COMM.SYS.300 COMMUNICATION THEORY

MATLAB EXERCISE #5

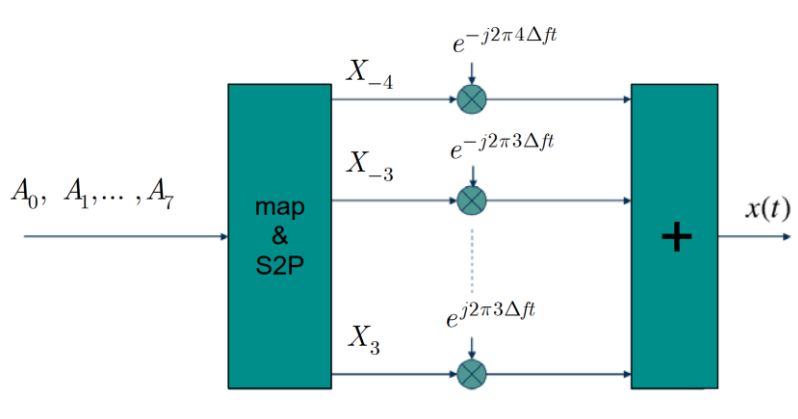
BASIC OFDM PRINCIPLES, SIGNAL PROPERTIES, CYCLIC PREFIX, DATA RATE

1 OFDM SIGNAL

Orthogonal Frequency Division Multiplexing is a multiplexing technique utilized in 4G networks. The data is modulated independently into N parallel subcarriers with rectangular pulse-shaping. To ensure the transmission quality, the signal has to be tightly time-limited.

1.1 OFDM BASICS AND SYMBOL GENERATION

The block transmission concept of the OFDM signal is shown in figure from the lecture notes below. The bits (A_0, A_1,…) are mapped into (usually QAM) symbols and mapped into the subcarriers (serial to parallel conversion). Each subcarrier is then multiplied by a different frequency with Δf frequency offset between neighboring subcarriers. The OFDM signal is then created by summing the individual subcarrier signals.



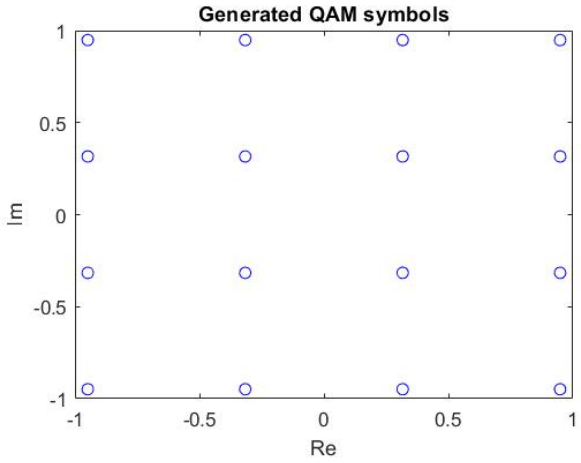
To implement this in MATLAB, define the following in the script:

```
>> Nsym = 50;                    % Number of OFDM symbols (in time)
>> Nsubcarr = 1000;              % 1000 subcarriers in band
>> Nactive = 600;                % 600 active subcarriers (which contain
data)
>> Modulation_order = 16;        % 16-QAM modulation = 4 bits per symbol
```

```
>> bits = randi([0 Modulation_order-1],Nactive,Nsym); %Generate bits at
random
>> QAMsymbols = qammod(bits,Modulation_order,'UnitAveragePower',true);
                                %M-QAM modulation of the bits (see help
qammod)
```

Next, **plot the created QAM symbols in complex plane**

- **label the axes**
- **show data points in blue color as circles**
- **the figure should look like this:**



The first symbol of the sequence will be a reference one, filled with ones.

```
% create 1 training symbol at the start of the sequence
>> training = ones(Nactive,1);
>> QAMsymbols(:,1) = training;
```

1.2 OFDM SYMBOL GENERATION, SPECTRUM

In the next part, let's map the QAM symbols into the subcarriers. First, define the following:

```
>> df = 15e3;                    % 15 kHz subcarrier spacing
>> Tsym = 1/df;                  % symbol duration in seconds
>> FFT_size = 1024;              % size of FFT
>> Fs = FFT_size*df;             % sampling frequency
>> Ts = 1/Fs;                    % sampling interval
>> Fc = 800e6;                   % 800 MHz carrier frequency
```

