

'eye eq' Program Tutorial

Jungsub Byun

When we send transmissions more closely in succession to increase the data transmission rate, interference between them is unavoidable. This phenomenon is called intersymbol interference(ISI) and this can be a devastating factor to the proper function of the optimum detector. So, communication engineers use equalization methods to minimize the effect of ISI. The equalization methods are designed to turn a bandlimited channel with ISI into a new memoryless flat channel at the receiver output. Actually, they hope to create another new flat AWGN-like channel.

One method to form a picture of this ISI is the eye diagram. If the trigger is synchronized to the symbol rate, the eye diagram is like the image of an oscilloscope. The eye diagram is depicted through overlapping a few serial symbol intervals of the modulated and filtered continuous-time waveform. In this eye diagram for binary transmission on a channel, a distinct opening can be noticed in the center of the plot. Because the ISI results in the spread among the path traces, the opening in the eye becomes narrower as the ISI increases.

Minimum-Mean-Square-Error Decision Feedback Equalizer(MMSE-DFE) utilizes the previous attempts to estimate the current symbol through an SBS detector. Any trailing intersymbol interference resulted from the former symbols is reorganized and then subtracted in the feedback system. So it is likely that the channel output signal can be a causal signal after going through the feedforward filter. The feedback section will then subtract (without noise enhancement) any trailing ISI.

For showing various effects on various channels in Equalization part, the 'eye_eq' program tutorial consists of

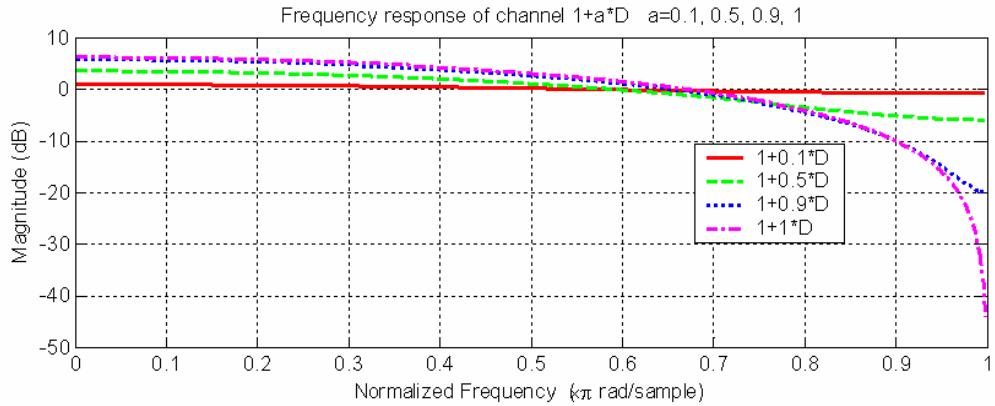
- [1] **Function description of eye_eq,**
- [2] **Plotting the frequency response of various channels,**
- [3] **Comparison between MMSE DFE and MMSE LE - FIR equalizer performance (SNR) versus varying from $1-1*D^{-1}$ to $1+1*D^{-1}$ channels with fixed SNRmfb/noise variance, and**
- [4] **Plotting and Comparison among different channel's 'EYE Diagram'- before Equalizer (Channel output)/ After Equalizer(Filter output). Calculate probability error with/without Equalizer – eye_eq program.**

[1] Function description of eye_eq

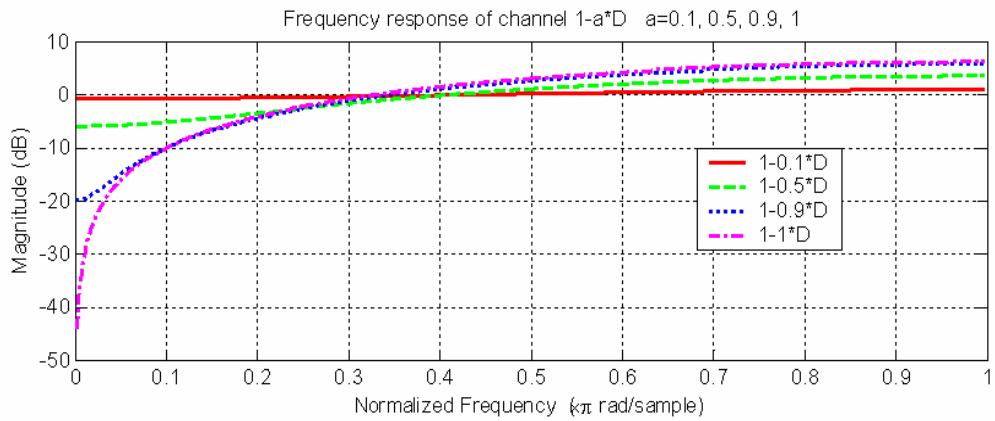
```
function []= eye_eq(p,Ex,noise_var,eq_type);
%
% p      = pulse response [a*D^-1 + 1] ==> [a 1]
% Ex     = average energy of signals, Ex_bar
% noise_var = noise variance
% eq_type = Z => ZERO FORCING
%           M => MMSE
%           D => MMSE-DFE
% outputs: pe_(zfe/mmle_dfe/mmse_le)= probability of error with Equalizer from N input
2PAM[+1/-1] data sequences
% outputs: pe_no_eq = probability of error without Equalizer
% outputs: dfseSNR = receiver(equalizer)SNR, unbiased in dB
% outputs: pe_SNR = error probability estimation from receiver(equalizer)SNR, Pe = Q
function of sqrt(dfseSNR)sdsds
% this function shows Frequency response of the channel p and equalizer filter, eye
diagram, receiver SNR, and probability of error.
% N = 250; % # of input 2 PAM[+1/-1] data, you can increase the N of input data
sequences in order to calculate the Pe accurately
% created 1/06 by Jungsub Byun and M. Malkin EE379A
%
function [dfseSNR,w_t]=dfecolor(l,p,nff,nbb,delay,Ex,noise); this program has come to be
used throughout the industry to compute/project equalizer performance. You will learn later.
Summary of eye_eq algorithm
For the channel p at SNRmf=10dB
a. generate the input binary 2PAM[+1 and -1] data sequences
b. make the data signal after the channel p without Gaussian noise through filter or conv
command
c. generate Gaussian noise
d. make received signal with noise for the ISI channel p at the receiver
e. make nff taps of feedforward filter and nbb taps of feedback filter by dfecolor program
f. convert the ISI received signal into a new AWGN-like channel at the receiver output, filtering
with nff taps of feedforward filter
g. make the operation part of nbb taps feedback filtering after feedforward equalizing
h. make the detection and decision part of the equalized data
i. calculate the error probability estimation after equalizing – the decision feedback equalizer of
nff taps of feedforward filter and nbb taps of feedback filter with comparing between the
equalized data and original input binary 2PAM[+1 and -1] data sequences
j. calculate probability of error without dfe equalizer, comparing between received signal with
noise for the channel p and original input binary sequences
k. estimate the probability of error from dfseSNR(dfecolor result), Pe = Q function of
sqrt(dfseSNR)
l. create EYE Diagram by a1=(sinc(x2)); a2=(sinc(x2-1)); a3=(sinc(x2-2)); % sinc function
m. Plot and compare EYE Diagrams –original binary Input data at the transmitter, before
Equalizer (Channel output), After Equalizer(MMSE DFE output), and decision data of after
MMSE DFE at the receiver.
```

