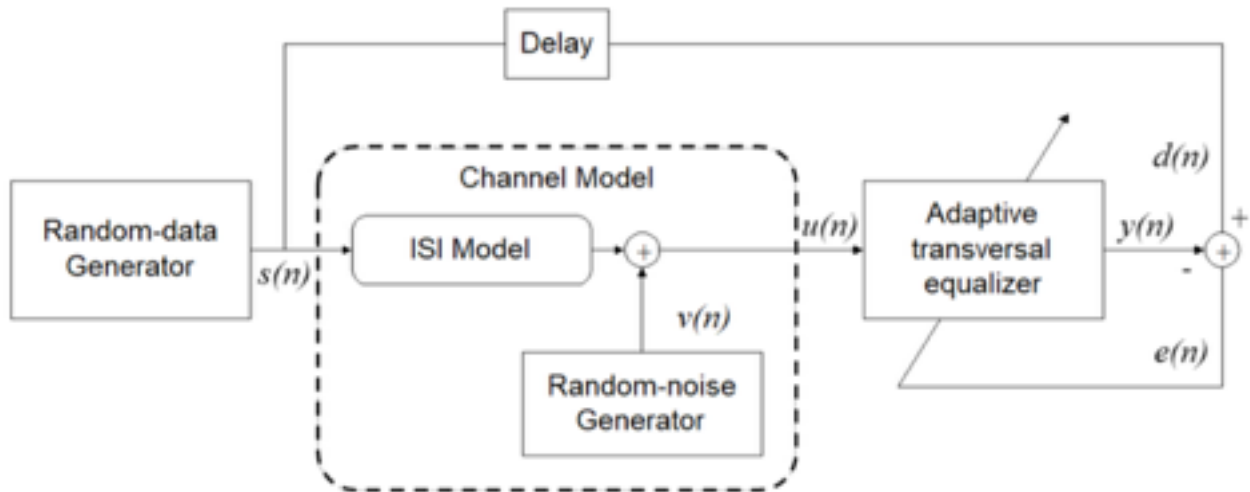


AVLSI Hw1 Report

studentID: b03901022

name: 卓伯鴻

1. architecture



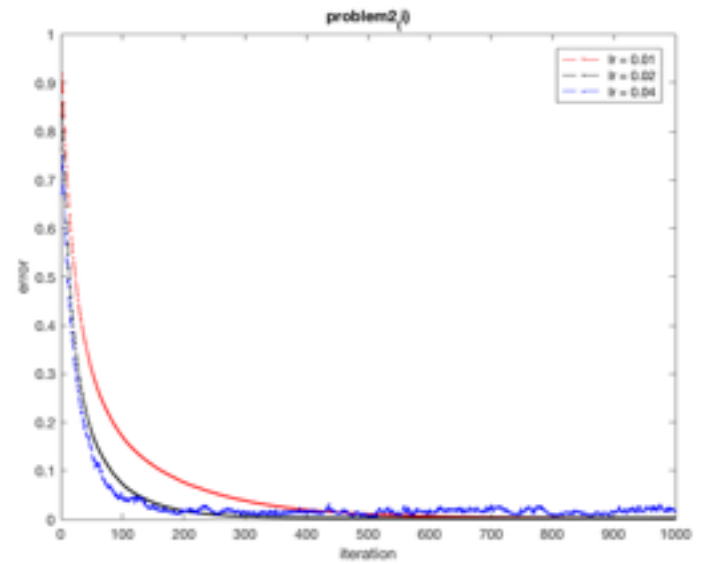
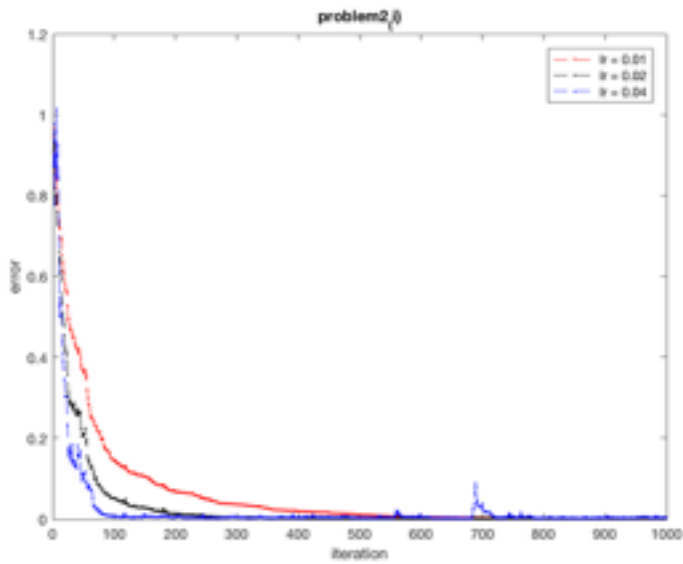
The input is a Bernoulli sequence (with value only equal to 1 or -1) whose size is (1000×1) randomly generated from a data generator. In ISI Model during Channel Modeling process, the impulse response of the channel is symmetric to time $n = 2$; In equalizer, it is symmetric to time $n = 5$ and has $M = 11$ taps, and thus the delay is set to 7 to match up the desired response for equalizer. In that, we will get $u(n)$ with size (1004×1) after Channel Modeling, and $y(n)$ with size (1014×1) after equalizing. With $d(n) = u(n - 7)$ we can compare the central 1000 result in $y(n)$ with $d(n)$ to get the root-mean-square error.

