# **Lab 10**

# **Orthogonal Frequency Division Multiplexing (OFDM)**

**Objectives**

* Implementation of a basic OFDM transceiver in MATLAB

**Intro:**

OFDM, also called Discrete Multi-Tone (DMT), is a technique for mitigating the effects of poor channels. Multipath effects, interference, and impulsive noise, for example, characterize mobile communication channels. These effects can prevent high-speed communication (several megabits per second). In normal FDM systems, adjacent channels are typically separated by guard spaces - unused frequencies intended to prevent the channels from interfering with each other. This, obviously, is inefficient in terms of spectral utility.

An alternative to this is to make the adjacent channels orthogonal to one another. Each channel then carries a relatively low bit rate, and the channel sidebands can overlap and effective signaling can still occur. Although in a simplistic implementation, the number of carrier generators and coherent demodulators could get quite large; in fact, when viewed in the Fourier transform domain, the signal can be demodulated by taking the transform at the receiver. A completely digital implementation of the receiver is then possible by taking the Fourier transform of the received signal, as the signals at the transmitter are combined by taking the inverse Fourier transform.

OFDM is a particular type of FDM that can be viewed as a combination of modulation and multiple-access scheme. Whereas TDMA segments according to time, and CDMA segments according to spreading codes, like regular FDM, OFDM segments according to frequency as well. The spectrum is divided into a number of equally spaced bandwidths the centers of which contain carrier tones. A portion of each user's information is carried on each tone. Whereas regular FDM typically requires there to be frequency guard bands between the carrier bands so that mutual interference is minimized, OFDM allows the spectrum of each tone to overlap. The tones in OFDM are orthogonal with each other, and because of this orthogonality, they do not interfere with each other.

Following figure illustrates the spectral properties of an OFDM system with five channels. This frequency domain representation of five tones in this case highlights the orthogonal nature of the tones. The peak of each tone corresponds to a zero level, or null, of every other tone. The result of this is that there is no interference between tones. When the receiver samples at the tone frequency at the center of the channel, the only energy present is that of the desired signal, plus of course, whatever other noise happens to be in the channel.

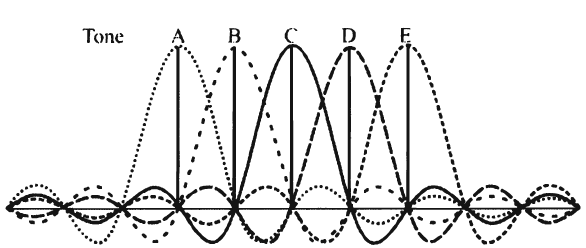


Figure 1: OFDM channels {A, B, C, D, E} each carry a relatively slow waveform but the signaling timing is such that they do not interfere with each other

The original data stream splits into *N* parallel data streams, each at a rate 1/ *N* of the original rate. Each stream is then mapped to a tone at a unique frequency and combined together using the *inverse fast Fourier transform* (lFFT) to yield the time-domain waveform to be transmitted. A block diagram indicating the significant functions in an OFDM transmitter is illustrated in the following figure. The cycle prefix insertion function is required to provide a guard time between symbols.

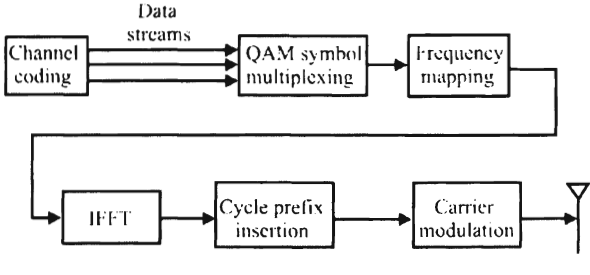


Figure 2: OFDM transmitter

By creating slower parallel data streams, the bandwidth of the modulation symbol is effectively decreased by a factor of N, or, due to the inverse relationship between bandwidth and symbol duration, the duration of the modulation symbol is increased by a factor of N.

Inter Symbol Interference (ISI) can be essentially eliminated, because typical multipath delay spread represents a much smaller proportion of the lengthened symbol time.

OFDM is also a multiple-access technique. This occurs when an individual tone or groups of tones are assigned to different users. The resultant structure is called Orthogonal Frequency Division Multiple Access (OFDMA).

To ensure orthogonality between tones, the symbol time must contain one or multiple cycles of each sinusoidal tone waveform. This is ensured by making the period of the tone frequencies integer multiples of the symbol period where the tone spacing is *1/T* as shown in the first figure. The following figure shows an example of three tones over a single symbol period, where each tone has an integer number of cycles during the symbol period.