[2] Frequency response of different channels

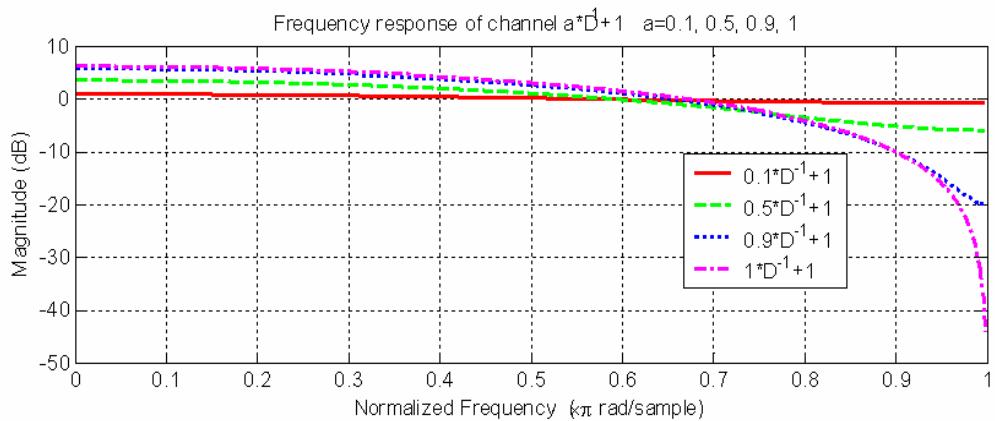
A. ('Frequency response of channel $1+a*D$ $a=0.1, 0.5, 0.9, 1$ ')



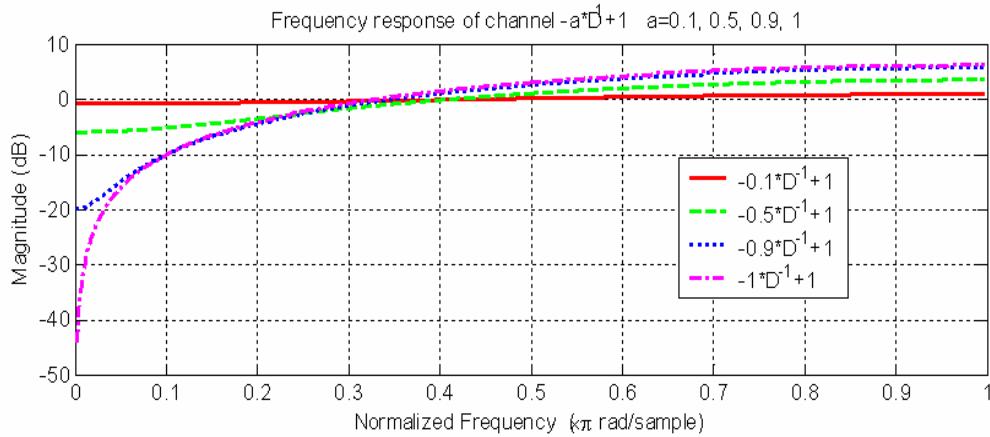
B. ('Frequency response of channel $1-a*D$ $a=0.1, 0.5, 0.9, 1$ ')