2. requirement 1

After the preprocessing of data, we get $u(n)$ with length of (1004×1) , and I concatenate with some zeroes at head and tail: $[0 \ 0 \ 0 \ u(n) \ 0 \ 0 \ 0]$, to assure we can get 1000 tap with tap $M = 11$. By calculating matrix $p = [u(n)d(n)]$ (a 11×1 matrix), and matrix $R = [u(n) \text{Inv}(u(n))]$ (a 11×11 matrix), I get matrix w from $\text{Inverse}(R) \times p$, which is the coefficient we want in the equalizer. In that, I can acquire $y(n)$ from the convolution of weight w and $u(n)$. Finally, get the root-mean-square error from $y(n)$ and $d(n)$, which is **0.002** in my simulation.

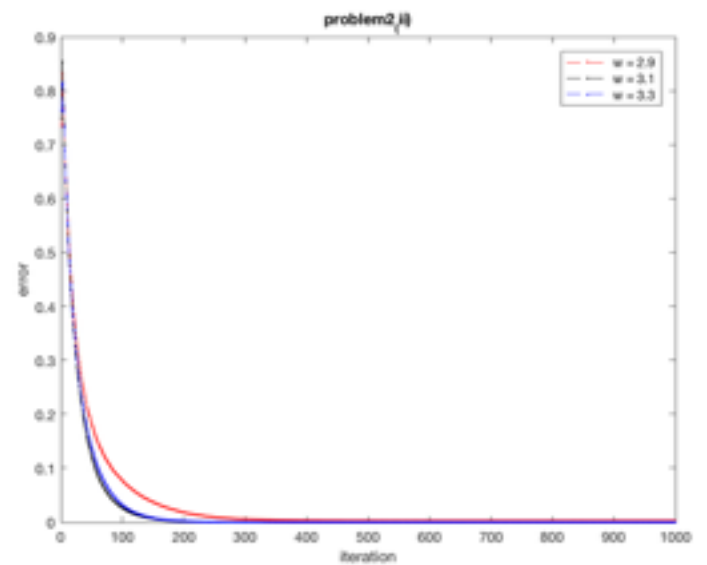
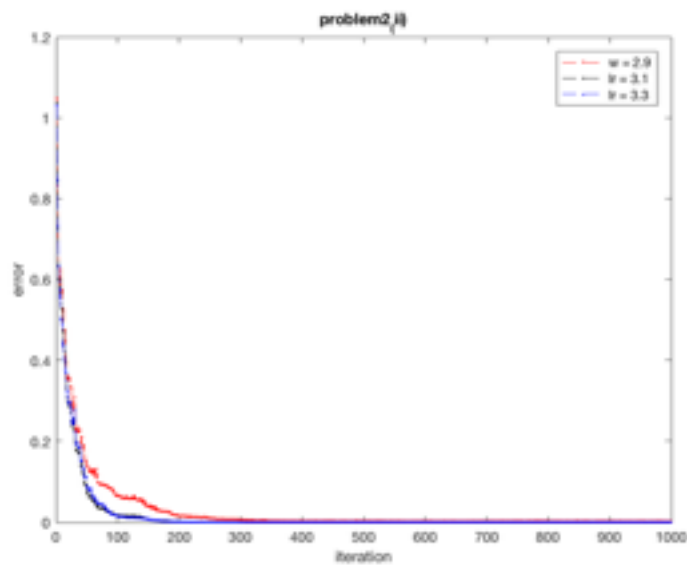
3. requirement 2.(a).(i)

