COA031: Broadband Transmission Technology, 2024
Department of Communication Engineering, NCU
Homework #1

Prof. *Dah-Chung Chang*

Date: 2024/3/27, Deadline: 2024/4/19

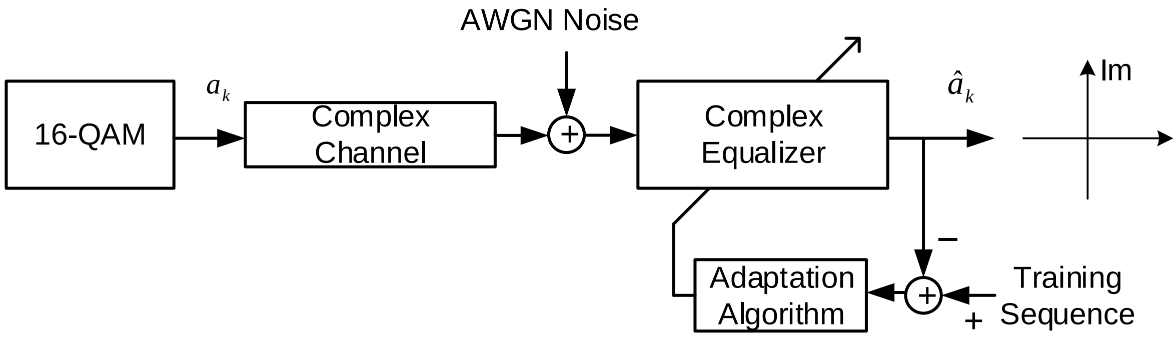
A 16-QAM baseband transmission system is depicted in the below figure. The taps of two evaluated baseband equivalent channels are listed as follows:

Channel #1=[1 0 0 0 0 0 0.4-j*0.3 0 0 0 0 0 0 0 0.2-j*0.1 0 0 0 -0.1];

Channel #2=[-0.3084-0.5368i, 0.0356+0.5375i, -0.0772+0.1704i, -0.1242+0.0886i, -0.0270-0.0481i, 0.0297+0.0053i, -0.0065+0.0001i, -0.0047+0.0127i];

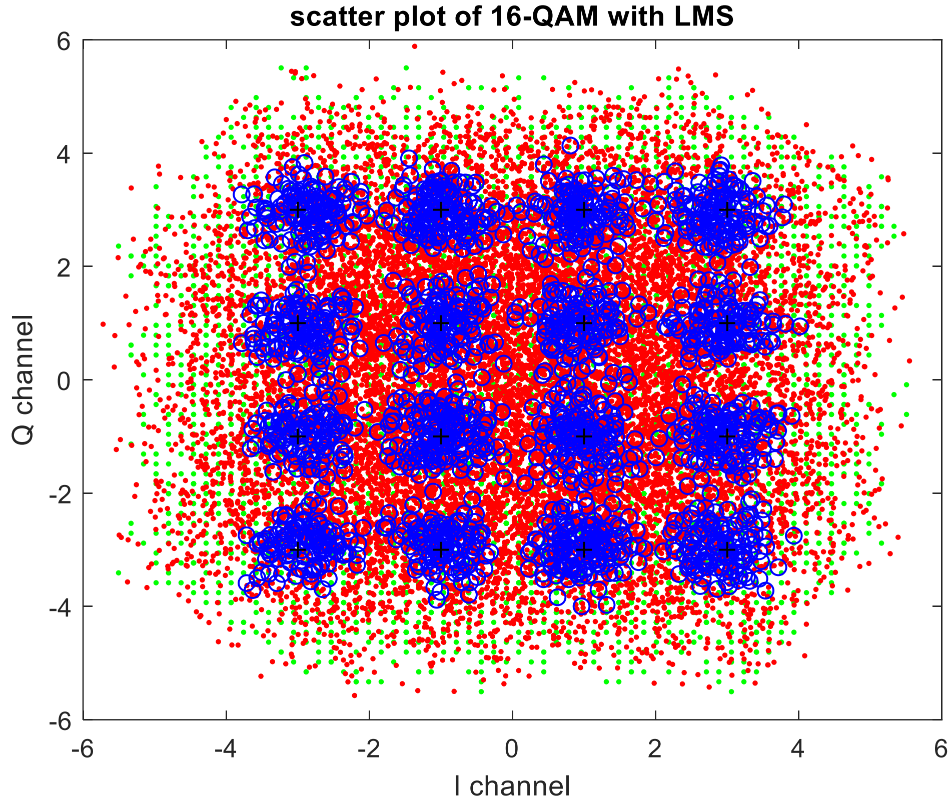
Assume that the complex AWGN noise is SNR=20 dB. All the environment setup is already in the appended MATLAB code. Please complete the N**LMS** and **RLS** algorithms for the simulation and show the scatter plots and the error signal learning curves of the received 16-QAM signals before and after equalization for the two channels.