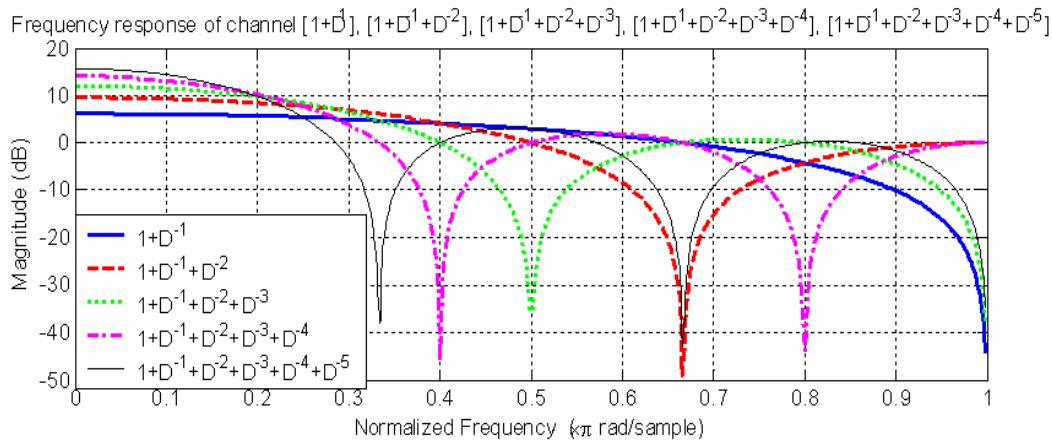
C. ('Frequency response of channel $a*D^{-1}+1$ $a=0.1, 0.5, 0.9, 1$ ')



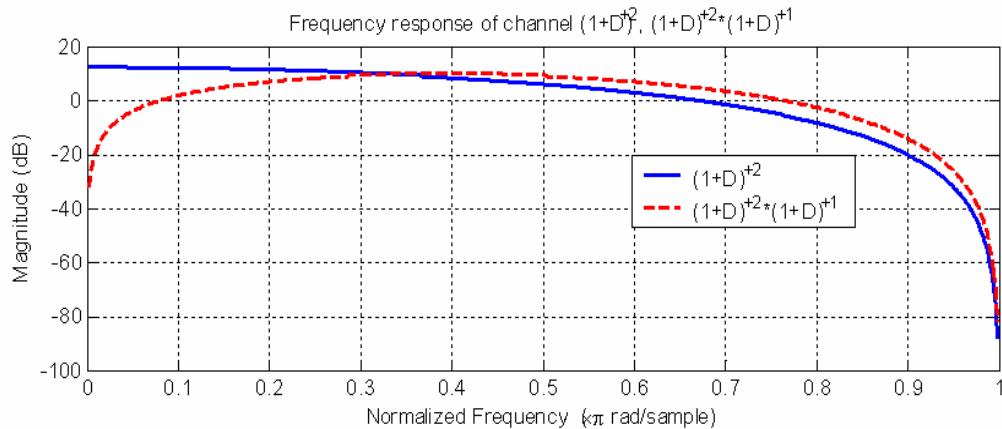
D. ('Frequency response of channel $-a*D^{-1}+1$ $a=0.1, 0.5, 0.9, 1$ ')



E. ('Frequency response of channel $[1+D^{-1}]$, $[1+D^{-1}+D^{-2}]$, $[1+D^{-1}+D^{-2}+D^{-3}]$, $[1+D^{-1}+D^{-2}+D^{-3}+D^{-4}]$, $[1+D^{-1}+D^{-2}+D^{-3}+D^{-4}+D^{-5}]$)