The choice of active subcarriers is described in the lecture notes. The active subcarriers should be the ones in the middle of the band, excluding the DC subcarrier. MATLAB considers one-sided spectrum initially (see Exercise 1). To properly work with the signal, the mapping is done as follows:

```
>> subcarrier_mapping = [QAMsymbols(1:Nactive/2,:);...
>>     zeros(FFT_size-(Nactive)-1,Nsym);...
>>     QAMsymbols(end-Nactive/2:end-1,:);...
>>     zeros(1,Nsym)];
```

- This way, after two-sided spectrum will correctly show the active subcarriers around 0 (DC subcarrier).

Now that the symbols are mapped into the active subcarriers, creating the OFDM signal is nothing else than Inverse Fourier Transform. To understand this step, it is necessary to realize that the symbol mapping was done in frequency, not in time as x-axis. Therefore, converting frequency-based data into time domain requires an IFFT operation (FFT size is defined above).

- **Convert the mapped symbols into an OFDM signal using *ofdm_symbols = ifft()* function with FFT size of 1024**

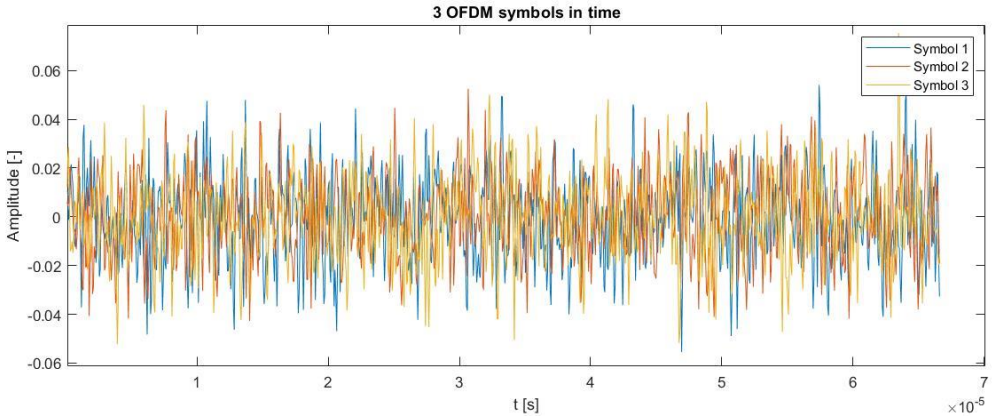
Next, to understand the OFDM better, plot the signal using the following code.

- **Fill in the BLANKs in the code so that the plot shows correctly**
- **The lines in graph indicate the frequency bands with active and inactive subcarriers**

```
>> figure;
>> OFDM_freq = BLANK_1(BLANK_2(ofdm_symbol(:),FFT_size*8));
>> freq_axis = -(Fs/2):Fs/length(OFDM_freq):Fs/2-Fs/length(OFDM_freq);
>> freq_axis = freq_axis/10^6;
>> bandwidth = Nsubcarr*df;
>> activebandwidth = Nactive*df;
>> plot(BLANK_3, 10*log10(OFDM_freq.*conj(OFDM_freq)))
>> xline(-bandwidth/2e6,'- r')
>> xline(bandwidth/2e6,'- r',{'OFDM bandwidth'})
>> xline(-activebandwidth/2e6,'- g')
>> xline(activebandwidth/2e6,'- g',{'active bandwidth'})
>> grid on
>> hold on
>> xlabel('frequency [MHz]')
>> ylabel('Amplitude [dB]')
>> title('Spectrum of OFDM signal')
```

Plot the OFDM symbols in time as well.

- **plot the first three OFDM symbols (excluding the training one) into the same plot**
- **use different color for each symbol**
- **don't forget title, data labels and legend**
- hint: Each symbol is represented by one column of *ofdm_symbol* variable. Plot only the real *real()* (or imaginary *imag()*) part of each symbol.



Questions:

- What does the OFDM signal look like in time and frequency? (remember Exercise 2, white and colored Gaussian noise)

- How is inter-symbol interference between neighboring subcarriers avoided?
- How can the interference between two consecutive OFDM symbols be eliminated?
- Is peak to average power ratio (PAPR) of the OFDM symbol high or low in comparison to single carrier systems?

