ECE 146B Lab: Modulation, demodulation and synchronization in OFDM

Lab Objectives: To develop a hands-on understanding of basic OFDM transmission and reception, including key concepts related to synchronization.

Reading: Section 8.3, Madhow (2014), plus the lab writeup.

Laboratory Assignment

Consider a discrete time complex baseband model in which a sequence of samples $\{s[n]\}$ sent by the transmitter give rise to a received signal

$$y[n] = (s * h)[n] + N[n] \ , \ n = 1, ..., M \tag{1}$$

The non-trivial part of the channel impulse response is of length L (the length is the difference between the earliest and latest nonzero entries in the impulse response, plus one), but we can model an additional delay of n_0 samples by inserting n_0 zeros at the beginning of the impulse response. For example, $h[n] = [0, 0, 1, 0, -0.5]$ is a channel of length $L = 3$, but the two zeros at the beginning model a delay of $\tau = 2$ samples.

We will be considering an OFDM system with N subcarriers and a cyclic prefix of P. In this lab, you will write code for general N and P, and demonstrate that the same code works for different values of N and P. Specifically, you will consider two example settings in this lab: (1) $N = 16$, $P = 4$ (helpful for debugging), (2) $N = 256$, $P = 16$ (to check that your program works for a larger set of subcarriers).

The receiver local oscillator (LO) is not *a priori* synchronized with the transmitter's LO, and there may be a Doppler shift due to relative motion between the transmitter and receiver. In discrete time, this may be modeled as a phase offset of $2\pi\delta/N$ per sample, where δ is the carrier offset *normalized to the subcarrier spacing of* $1/N$, as follows:

$$y[n] = e^{j2\pi\delta n/N}(s * h)[n] + N[n] \ , \ n = 1, ..., M \tag{2}$$

We organize the work in the lab as follows:

- OFDM modulation basics and peak-to-average ratio (PAR)
- Timing and frequency synchronization: these rely only on the structure of OFDM modulation
- Channel estimation: this uses known pilots
- Equalization (undoing the effect of the channel): this is done in the frequency domain after channel estimation
- System-level evaluation in the presence of noise
- (Optional) Learning about single carrier OFDMA and frequency domain equalization (often used on uplinks to reduce PAR)

Step 1 (OFDM modulation and time domain statistics): Write code implementing the basic OFDM modulator as follows.
1a) Generate N Gray coded QPSK symbols $\mathbf{B} = \{B[k], k = 1, .., N\}$. Take the inverse FFT to obtain time domain samples $\mathbf{b} = \{b[n], n = 1, ..., N\}$.
Remark: Matlab's *ifft* function implements:

$$b[n] = \frac{1}{N} \sum_{k=1}^{N} B[k] e^{j2\pi(n-1)(k-1)} \tag{3}$$

For BER simulations, you can use the function qpskmap developed in Software Lab 6.1 for this purpose. But for the development on synchronization in the first few parts, we can simply generate random $\{\pm 1 \pm j\}$ without keeping track of the bit-to-symbol map.

but if we want to normalize the complex exponential basis functions to unit norm, we could multiply this by $\sqrt{N}$.
1b) Append the last P time domain samples to the beginning, to get a length $N+P$ sequence of time domain samples $\mathbf{s}_1 = \{s[n], n = 1, ..., N+P\}$. That is, $s[1] = b[N-P+1], ..., s[P] = b[N], s[P+1] = b[1], ..., s[N+P] = b[N]$. We will term this an **OFDM symbol.**. Thus, we send N 2D symbols $\{B[k], k = 1, ..., N\}$ in a single OFDM symbol comprising $N+P$ samples.
1c) Concatenate K such OFDM symbols together (different sequences of 2D symbols are randomly generated and encoded into different OFDM symbols), where K is programmable. The number of time domain samples is therefore $K(N+P)$, and we denote this by $\mathbf{s}_K = \{s[n], n = 1, ..., K(N+P)\}$. Set $K = 10$ for the next few steps.
1d) Plot histograms of the real and imaginary parts of the time domain signal in 1(c). Do they look Gaussian?
1e) Compute the normalized correlation of the real and imaginary parts. Is it reasonable to model them as uncorrelated?
1f) Compute the peak-to-average power ratio of $\mathbf{s}_K$:

$$PAR = \frac{\max_n |s[n]|^2}{\overline{|s[n]|^2}} \tag{4}$$

Plot histograms of $PAR(dB) = 10\log_{10}(PAR)$ over multiple OFDM symbols generated as in 1(a)-(b) for $N = 16$ ($P = 4$) and for $N = 256$ ($P = 20$). Comment on the dependence on N.
Remark: We have defined the PAR for a block of K OFDM symbols. This is a random variable that depends on the randomly generated 2D symbols encoded in this block. In order to get a nice-looking histogram for this random variable, you will need to simulate over many such blocks, each with a different set of randomly generated 2D symbols.

We will assume that the 2D symbols in the first OFDM symbol are a known pilot sequence to be used for synchronization and channel estimation. These will then be used to demodulate the data sent in the remaining $K-1$ OFDM symbols, with the channel and synchronization parameters assumed to remain stable across the K OFDM symbols. In order to develop our ideas, let us focus on $K = 1$ for the next few steps.

Step 2 (Modeling and estimating delay) Even if the channel is ideal, the receiver does not know where the OFDM symbol starts. To model this, consider a channel

$$h = [zeros(\tau, 1); 1], \quad \textbf{pure delay} \tag{5}$$

where $\tau \geq 0$ is an integer. Set the frequency offset $\delta = 0$. Now, in (1), we obtain a signal y of length $N+P+\tau$, where the first τ entries are only noise. We would like to throw these entries away and then strip away the cyclic prefix. We do this by looking for a match between two segments of P samples spaced N samples apart, to estimate where the cyclic prefix is occurring. Compute the correlations between such segments for different hypothesized values of τ:

$$R[k] = \sum_{n=k+1}^{k+P} y^*[n]y[n+N] \ , k = 0, 1, ..., K \tag{6}$$

where $K + N + P = length(y)$. We can now estimate the best integer delay as

$$\hat{\tau} = \arg\max_{1 \leq m \leq M} |R[m]| \tag{7}$$

After doing this, we can strip away the cyclic prefix and get the following time domain received signal:

$$\hat{y}[n] = y[n + \hat{\tau} + P], \ n = 1, ..., N \tag{8}$$

The N-length received signal $\mathbf{Y}$ in the frequency domain is now given by

$$Y[k] = FFT(\hat{y}), \ k = 1, ..., N \tag{9}$$

Remark: Matlab's *fft* function implements

$$Y[k] = \sum_{n=1}^{N} \hat{y}[n] e^{-j2\pi(n-1)(k-1)} \ , \ k = 1, ..., N \tag{10}$$

You may multiply this by $1/\sqrt{N}$ if you want to normalize the complex exponential basis functions to unit norm.

2a) Implement the preceding procedures for estimating $\hat{\tau}$ and computing $\mathbf{Y} = (Y[1], ..., Y[N])^T$. Use $N = 16$, $P = 4$ to evaluate your code in the following.

2b) For a noiseless system with $\tau = 3$ check that you obtain $\hat{\tau} = 3$. Does a scatterplot of $\{Y[k], k = 1, ..., N\}$ correspond to a QPSK constellation.

2c) Now, suppose that $\tau = 3$. Manually set $\hat{\tau} = 5$ in (8) and display a scatterplot of $\{Y[k], k = 1, ..., N\}$. What does it look like? Can you explain what you are seeing?

3) Step 3 (Estimating and undoing small frequency offsets) Assume that the normalized frequency offset δ in (2) is in $(-0.5, 0.5)$. Set $N = 16$, $P = 4$, since we are still in debug mode.

3a) Argue that

$$\widehat{\delta} = \frac{\angle R[\hat{\tau}]}{2\pi} \tag{11}$$

provides an estimate for $\delta \in (-0.5, 0.5)$. Would this estimate work for larger values of δ (e.g., $\delta = 5.25$)?

3b) Consider a noiseless system with $\delta = 0.25$. Compare the estimate $\widehat{\delta}$ with the true value of δ.