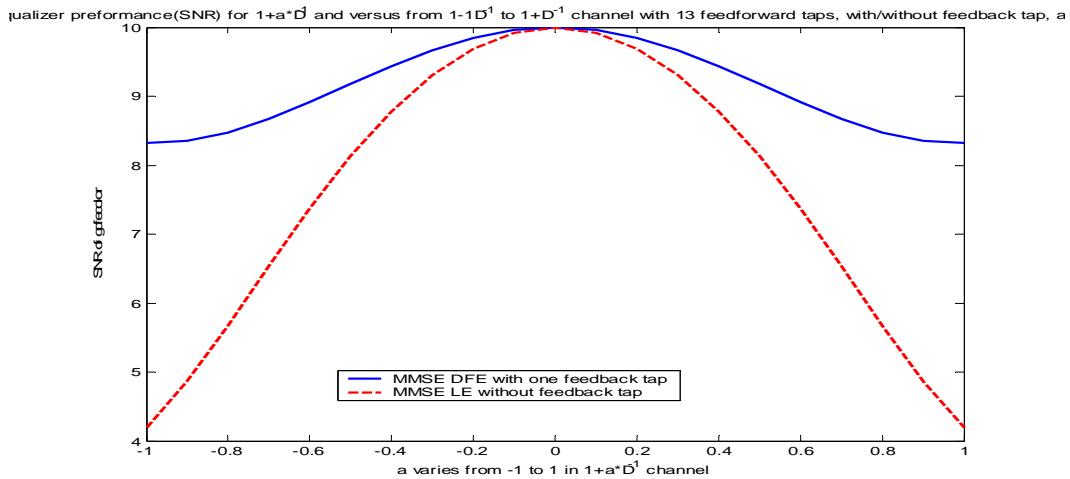


F. ('Frequency response of channel $(1+D)^2$, $(1+D)^{t2}*(1+D)+1$ ')



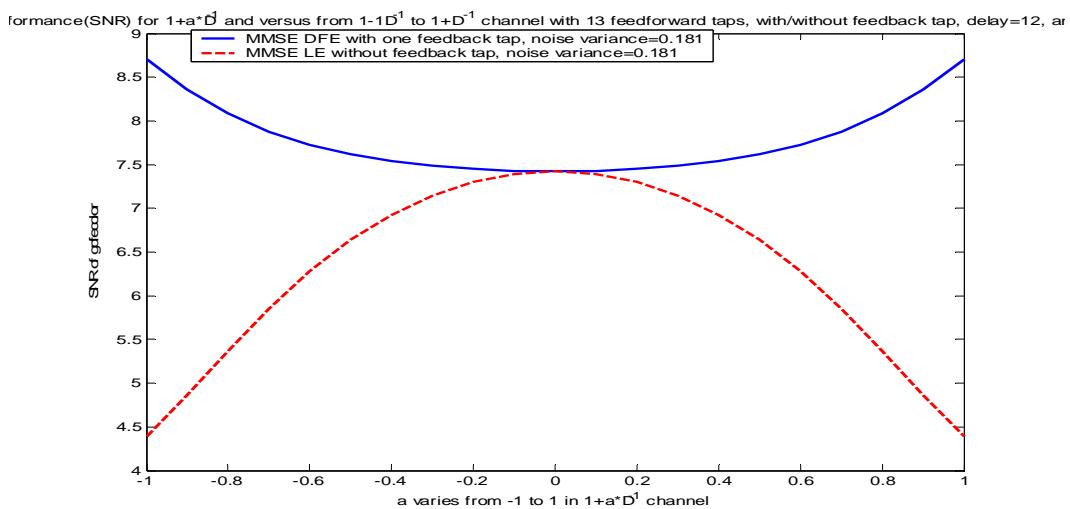
[3] Comparison between MMSE DFE and MMSE LE - FIR equalizer performance(SNR) for $1+a*D^{-1}$ versus varying from $1-1*D^{-1}$ to $1+1*D^{-1}$ channels with fixed SNRmfb/noise variance (comparing between MMSE DFE and MMSE LE)

A. Fix SNRmfb=10dB, Noise variance= $(1+a^2)/10 \leq [1+a*D^{-1}]$ channel
 'FIR equalizer performance(SNR) for $1+a*D^{-1}$ versus from $1-1*D^{-1}$ to $1+1*D^{-1}$ channels with 13 feedforward taps, with(MMSE DFE)/without(MMSE LE) feedback tap, and delay=12'



B. Fix noise variance=0.181

'FIR equalizer performance(SNR) for $1+a*D^{-1}$ versus from $1-1*D^{-1}$ to $1+1*D^{-1}$ channel with 13 feedforward taps, with(MMSE DFE)/without(MMSE LE) feedback tap, delay=12, and noise variance=0.181'



```

[4] function []= eye_eq(p,Ex,noise_var,eq_type);
%-----
% p      = pulse response [a*D^-1 + 1]==> [a 1]
% Ex     = average energy of signals, Ex_bar
% noise_var = noise variance
% eq_type = Z => ZERO FORCING
%           M => MMSE
%           D => MMSE-DFE
% outputs: pe_(zfe/mmle_dfe/mmse_le)= probability of error with Equalizer from N input
2PAM[+1/-1] data sequences
% outputs: pe_no_eq = probability of error without Equalizer
% outputs: dfseSNR = receiver(equalizer)SNR, unbiased in dB
% outputs: pe_SNR = error probability estimation from receiver(equalizer)SNR, Pe = Q
function of sqrt(dfseSNR)sdsds
% this function shows Frequency response of the channel p and equalizer filter, eye
diagram, receiver SNR, and probability of error.
% N = 250; % # of input 2 PAM[+1/-1] data, you can increase the N of input data
sequences in order to calculate the Pe accurately
% created 1/06 by Jungsub Byun and M. Malkin EE379A
%-----

```

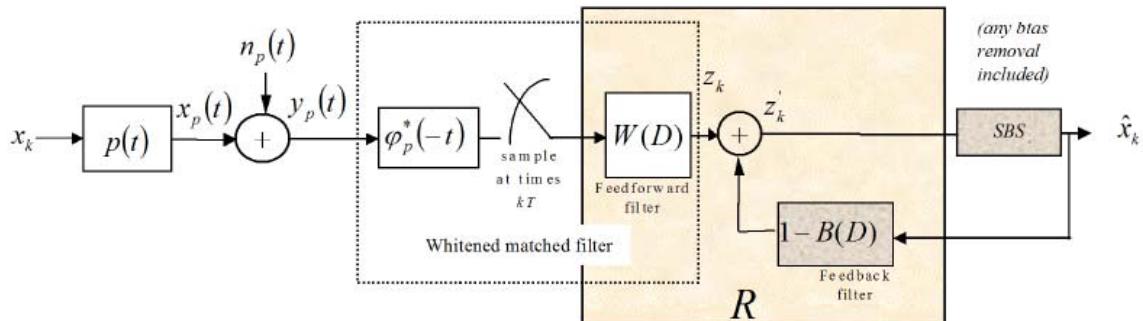
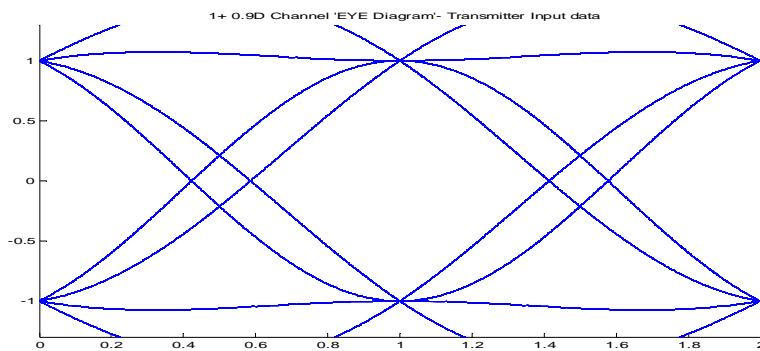


Figure 3.34: Decision feedback equalization.



