that transmitted symbol. To avoid the need for the delay block, choose channel impulse responses whose first tap is larger than the rest.

Extra credit A: Implement the gradient descent filter for a value of $L > 1$, and compare its performance with the standard LMS algorithm that uses $L = 1$ (note that the best value of μ for the two approaches may be different). How much benefit is gained by increasing L?

Extra credit B: Implement the optimal recursive least-squares filter (the one with the matrix inverse) by using the first L samples of data to compute the filter, and then using the filter to process the reference signal. Experimentally show how big L has to be relative to the filter length to get good performance.

Points possible: 200 + 50 (Extra credit A) + 50 (Extra credit B) = 300.

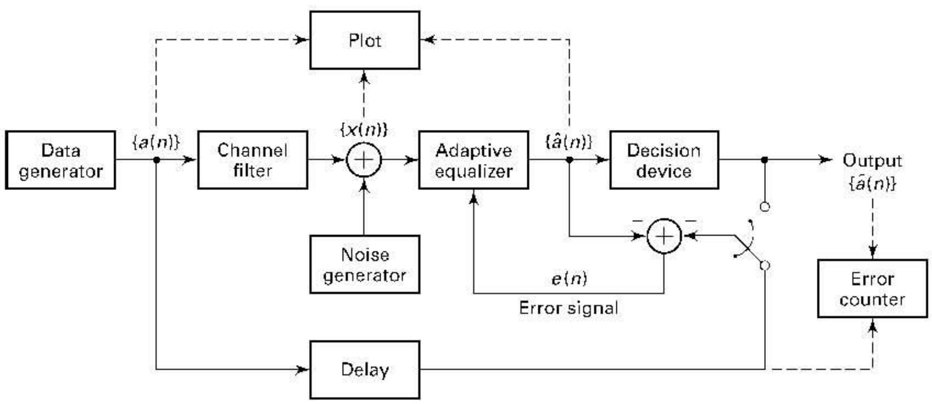

Figure 1: Adaptive Equalizer Block Diagram.