3c) Undo the frequency offset using your estimate. That is, replace $y[n]$ by $y[n]e^{-j2\pi\delta n/N}$ and then strip away the cyclic prefix and compute the length N symbol $\mathbf{Y}$ as in (8)-9).

3d) What frequency offset in parts per million (ppm) does $\delta = 0.25$ correspond to for a carrier frequency of 2.4 GHz, a bandwidth of 20 MHz and $N = 16$ subcarriers? How does your answer change for a carrier of 28 GHz, a bandwidth of 100 MHz and $N = 4096$ subcarriers?

4) Step 4 (adding in a non-trivial channel) Now, consider a simple two-tap channel (together with a delay τ)

$$h = [zeros(\tau, 1); 1; -0.5], \quad \textbf{two} - \textbf{tap channel} \tag{12}$$

We are now going to check that the methods of Step 2 and Step 3 still apply for getting coarse timing synchronization and an estimate of normalized frequency offset in $(-0.5.0.5)$. This is because the periodicity in the transmitted stream induced by the cyclic prefix still persists when we pass it through an LTI system.

4a) For a noiseless system with $\tau = 3$, redo Step 2(b) (with zero frequency offset) and show that you can still get a good estimate of τ.

4b For a system with normalized frequency offset $\delta = 0.25$, redo Step 3(b) and show that you can still get a good estimate of δ.

4c) Undo the frequency offset using your estimate. That is, replace $y[n]$ by $y[n]e^{-j2\pi\delta n/N}$ and then strip away the cyclic prefix and compute the length N symbol $\mathbf{Y}$ as in (8)-9).

What has been accomplished so far: At this point, your code should be able to estimate and correct for a bulk delay τ and a small normalized frequency offset $\delta \in (-0.5, 0.5)$ for an OFDM symbol, ending up with a $N \times 1$ frequency domain observation $\mathbf{Y}$.

Now we need to estimate the channel: We have simply used the structure of the OFDM symbol in Steps 2-4 above. That is, we can do coarse timing synchronization and carrier synchronization (if the frequency offset is small enough) without needing a training sequence. We can do this even if in the presence of a nontrivial channel. However, demodulating the transmitted symbols requires that we obtain an estimate of frequency domain channel coefficients $\{H[k]\}$ in a model of the form

$$Y[k] = H[k]B[k] + \text{noise} \ , \ k = 1, ..., N \tag{13}$$

applied to the received vector obtained in (9). Once we obtain estimates $\{\widehat{H}[k]\}$ of the frequency domain channel, the symbols can be estimated by dividing by the channel estimated for each subcarrier (with adjustments to protect against dividing by small values):

$$\widehat{B}[k] = \frac{Y[k]}{\widehat{H}[k]} \ , \ k = 1, ..., N \tag{14}$$

Simple frequency domain channel estimation: Now, suppose that you know the symbols $\{B[k]\}$ in first OFDM symbol, and wish to use these to estimate the channel. Starting from the end of Step 4(c), after stripping away the cyclic prefix and undoing the frequency offset, we can estimate the frequency domain channel independently for each subcarrier using the known training sequence as follows:

$$\widehat{H}[k] = \frac{Y[k]}{B[k]} \ , \ k = 1, ..., N \tag{15}$$

You can now use these estimates to equalize the channel seen by the remaining OFDM symbols as in (14).

5) Step 5 (Testing simple frequency domain channel estimation) For K successive OFDM symbols as in Step 1, write code for estimating the channel based on the first OFDM symbol, and then using it to demodulate the remaining $K-1$ OFDM symbols.
5a) Run your code for debugging for $N=16$, $P=4$, $h=(0;0;1,-0.5)$, $\delta=0.25$ (you may set $\delta=0$ to start with), and nominal channel length $L=P=4$. Plot the channel amplitude and phase estimates against subcarrier index. Compare with the ideal plots based on the estimates of τ and δ that you obtained.
5b) For $K=2$, show a scatter plot of the symbol estimates (14) obtained for the second OFDM symbol using the channel estimate from the previous step.
5c) Redo 5(a) and 5(b) for $N=256$, $P=16$, $h=(0;0;0;0;1;0;-0.5;0.25j)$, $\delta=0.25$.