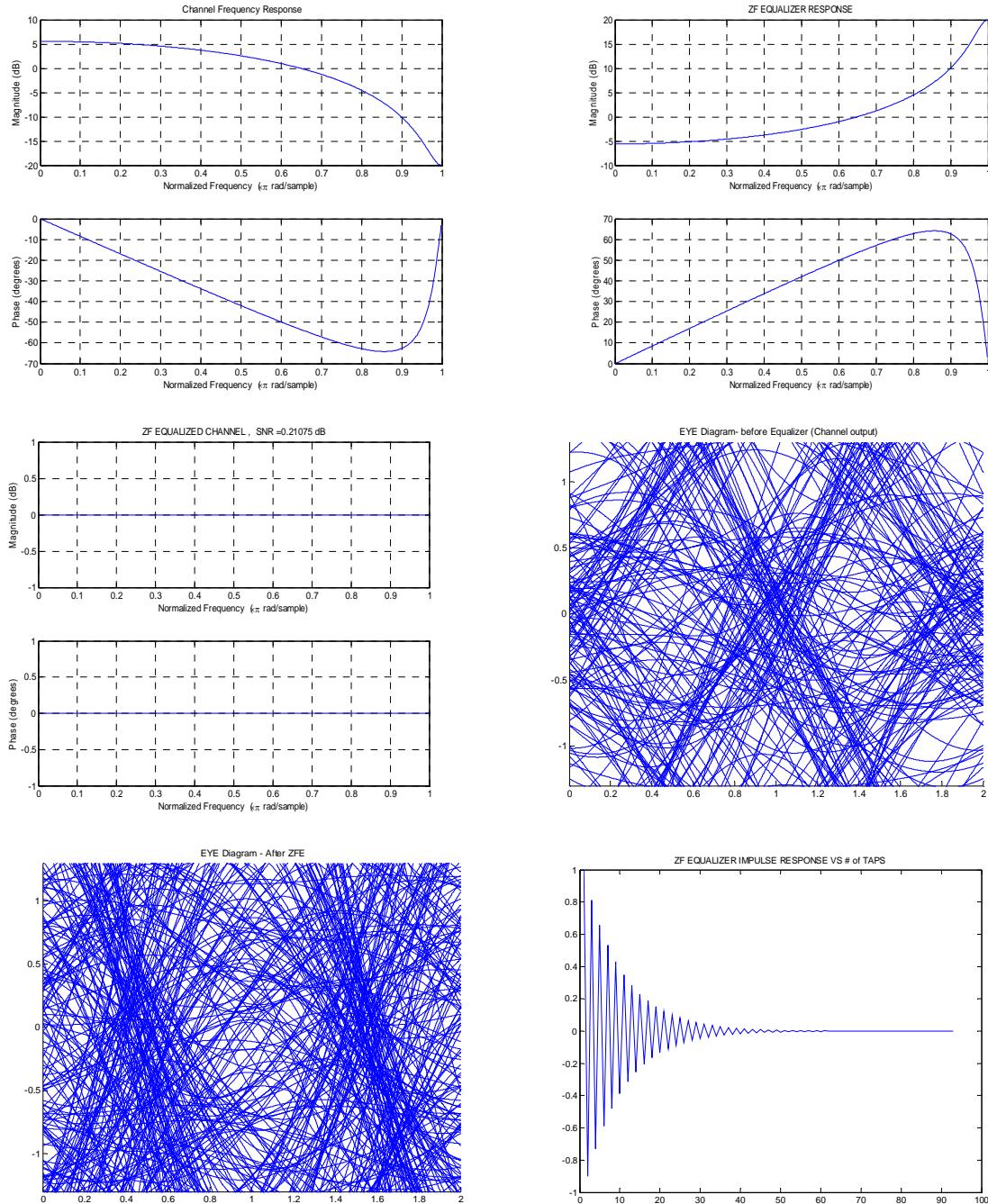
EXAMPLE 3.4.1, p176, $1+0.9*D^{-1}$ channel
eye_eq([.9 1],1,0.181,'z')

pe_zfe = 0.1200 probability of error with Equalizer from N input 2PAM[+1/-1] data sequences

pe_no_eq = 0.1793 probability of error without Equalizer

pe_SNR = 0.3231(error probability estimation from dfseSNR, Pe = Q function of sqrt(dfseSNR))