Same as the previous requirement, I capitalize the same data processing. The most different part from it is the LMS algorithm which give a closer simulation situation to the real-world communication. I make a change to the weight coefficient every tap and record the corresponding root-mean-square error and thus acquire a learning curve of $E[e(n)^2]$. However, the plot will have some furry curves as every modification to weight coefficient is quite dependent to the original signal and its noise counterpart. As a consequence, I iterate the same process for 200 times to eliminate the furry part by get the average of them. The plots are as follows, while the left plot is the result of one iteration, and the right plot is the result of two iterations. In addition, I change the **learning rate** (value μ) in three different value: **0.01, 0.02, 0.04**.



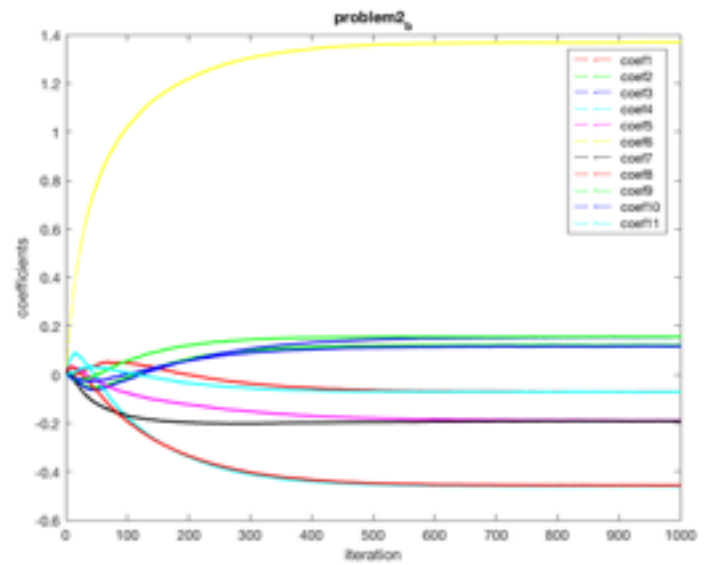
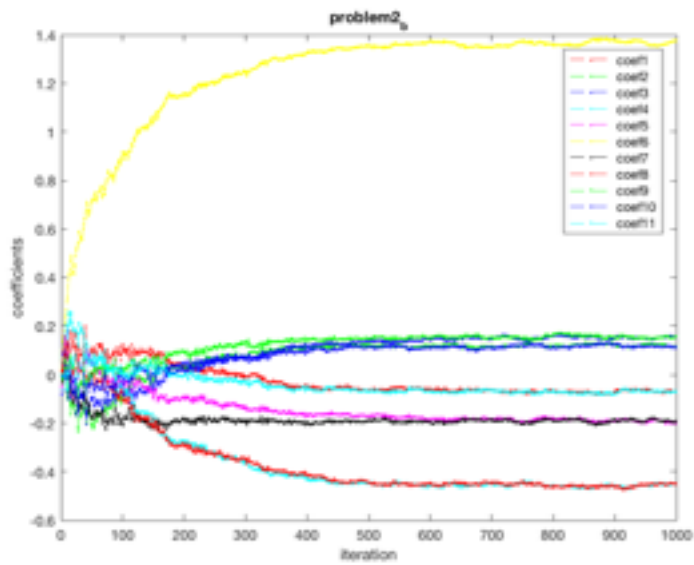
4. requirement 2.(a).(ii)

In this section, I change the parameter W controls the amount of amplitude distortion produced by the channel with three different value $W = 2.9, 3.1, 3.3$ and the plot are as follows, while the left plot is the case iteration = 1, and the right plot is the case iteration = 200.



5. requirement 2.(b)

In this section, $W = 2.9$ and $u = 0.02$, we acquire the 11 **weight coefficients** change in each tap, while the left plot represents iteration = 1, and the right plot represents iteration = 200.



6. requirement 3

In this section, I implement another method to modify the weight coefficient (NLMS), which change the rate of learning corresponding to its own L2-norm. The following plots represent the **performance of NLMS and LMS**, while the left plot shows the result of iteration = 1, and the right plot shows the result of iteration = 200.