Step 6 (adding noise)
6a) In the setting of 5(c), go back to (2) and modify your code to add i.i.d. $CN(0,2\sigma^2)$ noise samples $N[n]$ in the time domain, choosing σ^2 as a function of a specified (programmable) E_b/N_0. Use an analytical estimate of E_b accounting for the cyclic prefix, the scaling you have adopted in the IFFT at the transmitter, and the channel, but also check it against simulation, where E_b is computed by taking the energy of the noiseless received signal and dividing it by the number of bits in the payload. For this computation, you may assume that the pilot symbols are also part of the payload.
Remark: You will need to simulate over many blocks of K OFDM symbols, each encoding independent randomly generated 2D symbols. The number of such blocks depends on the smallest BER you wish to estimate. How should you choose this?
6b) For a large E_b/N_0 of 20 dB, run your code for coarse synchronization, offset correction, channel estimation and demodulation, and check via a scatterplot that the equalized symbols look like a QPSK constellation.
6c) Can you improve the channel estimates by interpolating across subcarriers? Try some simple ideas and report on whether the scatterplots in Step 6(b) look better. If you obtain a good scheme, you can use it instead of the simple per-subcarrier estimates in (15).

Step 7 (BER simulations)
Now, set $K=10$, with the first OFDM symbol carrying known training symbols, and the remaining $K-1$ carrying payload. Allow the $E_b/N_0(sync)$ for the first synchronization symbol to be different from $E_b/N_0(payload)$ values for the payload symbols. Estimate the BER for payload data (using multiple runs), and plot it (on a log scale) against $E_b/N_0(payload)$ (dB) (over a range 5-20 dB, say). Plot three curves: (1) ideal QPSK, (2) Your system with $E_b/N_0(sync)$ of 30 dB, which should yield near-noiseless performance (3) Your system with $E_b/N_0(sync)$ of 12 dB. Comment on the relation between the curves.

Step 8 (Optional: introduction to frequency domain equalization for singlecarrier OFDMA): Given the high PAR for OFDM, cellular mobile uplinks often use singlecarrier OFDMA, in which each mobile might be allocated a different resource block, typically consisting of a block of subcarriers over a set of OFDM symbols. For example, suppose that a given mobile user is allocated N_u subcarriers $i, i+1, ..., i+N_u-1 \in [1, ..., N]$ over K symbols. Assuming that all mobile users arrive roughly at the same time at the base station (this is ensured through a base station feedback mechanism called timing advance), separate blocks of subcarriers will remain orthogonal, and the signals from different users will not interfere with each other. Let us try to illustrate this concept for a single user.
8a) Let us go back to Step 1 with $N=256$, $P=16$ and allocate subcarriers $1, ..., N_u$ ($N_u=16$) to a given user. Write code mapping singlecarrier symbols to the frequency domain as follows. The user generates Gray coded QPSK symbols $\{b_u[n], n=1,...,N_u\}$ (the lower case b indicates that these are time domain symbols). Using an N_u point FFT, we convert these into N_u frequency domain symbols $\{B_u[k], k=1,..,N_u\}$, and map these to the N_u allocated subcarriers. Within the overall set of N subcarriers, this user utilizes only the first N_u, setting the remainder to zero.
8b) Now, take an N-point inverse FFT to map to the time domain symbols $\mathbf{b}=\{b[n], n=1,...,N\}$ as in Step 1(a). Append P samples at the beginning as a cyclic prefix as usual. These are the samples $s[n], n=1,..,N+P$ sent by the user. Estimate the PAR as in Step 1(f).
8c) Try passing through a simple channel in a noiseless setting, doing frequency domain equalization on the N_u subcarriers assuming ideal channel knowledge, and then taking an N_u-point FFT to map back to the originally transmitted time domain symbols.
8c) Feel free to add in synchronization and estimation as for standard OFDM, and to add noise, as in Steps 1-7.

Lab Report: Your lab report should document the results of the preceding steps in order. Describe the reasoning you used and the difficulties you encountered.