dfseSNR = 0.2108 receiver(equalizer)SNR, unbiased in dB



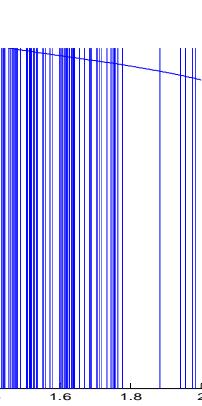
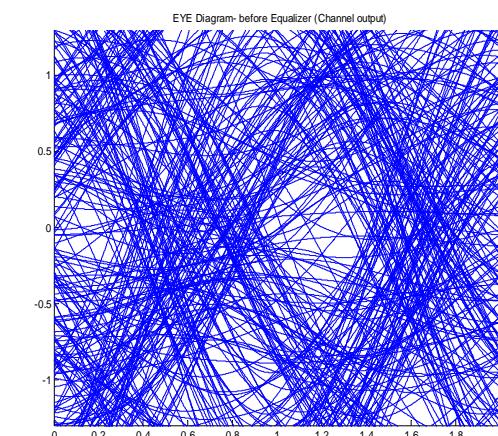
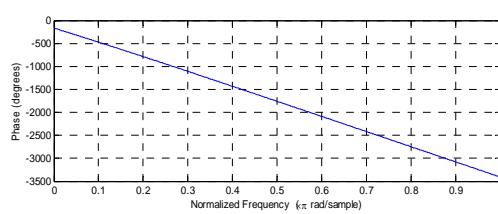
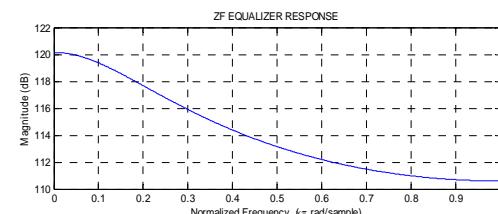
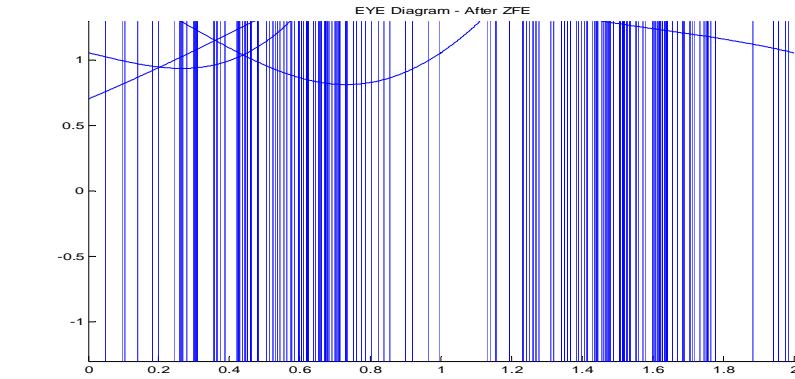
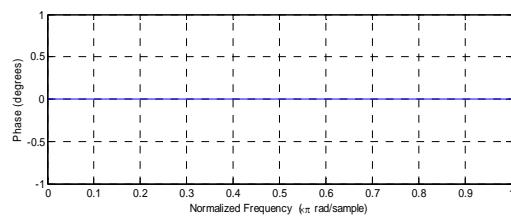
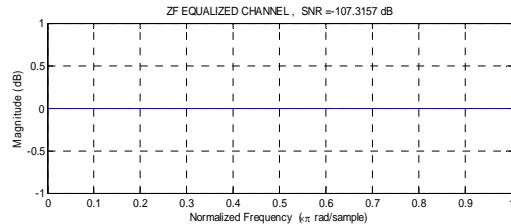
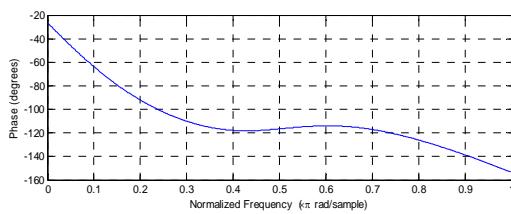
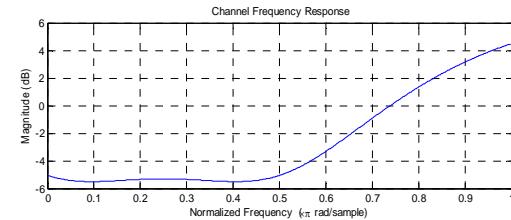
EXAMPLE 3.4.2, p180

