**Demodulating OFDM Signals**

In this project, you are given an OFDM signal trace, which is 480 complex numbers, starting with a training symbol, followed by 5 data symbols. You are asked to learn the Channel State Information (CSI) from the training symbol, and obtain the data values modulated on the data symbols.

An OFDM symbol in this project is a vector of length 80, where the first 16 are the cyclic prefix, which similar to Wi-Fi. However, only the following 16 subcarriers are used:

2 6 10 14 18 22 26 30 34 38 42 46 50 54 58 62

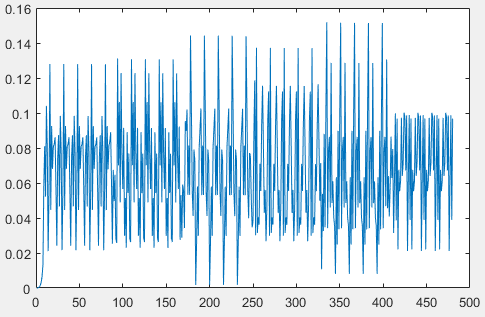
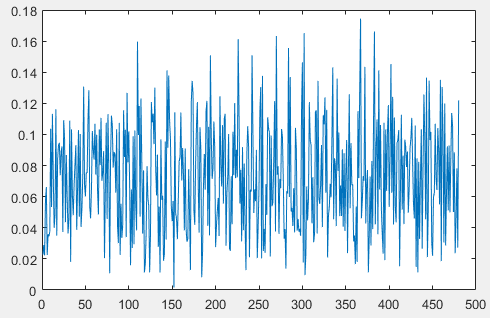
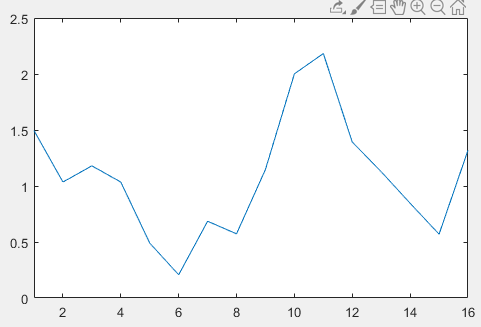
where the subcarrier index starts from 1. The modulation is BPSK. The values modulated on the training symbol subcarriers are

-1 1 -1 1 -1 1 -1 -1 -1 1 1 1 -1 1 1 -1

Recall that the CSI is the received value when the modulated data is 1.

The signal has been passed to the LTE ETU channel model, and then added with white Gaussian noise. The SNR is 30 dB. There is no CFO. You can also assume the first sample in the trace is the first sample of the first symbol.

The samples can be found in file “ofdm\_samples1” in “FILES” on Canvas. The following shows some information about it. Note that the CSI is highly non-uniform. The actual modulated data can be found in file “ofdm\_data1” in “FILES” on Canvas.

The Clean Signal The Received Signal The CSI

To demodulate the OFDM signal in this project, you would basically just need to take 64 consecutive samples and pass them to FFT. The output of FFT is 64 complex numbers, in which only 16 numbers (the subcarriers we use) are useful while the others are close to 0.

For your reference, the following is the code I wrote to generate the signal:

N = 64;

cplen = 16;

smbllen = cplen + N;

data\_subc\_idx = [2:4:N-1];

data\_smbl\_num = 5;

SNR = 30; % NOTE: in dB

SNR\_mul = 1/sqrt(power(10,SNR/10));

train\_val = [ -1 1 -1 1 -1 1 -1 -1 -1 1 1 1 -1 1 1 -1];

data\_val = [];

for h=1:data\_smbl\_num

data\_val = [data\_val; sign(rand(1,length(data\_subc\_idx))-0.5)];

end

tomodval = [train\_val; data\_val];

modsymbol = [];

for symidx=1:size(tomodval)

toifft = zeros(1,N);

toifft(data\_subc\_idx) = tomodval(symidx,:);

afterifft = ifft(toifft);

thissymbol = [afterifft(end-cplen+1:end),afterifft];

modsymbol = [modsymbol,thissymbol];

end

chcfg.DelayProfile = 'ETU';

chcfg.NRxAnts = 1;

chcfg.DopplerFreq = 5; % orig: 5

chcfg.MIMOCorrelation = 'Low';

chcfg.Seed = 1;

chcfg.InitPhase = 'Random';

chcfg.ModelType = 'GMEDS';

chcfg.NTerms = 16;

chcfg.NormalizeTxAnts = 'On';

chcfg.NormalizePathGains = 'On';

chcfg.SamplingRate = 1000000;

chcfg.InitTime = 0;

chcfg.Seed = ceil(rand(1)\*1000000);

cleansig = lteFadingChannel(chcfg, transpose(modsymbol)); cleansig = transpose(cleansig);

noise = (randn(1,length(cleansig)) + 1i\*randn(1,length(cleansig)))/sqrt(2)\*SNR\_mul;

outp = cleansig + noise;