무선통신공학 HW4 - Equalization

2022284704 이동은

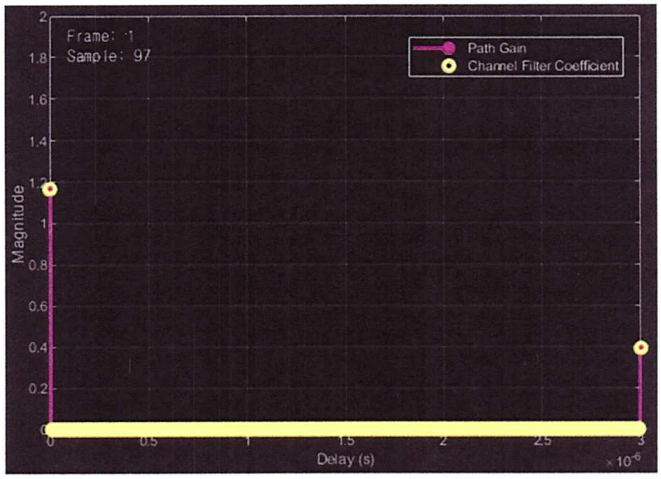
제출일: 2022.05.25

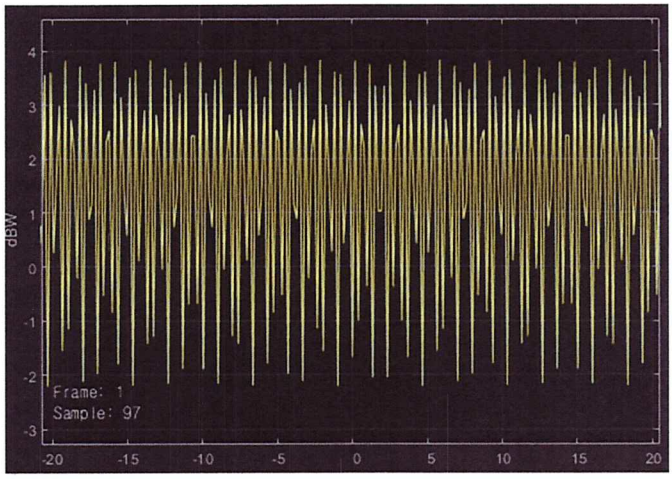
1. Equalizer

Equalizer는 multipath로 인해 frequency selective channel이 된 상황에서 channel의 frequency response를 flat하게 만들어주는 역할을 한다. 종류로는 ZF(zero-forcing) equalizer, MMSE(minimum mean square error) equalizer, ML(maximum likelihood) equalizer 등이 있다. 이 중 ZF와 MMSE equalizer를 설계하여 실제로 frequency selective한 채널을 frequency flat하게 만드는지 확인하였다.

2. 시뮬레이션 결과

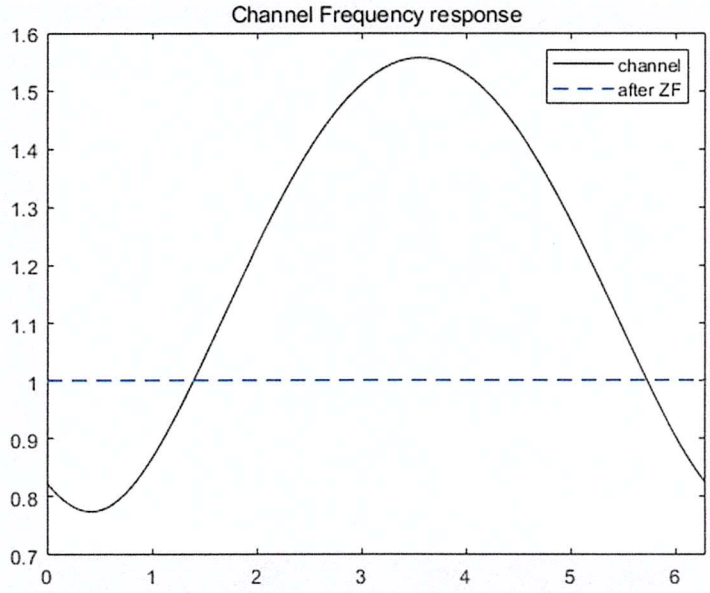
1) Multipath channel (Frequency selective channel)