`eye_eq([-0.5 1+0.25*i -0.5*i],1,0.15625,'z')`

`pe_zfe = 0.5175`

`pe_no_eq = 0.5657`

`dfseSNR = -107.3157`



EXAMPLE 3.5.1, p187

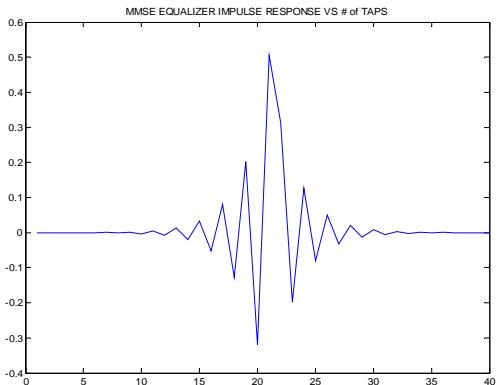
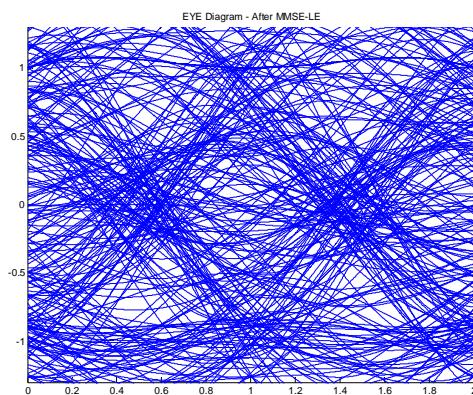
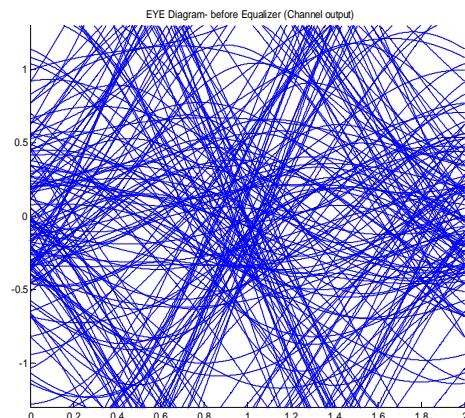
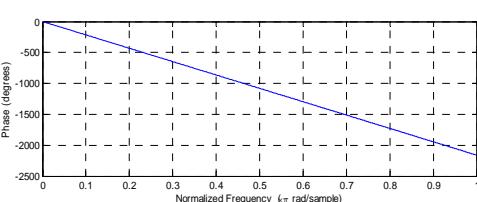
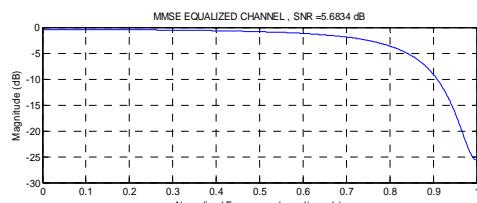
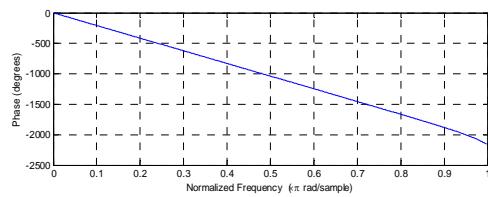
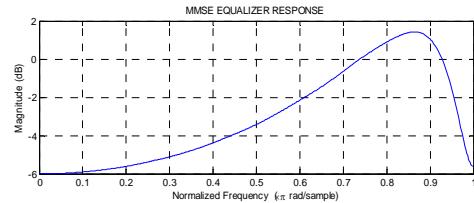
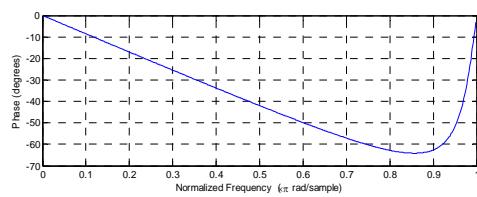
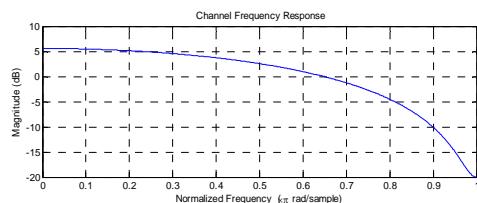
`eye_eq([.9 1],1,0.181,'m')`

`pe_mmse_le = 0.0155`

`pe_no_eq = 0.1768`

`pe_SNR = 0.0086`

`dfseSNR = 5.6834`



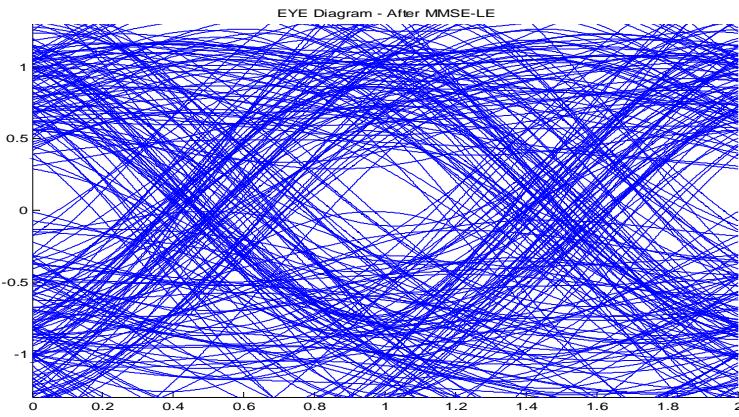
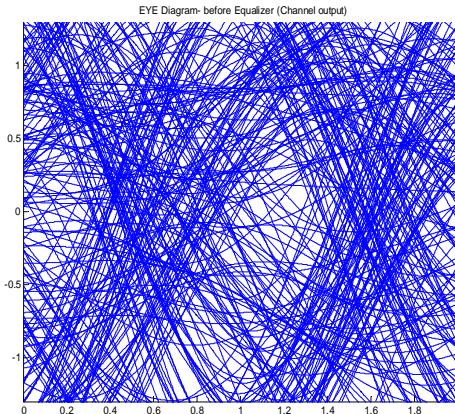
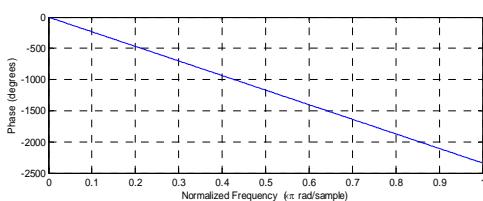
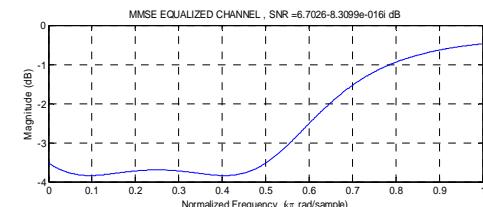
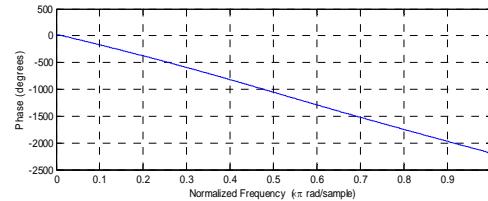
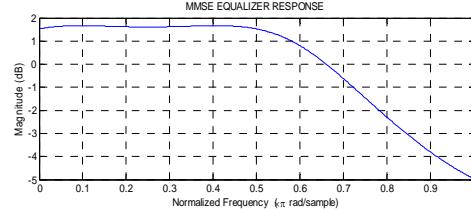
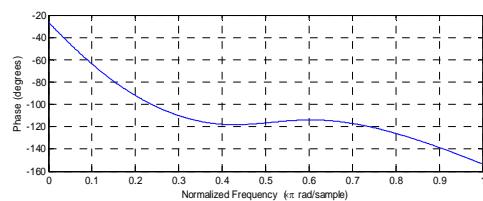
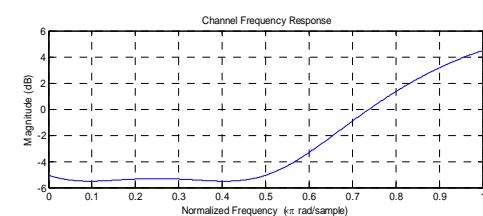
EXAMPLE 3.5.2, p 190