2 CYCLIC PREFIX

To protect the information within the signal from the negative effects of the channel (multipath propagation, doppler effect, interference from subsequent symbols), a guard interval is inserted between every two symbols. This interval is either a zero-padding (sequence filled with zeroes), or a cyclic prefix (which is preferred in terms of spectrum).

The principle of cyclic prefix is to copy an end of each symbol (of the defined length) and place it to the beginning. Let's demonstrate the idea on a single OFDM symbol. The length of the cyclic prefix depends on the required guard interval, usually it corresponds to the maximum delay spread of the symbol. For better understanding, **reduce the number of active subcarriers to 20 for the following part**.

First, create a demo with a single symbol.

```
>> symbol = ofdm_symbol(:,2);          % pick 1st ODFM symbol
>> cp_length = 1/4*length(symbol);
                      % consider cp length of 1/4 of the symbol duration
>> cp = symbol(end-cp_length+1:end);   % copy the end part of the symbol
>> cp_symbol = [cp;symbol];
                      % add the end part of the symbol to the beginning
>> Tsym_cp = Tsym+1/4*Tsym;
                      % the duration of each symbol increases accordingly
```

- **Plot the symbol with the cyclic prefix as well as the original symbol.**
    - Why does adding the end of the symbol to the beginning help? (hint: consider a receiver with correlators)

Now apply the CP to all symbols. Return the number of active subcarriers to 600.

```
>> cp_length = 2e-6;  % cp length 2 microseconds
>> cp_length_samples = round(cp_length/Ts); % equal to 31 samples
>> cp = ofdm_symbol(end-cp_length_samples+1:end,:);
>> cp_ofdm_symbol = [cp;ofdm_symbol];
>> Tsym_cp = Tsym+Ts*cp_length_samples;

>> T_cp = [0:Ts:Tsym_cp-Ts]';    % Time vector for the symbols
>> T_nocp = [0:Ts:Tsym-Ts]';
```

3 DATA RATE

One of the main objectives of every communication system is achievable data rate. Implement the formula for calculating the theoretical data rate of the OFDM system in MATLAB. Consider the model from this exercise with 600 active subcarriers as reference.

$$Rbit = \frac{N \cdot log_2(M)}{T_{sym}} \; [bits\ per\ second]$$

N refers to the number of active subcarriers, M is the alphabet size and T_{sym} is the symbol duration in seconds.

- What is the achievable data rate in current setting?
- How does the data rate change in case the system uses CP of 1/4?
- How does the data rate change if modulation changes from 16-QAM to 256-QAM or BPSK?
- What is the maximum achievable data rate in case all subcarriers are active?

4 CHANNEL MODEL WITH MULTIPATH PROPAGATION

The channel model in this exercise will consist of several multipaths. First one is direct, line of sight path and others scattered. For this, Rayleigh Channel model is used. The last path is attenuated and with delay of 1.4 microseconds. The channel has strong frequency selectivity. The channel model is implemented directly in MATLAB's Communications Toolbox with visualization options.

```
>> delays = [0 1e-6 1.4e-6];
>> gains = [0 -1 -3];
>> Channel =
comm.RayleighChannel('SampleRate',Fs,'PathDelays',delays,'AveragePathGains'
,gains,'MaximumDopplerShift',0);
>> release(Channel);
>> Channel.Visualization = 'Impulse and frequency responses';
>> Channel.SamplesToDisplay = '10%';

>> Rx_symbols_cp = zeros(size(cp_ofdm_symbol)); % prepare an empty matrix
for received symbols
```

5 TRANSMISSION LOOP: TRANSMITTER – CHANNEL – RECEIVER

Now, the sequence is transmitted symbol-per-symbol through the channel and received at the Rx.

```
>> % instead of parallel to serial transformation, send symbols in a loop
>> for symb = 1:Nsym
>> Tx_ofdm_cp = cp_ofdm_symbol(:,symb);                % symbol by sybol
transfer
>> Tx_ofdm_cp = Tx_ofdm_cp.*exp(1j*2*pi*Fc*T_cp);      % move signal to the
carrier frequency band
>> Rx_ofdm_cp = step(Channel,Tx_ofdm_cp);
>> % Receiver design
>> Rx_ofdm_cp = Rx_ofdm_cp.*exp(-1j*2*pi*Fc*T_cp);     % Move bandpass
signal back to >> lowapss frequencies
>> Rx_symbols_cp(:,symb) = Rx_ofdm_cp; % Transform symbol-wise
>> end
```