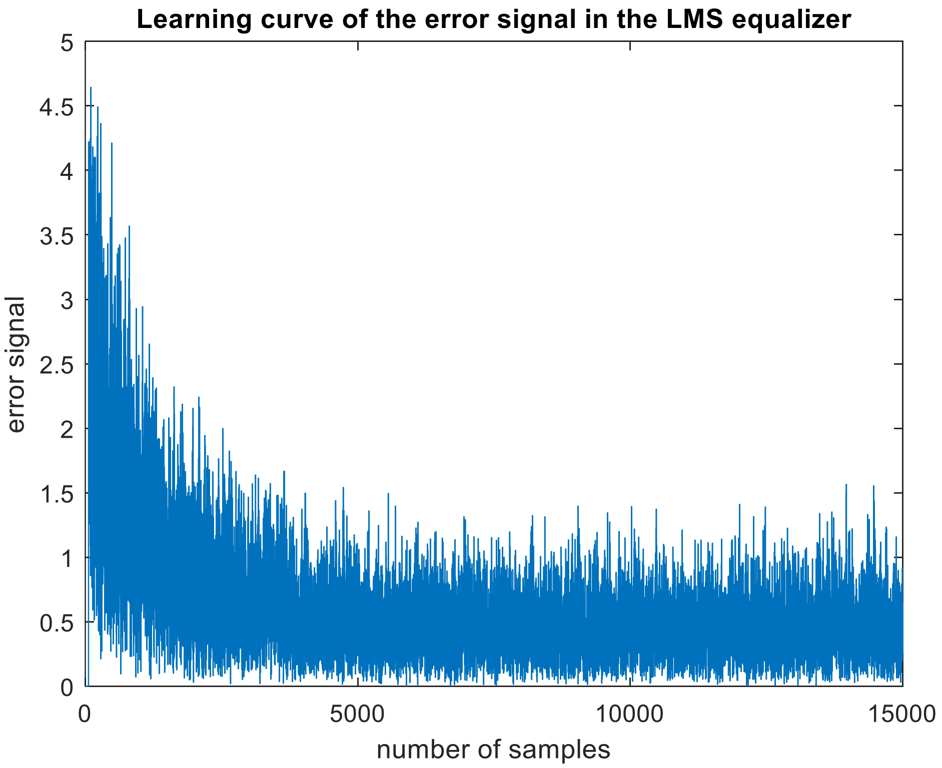
Discuss the results with different filter tap lengths (80, 60, or 20?) and step sizes (1, 0.1, 0.01, or 0.0001?).



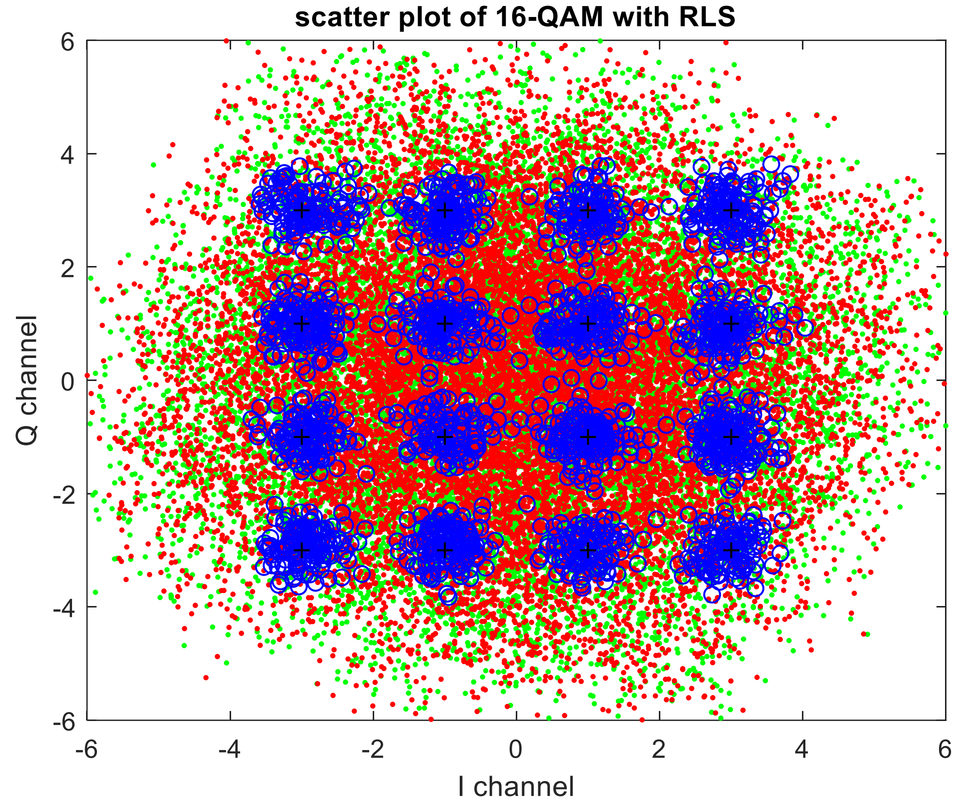
Reference Results:

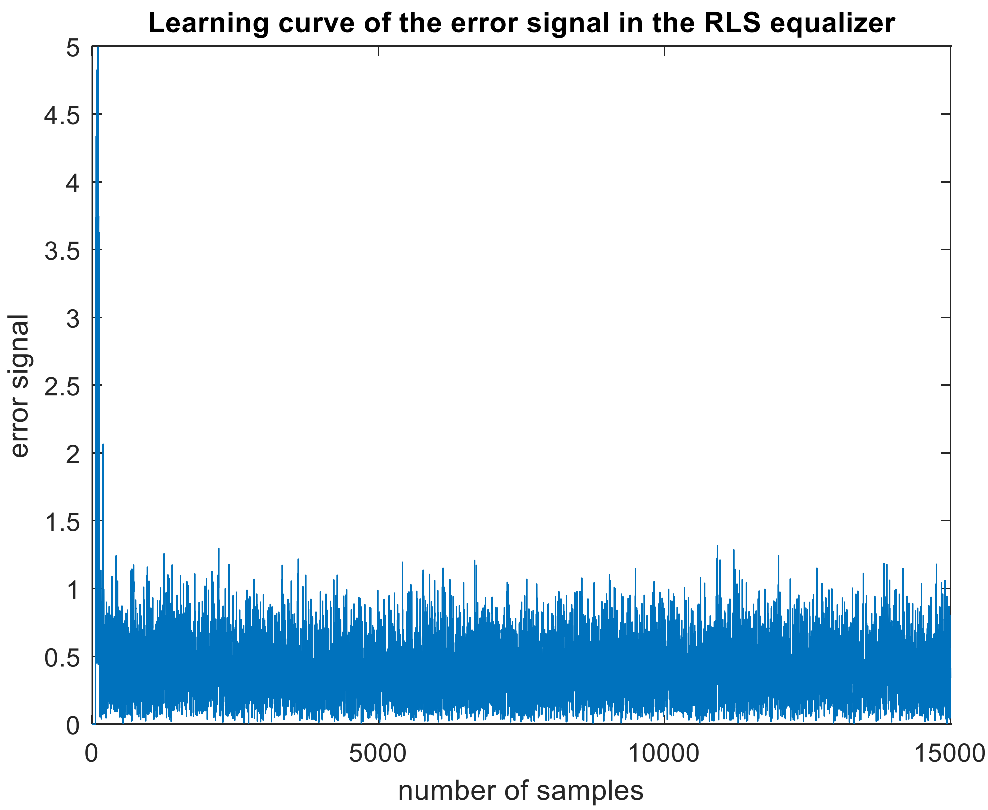
Channel #1: LMS





Channel #2 with RLS





Reference Code:

```
%
% BTT Ex1.
% DCChang NCU/CE
%
clear all
hold off

adaptive_mode=0;   %0 for NLMS, 1 for RLS
% 16-QAM Sig. Generator
M = 16;                     % Size of signal constellation
k = log2(M);                 % Number of bits per symbol
n = 60000;                   % Number of bits to process
                        EbNo =14;
snr = 20;                    % =EbNo + 10*log10(k);   %dB

dataIn = randi([0 1],n,1);  % Generate vector of binary data
dataInMatrix = reshape(dataIn,length(dataIn)/k,k);   % Reshape data into
binary k-tuples, k = log2(M)
dataSymbolsIn = bi2de(dataInMatrix);                 % Convert to integers
dataMod = qammod(dataSymbolsIn,M); % Gray coding, phase offset = 0

% Adding Channel and AWGN
channel_1=[1 0 0 0 0 0 0.4-j*0.3 0 0 0 0 0 0 0 0.2-j*0.1 0 0 0 -0.1];
channel_2=[-0.3084-0.5368i, 0.0356+0.5375i, -0.0772+0.1704i,
-0.1242+0.0886i, -0.0270-0.0481i, 0.0297+0.0053i, -0.0065+0.0001i,
-0.0047+0.0127i];
channel=channel_1;
channel=channel/norm(channel);
datach = conv(dataMod, channel);
receivedSignal= awgn(datach,snr,'measured');

% Channel Equalization
mu=0.1;                   %channel_1:0.1  channel_2:0.1
tap=60;                   %channel_1:60   channel_2:20
delay=10;                  %10
eps=10^(-8);
lambda=1;
P=diag(ones(1,tap));
w=zeros(tap,1);
it=n/k;
y=receivedSignal;          %received signal

if adaptive_mode==0
%% NLMS Algorithm
for n=max(tap,delay+1):it
%%-homework #1-NLMS Equalizer------------------------%%

                  ???

%%---------------------------------------------------%%
end
%%
else
%% RLS Algorithm
```

```
for n=max(tap,delay+1):it
%%--homework #1-RLS Equalizer-------------------------%%

                    ???

%%--------------------------------------------------------%%
end
%%
end

% Plot
figure(1)
plot(datach,'.g')
hold on
plot(receivedSignal,'.r')
plot(dataout(n-2000:n),'ob')    %dataout is the output of the equalizer
plot(dataMod,'+k')
axis([-6 6 -6 6])
hold off


figure(2)
plot(abs(em1))
```