`eye_eq([-0.5 1+0.25*i -0.5*i],1,0.15625,'m')`

`pe_mmse_le = 0.0129`

`pe_no_eq = 0.5480`

`dfseSNR = 6.7026 - 0.0000i`

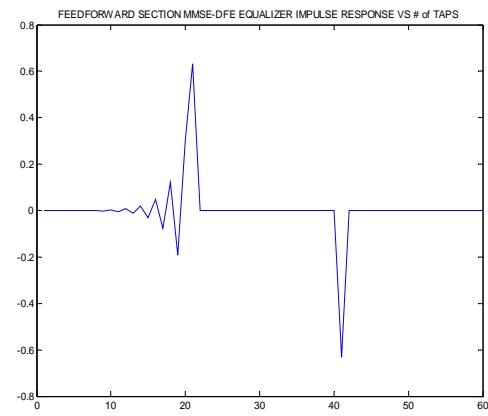
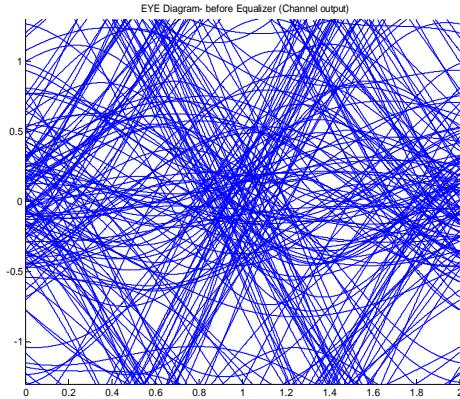
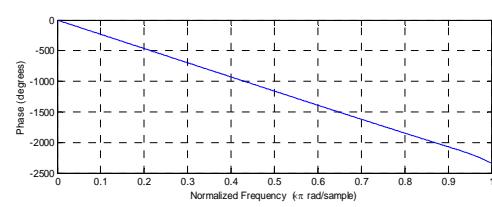
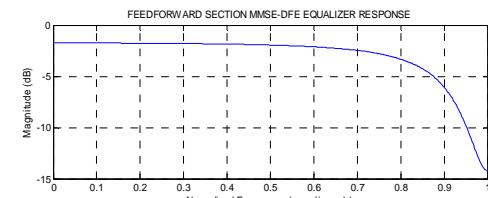
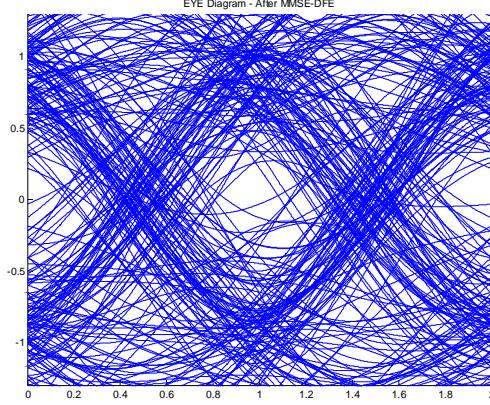
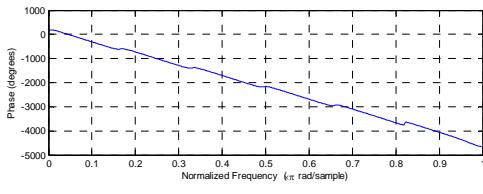
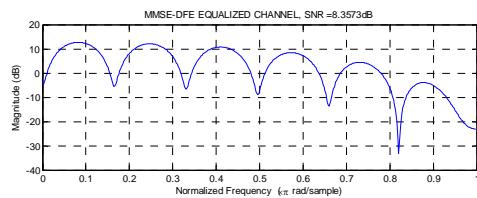
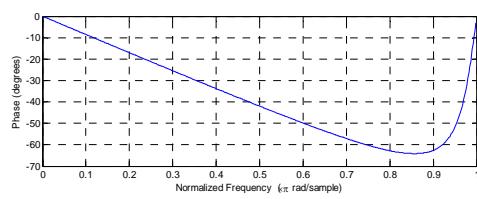
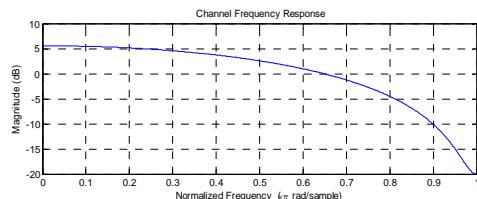


EXAMPLE 3.6.1, p 201

```

eye_eq([.9 1],1,0.181,'d')
pe_mmse_dfe = 0.0129
pe_no_eq = 0.2247
pe_SNR = 0.0019
dfseSNR = 8.3573

```



EXAMPLE 3.6.2, p 203

```

eye_eq([- .5 1+.25*i -.5*i],1,0.15625,'d')
pe_mmse_dfe = 0.0052
pe_no_eq = 0.5884
dfseSNR = 8.3665 - 0.0000i

```