After receiving the symbols, execute the following operations:

- Remove CP
- Fourier Transform of the received block
- Discard the empty subcarriers
- Equalize the received symbols using Zero-Forcing equalizer and training symbols

- Demodulate the QAM symbols

```
>> Rx_symbols = Rx_symbols_cp(cp_length_samples+1:end,:);        % Remove
CP
>> Rx_fft = fft(Rx_symbols,FFT_size);                  % DFT
>> Rx_nopadding = [Rx_fft(1:Nactive/2,:); Rx_fft(end-Nactive/2:end-1,:)]; %
select the active subcarriers
>> Channel_estimation = Rx_nopadding(:,1)./QAMsymbols(:,1);      % apply
Zero-Forcing channel equalizer
>> Equalizer = 1./Channel_estimation;
>> for k = 1:Nsym-1
>>     equalized_symbols(:,k) = Rx_nopadding(:,k+1).*Equalizer; % equalize
symbols
>> end
>> Rxbits =
qamdemod(equalized_symbols,Modulation_order,'UnitAveragePower',true); >>
%M-QAM demodulation
```

Plot the amplitude spectrum of the OFDM signal at the receiver. Compare the transmitter and receiver spectra to the channel response.

```
>> % plot the amplitude spectrum of the received OFDM signal
>> figure;
>> OFDM_freq = BLANK(BLANK(Rx_symbols(:),FFT_size*8));
>> freq_axis = -(Fs/2):Fs/length(BLANK):Fs/2-Fs/length(OFDM_freq);
>> freq_axis = freq_axis/10^6;
>> bandwidth = Nsubcarr*df;
>> activebandwidth = Nactive*df;
>> plot(freq_axis, 10*log10(OFDM_freq.*conj(OFDM_freq)))
>> xline(-bandwidth/2e6,'- r')
>> xline(bandwidth/2e6,'- r',{'OFDM bandwidth'})
>> xline(-activebandwidth/2e6,'- g')
>> xline(activebandwidth/2e6,'- g',{'active bandwidth'})
>> grid on
>> hold on
>> xlabel('frequency [MHz]')
>> ylabel('Amplitude [dB]')
>> title('Spectrum of the received OFDM signal')
```

To check whether the system works correctly, plot the transmitted and received symbols in complex plane.

```
>> figure
>> hold on;
>> plot(equalized_symbols(:,4),'rx');
>> plot(QAMsymbols(:,2:end),'bo');
>> xlabel('Re')
>> ylabel('Im')
>> title('Generated and received QAM symbols')
```

6 MODEL TESTING

Now the model should be perfectly functional. Transmitted and received symbols are perfectly aligned. In the next part, various parts of the system will be adjusted or removed to show their impact on the result.

Now, consider the model without equalizer. **Remove the equalization** from the code and plot the results from different subcarriers. The easiest way to do that is to redefine the equalizer to one.

```
>> Equalizer = 1;
```

Plot the received symbols from various subcarriers.

```
>> figure
>> subplot(2,2,1)
>> hold on;plot(equalized_symbols,'rx');plot(QAMsymbols,'bo');
>> xlabel('Re')
>> ylabel('Im')
>> title('Symbols of all subcarriers')
>> subplot(2,2,2)
>> hold
on;plot(equalized_symbols(3,:),'rx');plot(QAMsymbols(:,2:end),'bo');
>> xlabel('Re')
>> ylabel('Im')
>> title('Symbols of subcarrier 3')
>> subplot(2,2,3)
>> hold on;
>>
plot(equalized_symbols(15,:),'rx');plot(QAMsymbols(:,2:end),'bo');xlabel('R
e')
>> ylabel('Im')
>> title('Symbols of subcarrier 15')
>> subplot(2,2,4)
>> hold on;
>>
plot(equalized_symbols(20,:),'rx');plot(QAMsymbols(:,2:end),'bo');xlabel('R
e')
>> ylabel('Im')
>> title('Symbols of subcarrier 20')
```

Turn on the equalizer again and evaluate the effect of the CP length on the resulting signal. **Change the length of the Cyclic Prefix to 0, 0.5 microsecond and 1 microsecond**.

- Why is the signal distorted?
- What is the relation between CP length and delay spread of the channel? Try changing the channel delay and see what changes.