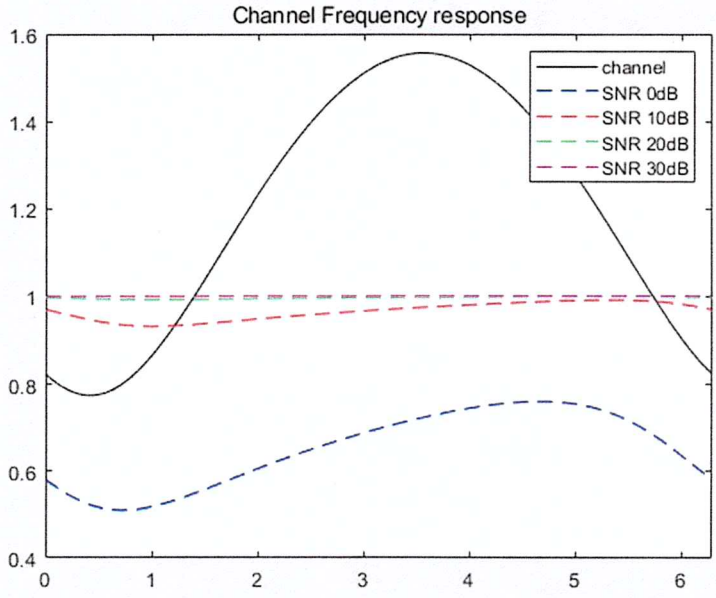
0s와 3μs의 two-path components를 갖는 multipath channel의 impulse response와 frequency response를 나타냈다. 이 채널은 frequency selective 채널을 갖는 것을 확인할 수 있다.

2) ZF equalizer



ZF equalizer를 통과한 channel frequency response이다. ZF equalizer는 frequency response의 inverse로 주어지기 때문에 이를 적용시켰을 때 frequency flat한 결과로 나타나는 것을 확인할 수 있다.

3) MMSE equalizer



ZF equalizer를 통과한 channel frequency response를 SNR에 따라 나타냈다. SNR이 높아짐에 따라 ZF equalizer 성능에 근접하는 결과를 보이는 것을 확인할 수 있다.

Appendix

```
% Multipath channel
fs = 1e8;
Ts = 1/fs;
N = 1024;
pathDelays = [0 300]*Ts; % two paths
avgPathGains = [0.775 -0.632]; % dB
rayleighchan = comm.RayleighChannel('SampleRate',fs, ...
    'PathDelays',pathDelays, ...
    'AveragePathGains',avgPathGains, ...
    'ChannelFiltering',false, ...
    'RandomStream','mt19937ar with seed', ...
    'Seed',15);
% 'Visualization','Impulse and frequency responses'
pathgains = rayleighchan();
gains = pathgains(1,:);
H = fft(gains,N);

% ZF equalizer
ZF_eq = H.^(-1);
after_ZF = H.*ZF_eq;

% MMSE equalizer
% signal power는 1W로 계산
SNR = [0, 10, 20, 30] ; % dB
N0 = 10.^(-SNR/10); % noise power

MMSE_eq0 = 1./(H+N0(1));
after_MMSE0 = H.*MMSE_eq0;
MMSE_eq10 = 1./(H+N0(2));
after_MMSE10 = H.*MMSE_eq10;
MMSE_eq20 = 1./(H+N0(3));
after_MMSE20 = H.*MMSE_eq20;
MMSE_eq30 = 1./(H+N0(4));
after_MMSE30 = H.*MMSE_eq30;

figure(1)
plot(0:2*pi/N:2*pi/N*(N-1),abs(H),'k')
hold on
plot(0:2*pi/N:2*pi/N*(N-1),abs(after_ZF),'b--')
legend('channel','after ZF')
xlim([0 2*pi])
title('Channel Frequency response')

figure(2)
plot(0:2*pi/N:2*pi/N*(N-1),abs(H),'k')
hold on
plot(0:2*pi/N:2*pi/N*(N-1),abs(after_MMSE0),'b--')
plot(0:2*pi/N:2*pi/N*(N-1),abs(after_MMSE10),'r--')
plot(0:2*pi/N:2*pi/N*(N-1),abs(after_MMSE20),'g--')
plot(0:2*pi/N:2*pi/N*(N-1),abs(after_MMSE30),'m--')
legend('channel','SNR 0dB','SNR 10dB','SNR 20dB','SNR 30dB')
xlim([0 2*pi])
title('Channel Frequency response')
```