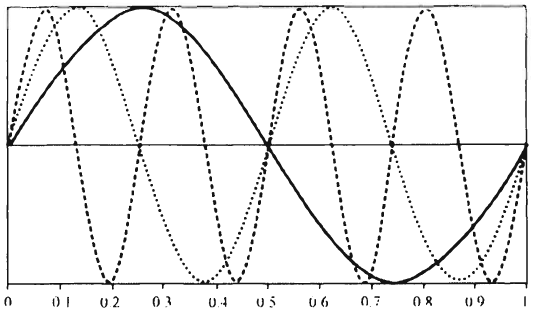
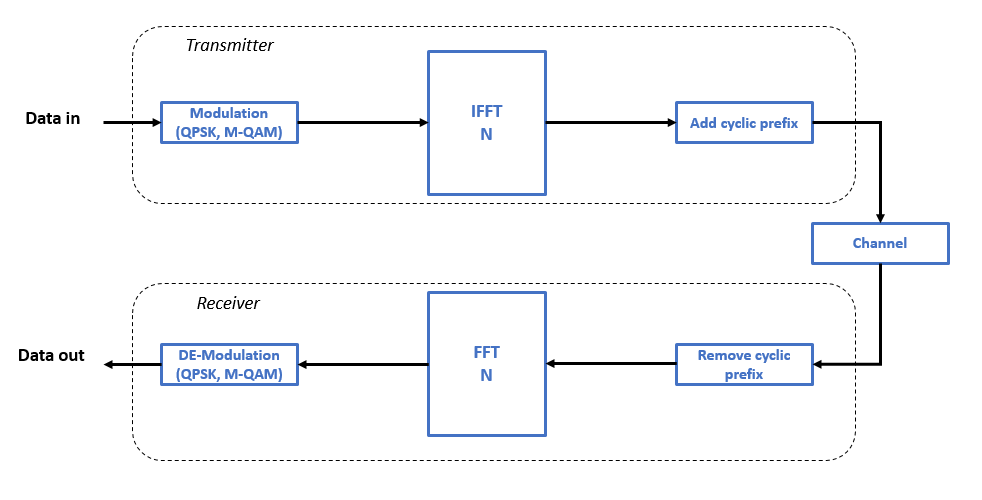


Figure 3: The number of sinusoid periods are integer multiples of each other. In this case there are three waveforms in the same period.

In this Lab we will implement the OFDM system following the steps in figure below:



B) Implementation OFDM transceiver in MATLAB

% Configuration Parameters

%% Setting Parameters

Subcarriers = 64; % total number of subcarrier (IFFT length equal to Subcarriers)

M = 16; % number of constellations

k = log2(M); % number of bits per constellation

numOfSym = 10^3; % number of OFDM Symbols

GI = 1/4; % Guard Interval or Cyclic Prefix, normaly 25 of the entire OFDM symbols

snr = 15; % Signal to noise ratio in dB

%% --------------------- TRANSMITER --------------------------------------

%%------------------------------------------------------------------------

%% Generate Data to be modulated on the subcarriers

TxData = randi([0,M-1], Subcarriers, numOfSym);

%% Implement QAM modulation

TxData\_Modulated = qammod(TxData,M);

%% Perform IFFT

TxData\_IFFT = ifft(TxData\_Modulated);

%% Adding cyclic Prefix

TxData\_GI = [TxData\_IFFT((1-GI)\*...

Subcarriers+1:end,:);TxData\_IFFT];

%% Plotting OFDM signal in time domain

[row , col] = size(TxData\_GI);

len = row\*col;

ofdm\_signal = reshape(TxData\_GI, 1, len);

figure(1);

plot(real(ofdm\_signal));

xlabel('Time');

ylabel('Amplitude');

title('OFDM Signal');

grid on;

%% Channel

rx\_signal = awgn(TxData\_GI ,snr,'measured');

%% --------------------- RECEIVER ----------------------------------------

%%------------------------------------------------------------------------

%% CP removal

Recieve\_GIremoved = rx\_signal(GI\*Subcarriers+1 : Subcarriers+GI\*Subcarriers, :);

%% FFT operation

RecieveData\_FFT = fft(Recieve\_GIremoved);

Signal\_Magnitude = real(RecieveData\_FFT);

Signal\_Phase = imag(RecieveData\_FFT);

%% plot the received constellations for a specific subcarrier

n = 4; % selected subcarrier

scatterplot(RecieveData\_FFT(n,:));

title('FFT Output 16-QAM');

%% Demodulation

RecieveData = qamdemod(RecieveData\_FFT,M);

%% Number of Bit Errors and Bit Error Rate computation

[num , BER] = biterr(TxData, RecieveData);

Exercise A

1. Add a for loop to use a vector of different SNR values.

Let SNR values vary from -20dB to +20 dB with a step size of 2dB by adding:

snrVec = -20:2:20;

Calculate the BER value (based on the simulated data) for each SNR and compare it with the theoretical value.

To calculate the theoretical BER value use the following commands:

EbNoVec = snr - 10\*log10(k); % in terms of energy per bit to noise power spectral density ratio (Eb/No)

ber\_theo(1, i) = berawgn(EbNoVec,'qam',M);

You will have the following output:



To plot you can use the following code:

figure

semilogy(snrVec, ber,'-ok', snrVec, ber\_theo);

grid;

ylabel('BER');

xlabel('SNR [dB]');

legend('AWGN-simulation', 'AWGN-theoretical');

Exercise B

In OFDM, multiple access can be achieved by assigning subsets of sub-carriers to different users.

Let us suppose that there are 4 users sharing the same spectrum and the Base station will assign 16 sub-carriers for each of the 4 users after divided the spectrum into 64 sub-carriers.

You must do the following tasks:

* Create a new script.
* After generating 64 subcarriers as we did in task 1, assign 16 sub-carriers for each user. For simplicity assign 1:16 subcarriers to user 1, 17:32 subcarriers to user 2, 33:48 subcarriers to user 3 and 49:64 subcarriers to user 4.
* Modulate the assigned subcarriers with different modulations: 4-QAM for user 1, 16-QAM for user 2, 64-QAM for user 3 and 256-QAM for user 4.
* Select number of OFDM symbols equal to 10^4.
* Use AWGN channel.
* Use a vector consisting of different SNR values from -20dB to +20 dB with a step size of 2dB.
* Plot the BER curve vs. SNR related to the four users (see the below figure to the left as an example).
* Compare between users in terms of BER (as shown in the figure to the right).

It is expected to get the following outputs:

