ECE 146B/UCSB: Digital Communication
Lab 4: Introduction to OFDM

Reading: Chapters 4 and 6; background section in this lab assignment

Lab Objectives: To learn about Orthogonal Frequency Division Multiplexing (OFDM) for communication over dispersive channels.

Background

We have seen in Chapter 4 that, for an ideal channel, intersymbol interference (ISI) can be avoided in linearly modulated systems by designing the transmit and receive filters so that their cascade is Nyquist at the desired symbol rate; that is, symbols can be isolated from each other, provided that we sample at the appropriate times. However, since we typically do not have control over the channel, a Nyquist design still incurs ISI when the channel is dispersive. Lab 3 provided a hands-on introduction to equalization techniques for compensating such ISI. In this lab, we introduce an alternative approach to communication over dispersive channels whose goal is to isolate symbols from each other for any dispersive channel. The idea is to employ frequency domain transmission, sending symbols $B[n]$ using complex exponentials $s_n(t) = e^{j2\pi f_n t}$, which have two key properties:
P1) When $s_n(t)$ goes through an LTI system with impulse response $h(t)$ and transfer function $H(f)$, the output is a scalar multiple of $s_n(t)$. Specifically,

$$e^{j2\pi f_n t} * h(t) = H(f_n) e^{j2\pi f_n t}$$

P2) Complex exponentials at different frequencies are orthogonal:

$$\langle s_n, s_m \rangle = \int s_n(t) s_m^*(t) dt = \int_{-\infty}^{\infty} e^{j2\pi(f_n - f_m)t} dt = \delta(f_m - f_n) = 0 \ , \ f_n \neq f_m$$

This is analogous to the properties of eigenvectors of matrices. Thus, complex exponentials are *eigenfunctions* of any LTI system.

Conceptual basis for OFDM: For frequency domain transmission with symbol $B[k]$ modulating the complex exponential $s_n(t) = e^{j2\pi f_n t}$, the transmitted signal is given by

$$u(t) = \sum_n B[n] e^{j2\pi f_n t}$$

When this goes through a dispersive channel $h(t)$, we obtain (ignoring noise)

$$(u * h)(t) = \sum_n B[n] H(f_n) e^{j2\pi f_n t}$$

Note that the symbols $\{B[n]\}$ do not interfere with each other after passing through the channels, since different complex exponentials are orthogonal. Furthermore, regardless of how complicated the time domain channel $h(t)$ is, in the frequency domain, the problem of equalization has been parallelized: we only need to estimate and compensate for the complex scalar $H(f_n)$ in demodulating the nth symbol. We now discuss how to translate this concept into practice.

Finite signaling interval: The first step is to constrain the signaling interval, say to length T. The complex baseband transmitted signal is therefore given by

$$u(t) = \sum_{n=0}^{N-1} B[n] e^{j2\pi f_n t} I_{[0,T]}(t) = \sum_{n=0}^{N-1} B[n] p_n(t) \tag{1}$$

where $B[n]$ is the symbol transmitted using the modulating signal $p_n(t) = e^{j2\pi f_n t} I_{[0,T]}$, using the nth subcarrier at frequency f_n. Let us now see how the properties P1 and P2 are affected by time limiting. The time limited tone $p_n(t)$ has Fourier transform $P_n(f) = T\text{sinc}((f - f_n)T)e^{-\pi f T}$, which decays quickly as $|f - f_n|$ takes on values of the order of $\frac{k}{T}$. For a channel whose impulse response $h(t)$ is approximately timelimited to T_d (this is termed the channel delay spread), the transfer function is approximately constant over frequency intervals of length B_c roughly inversely proportional to $1/T_d$ (this is termed the channel coherence bandwidth). If the signaling interval is large compared to the channel delay spread ($T \gg T_d$, then $1/T$ is small compared to the channel coherence bandwidth ($\frac{1}{T} \ll B_c$), so that the gain seen by $P_n(f)$ is roughly constant, and the eigenfunction property is roughly preserved. That is, when $P_n(f)$ goes through a channel with transfer function $H(f)$, the output

$$Q_n(f) = H(f)P_n(f) \approx H(f_n)P_n(f) \tag{2}$$

Regarding the orthogonality property P2, two complex exponentials that are constrained to an interval of length T are orthogonal if the frequency separation is an integer multiple of $1/T$:

$$\int_0^T e^{j2\pi f_n t} e^{-j2\pi f_m t} dt = \frac{e^{j2\pi(f_n - f_m)T} - 1}{j2\pi(f_n - f_m)} = 0 \ , \quad \text{for } (f_n - f_m)T = \text{ nonzero integer}$$

Thus, if we wish to send N symbols in parallel using N *subcarriers* (the term used for each time-constrained complex exponential), we need a bandwidth of roughly N/T. When we pass the complex exponentials through a dispersive channel, this orthogonality is approximately subject to the approximation in (2), which says that the channel output is approximately $H(f_n)e^{j2\pi f_n t} I_{[0,T]}(t)$ for the nth timelimited tone.

DSP-centric implementation: The proliferation of OFDM in commercial systems (including wireline DSL, wireless local area networks and wireless cellular systems) has been enabled by the implementation of its transceiver functionalities in digital signal processing, which leverages the economies of scale of digital computation (Moore's "law"). For T large enough, the bandwidth of the OFDM signal u is approximately N/T, where N denotes the number of subcarriers. Thus, we can represent $u(t)$ accurately by sampling at rate $1/T_s = N/T$, where T_s is the sampling interval. From (1), the samples are given by

$$u(kT_s) = \sum_{n=0}^{N-1} B[n] e^{j2\pi nk/N}$$

We can recognize this simply as the inverse DFT of the symbol sequence $\{B[n]\}$. We make this explicit in the notation as follows:

$$b[k] = u(kT_s) = \sum_{n=0}^{N-1} B[n] e^{j2\pi nk/N} \tag{3}$$

If N is a power of 2 (which can be achieved by zeropadding if necessary), the samples $\{b[k]\}$ can be efficiently generated from the symbols $\{B[n]\}$ using an inverse Fast Fourier Transform (IFFT). The complex baseband waveform $u(t)$ can now be obtained from its samples by digital-to-analog (D/A) conversion. This implementation of an OFDM transmitter is as shown in Figure 1: the bits are mapped to symbols, the symbols are fed in parallel to the inverse FFT (IFFT) block, and the complex baseband signal is obtained by D/A conversion of the samples (after insertion of a cyclic prefix, to be discussed after we motivate it in the context of receiver implementation). Typically, the D/A converter is an interpolating filter, so that its effect can be subsumed within the channel impulse response.

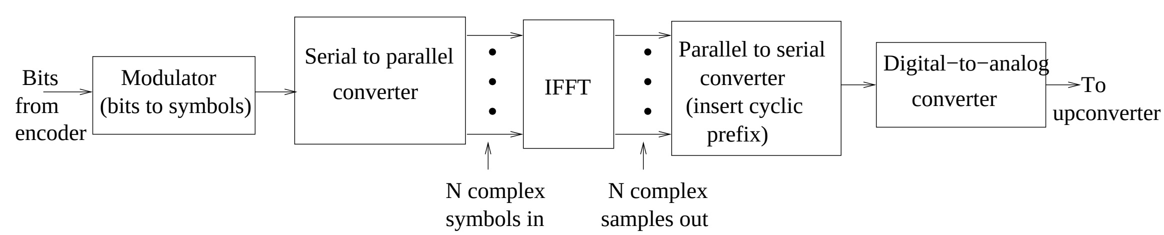

Figure 1: DSP-centric implementation of an OFDM transmitter.

Note that the relation (3) can be inverted as follows:

$$B[n] = \frac{1}{N} \sum_{k=0}^{N-1} b[k] e^{-j2\pi nk/N} \tag{4}$$

This is exploited in the digital implementation of the OFDM receiver, discussed next.

Remark on Matlab FFT and IFFTconventions: Matlab puts a factor of $1/N$ in the IFFT rather than in the FFT as done in (3) and (4). In both cases, however, IFFT followed by FFT gives the identity. Note also that Matlab numbers vector entries starting with one, so the FFT of $x[n], n = 1, ..., N$ is given by:

$$X[k] = \sum_{n=1}^{N} x[n] e^{j2\pi(n-1)(k-1)/N}$$

The corresponding IFFT is given by

$$x[n] = \frac{1}{N} \sum_{k=1}^{N} X[k] e^{j2\pi(n-1)(k-1)/N}$$

We know that, once we limit the signaling duration to be finite, the ISI avoidance property of OFDM is approximate rather than exact. However, as we now show, orthogonality between subcarriers can be restored *exactly* in discrete time by using a cyclic prefix, which allows for efficient demodulation using an FFT. The noiseless received OFDM signal is modeled as

$$v(t) = \sum_{k=0}^{N-1} b[k] p(t - kT_s)$$

where the "effective" channel impulse response $p(t)$ includes the effect of the D/A converter at the transmitter, the physical channel, and the receive filter. When we sample this signal at rate $1/T_s$, we obtain the discrete-time model

$$v[m] = \sum_{k=0}^{N-1} b[k] h[m - k] \tag{5}$$

where $\{h[l] = p(lT_s)\}$ is the effective discrete time channel of length L, assumed to be less than N. We assume, without loss of generality, that $h[l] = 0$ for $l < 0$ and $l \geq L$. We can rewrite (5) as

$$v[m] = \sum_{l=0}^{L-1} h[l] b[m - l] \tag{6}$$

Let H denote the N point DFT of h:

$$H[n] = \sum_{l=0}^{N-1} h[l] e^{-j2\pi nl/N} = \sum_{l=0}^{L-1} h[l] e^{-j2\pi nl/N} \tag{7}$$

As noted in (4), the DFT of $\{b[k]\}$ is the symbol sequence $B[n]$ (the normalization is chosen differently in (4) and (7) to simplify the forthcoming equations.) If we could replace the linear convolution of (6) by the circular convolution $\tilde{v} = h \odot b$ given by

$$\tilde{v}[m] = (h \odot b)[m] = \sum_{l=0}^{N-1} h[l \bmod N] b[(m - l) \bmod N] \tag{8}$$

then the corresponding N-point DFTs would satisfy

$$\tilde{V}[n] = H[n] B[n] \ , \quad n = 0, ..., N - 1$$

(The proof is omitted.) Since $L < N$, we can write the circular convolution (8) as

$$\tilde{v}[m] = \sum_{l=0}^{\min(L-1,m} h[l]b[m-l] + \sum_{l=m+1}^{L-1} h[l]b[m-l+N] \tag{9}$$

Comparing the linear convolution (6) and the cyclic convolution (9), we see that they are identical *except* when the index $m-l$ takes negative values: in this case, $b[m-l] = 0$ in the linear convolution, while $b[(m-l)\mathrm{mod}N] = b[m-l+N]$ contributes to the circular convolution. Thus, we can emulate a cyclic convolution using the physical linear convolution by sending a *cyclic prefix*; that is, by sending

$$b[k] = b[N+k], \quad k = -(L-1), -(L-2), ..., -1$$

before we send the samples $b[0], ..., b[N-1]$. That is, we transmit the samples

$$b[N-L+1], ..., b[N-1], b[0], ..., b[N-1]$$

incurring an overhead of $(L-1)/N$ which can be made small by choosing N to be large.

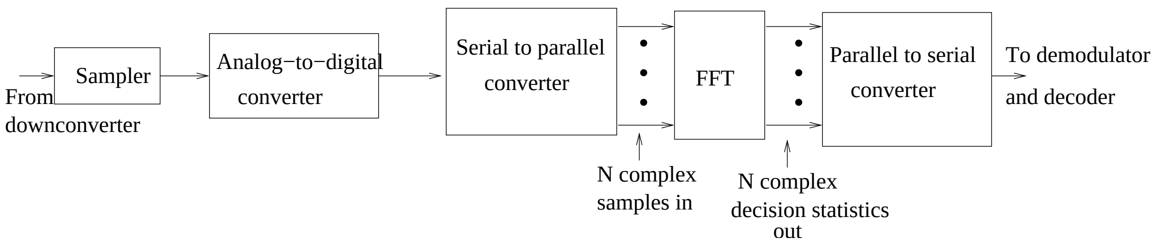

Figure 2: DSP-centric implementation of an OFDM receiver. Carrier and timing synchronization blocks are not shown.

At the receiver, the complex baseband signal is sampled at rate $1/T_s$ to obtain noisy versions of the samples $\{b[k]\}$. The FFT of these samples then yields the model

$$Y[n] = H[n]B[n] + N[n] \tag{10}$$

where the frequency domain noise samples $N[n]$ are modeled as i.i.d. $CN(0, 2\sigma^2)$, being the DFT of i.i.d. $CN(0, 2\sigma^2)$ time domain noise samples. If the receiver knows the channel, then it can implement ML reception based on the statistic $H^*[n]Y[n]$. Thus, the task of channel equalization has been reduced to compensating for scalar channel gains for each subcarrier. This makes OFDM extremely attractive for highly dispersive channels, for which time-domain singlecarrier equalization strategies would be difficult to implement.

Channel estimation: Channel estimation (along with timing and carrier synchronization, which are not considered here) is accomplished by sending pilot symbols. In this lab, we will send an entire OFDM symbol as a pilot, followed by a succession of other OFDM symbols with payload.

Laboratory Assignment

We would like to leverage the code from Lab 3 as much as possible, so we will pick the DAC filter to be the transmit filter in Lab 3, and the receive filter to its matched filter as before. The main difference is that the time domain samples sent through the DAC filter are obtained by taking the inverse FFT of the frequency domain symbols, and inserting a cyclic prefix. We will be fixing the constellation as Gray coded QPSK.

Step 1 (Exploring time and frequency domain relationships in OFDM): Let us first discuss the structure of a single "OFDM symbol," which carries N complex-valued symbols in the frequency domain. Here N is the number of subcarriers, chosen to be a power of 2. Set L to be length of the cyclic prefix. Set $N = 256$, $L = 20$ for these initial explorations, but keep the parameters programmable for later use.
1a) Generate N Gray coded QPSK symbols $\mathbf{B} = \{B[k], k = 1, .., N\}$. Take their inverse FFT to obtain time domain samples $\mathbf{b} = \{b[n], n = 1, ..., N\}$.

1b) Append the last L time domain samples to the beginning, to get a length $N+L$ sequence of time domain samples $\mathbf{b}' = \{b'[n], n = 1, ..., N+L\}$. That is, $b'[1] = b[N-L+1], ..., b'[L] = b[N], b'[L+1] = b[1], ..., b'[N+L] = b[N]$.
1c) Take the first N symbols of $\mathbf{b}'$, say $\mathbf{r}_1 = \{b'[1], ..., b'[N]\}$. Show that the FFT output (say $\mathbf{R}_1 = \{R_1[k]\}$) is related to the original frequency domain symbols $\mathbf{B}$ through a frequency domain channel $\mathbf{H}$ as follows: $R_1[k] = H[k]B[k]$. Find and plot the amplitude $|H[k]|$ and phase $arg(H[k])$ versus k.
1d) Repeat 1c) for the time domain samples $\{b'[3], ..., b'[N+3]\}$ (i.e., skip the first two samples of $\mathbf{b}'$. How are the frequency domain channels in 1c) and 1d) related? (What we are doing here is exploring what cyclic shifts in the time domain do in the frequency domain.)

Step 2 (generating multiple OFDM symbols): Now, we will generate K frames, each carrying N Gray coded QPSK symbols. Set $N = 256, L = 20, K = 5$ for numerical results and plots in this step.
2a) For each frame, generate time domain samples and add a cyclic prefix, as in Steps 1a) and 1b). Then, append the time domain samples for successive frames together. We now have a stream of $K(N+L)$ time domain samples, analogous to the time domain symbols sent in Lab 3.
2c) Pass the time domain symbols through the same transmit filter (this is the DAC in Figure 1) as in Labs 1-3, again oversampling by a factor of 4. (That is, if the time domain samples are at rate $1/T_s$, the filter is implemented as rate $4/T_s$. This gives us a rate $4/T_s$ transmitted signal.
2d) Compute the peak to average power ratio (PAPR) in dB for this transmitted signal (OFDM is notorious for having a large PAPR). This is done by taking the ratio of the maximum to average value of the magnitude squared of the time domain samples.
2e) Note that the original QPSK symbols in the frequency domain have a PAPR of one, but the time domain samples are generated by mixing these together. The time domain samples could be expected, therefore, to have a Gaussian distribution, invoking the central limit theorem. Plot a histogram of the I and Q components from the time domain samples. Do they look Gaussian?
2f) As in Lab 2, assume an ideal channel filter and pass the transmitted signal through a receive filter matched to the transmit filter. This gives a rate $4/T_s$ noiseless received signal.
2g) Subsample the received signal at rate $1/T_s$, starting with a delay of d samples (play around and see what choice of d works well–perhaps based on the peak of the cascade of the transmit, channel and receive filter). The first N samples corresponding to the first frame. Take the FFT of these N samples to get $\{R_1[k]\}$. Now, estimate the frequency domain channel coefficients $\{H[k]\}$ by using the known transmitted symbols $B_1[k]$ in the first frame as training. That is,

$$\hat{H}[k] = R_1[k]/B_1[k]$$

Plot the magnitude and phase of the channel estimates and comment on how it compares to what you saw in 1c) and 1d).
2h) Now, use the channel estimate from 2g) to demodulate the succeeding frames. If frame m uses time domain samples over a window $[a, b]$, then frame $m+1$ uses time domain samples over a window $[a+(N+L)T_s, b+(N+L)T_s]$. Denoting the FFT of the time domain samples for frame m as $R_m[k]$, the decision statistics for the frequency domain symbols for the mth frame are given by

$$\hat{B}_m[k] = \hat{H}^*[k]R_m[k]$$

You can now decode the bits and check that you get a BER of zero (there is no noise so far). Also, display scatter plots of the decision statistics to see that you indeed are seeing a QPSK constellation after compensating for the channel.

Step 3 (Channel compensation): We now introduce a nontrivial channel (still no noise). Increase the cyclic prefix length if needed (it should be long enough to cover the cascade of the transmit, channel and receive filters. But remember that the cyclic prefix is at rate $1/T_s$, whereas the filter cascade is at rate $4/T_s$.
3a) Repeat Step 2f, but now with a nontrivial channel filter modeled at rate $4/T_s$. Use the channels you have tried out in Lab 3 (still no noise). For example:

$$\text{channel\_filter} = [-0.7, -0.3, 0.3, 0.5, 1, 0.9, 0.8, -0.7, -0.8, 0.7, 0.8, 0.6, 0.3]';$$

3b) Repeat Step 2g. Comment on how the magnitude and phase of the frequency domain channel differs from what you saw in 2g, 1c and 1d.
3c) Repeat Step 2h. Check that you get a BER of zero, and that your decision statistics give nice QPSK scatter plots.
3d) Check that everything still works out as you vary the number of subcarriers N (e.g., $N = 512, 1024, 2048$), the cyclic prefix length L and the number of frames K.

Step 4 (Effect of noise): Now, add noise as in Labs 2 and 3. Specifically, at the input to the receive filter, add independent and identically distributed (iid) complex Gaussian noise, such that the real and imaginary part of each sample are iid $N(0, \sigma^2)$ (you will choose $\sigma^2 = \frac{N_0}{2}$ corresponding to a specified value of $\frac{E_b}{N_0}$. Let us fix $N = 1024$ for concreteness, and set the cyclic prefix to just a little longer than the minimum required for the channel you are considering. Set the number of frames to $K = 10$. Try a couple of values of E_b/N_0 of 5 dB and 8 dB.
4a) While you can estimate E_b analytically, estimate it by taking the energy of the transmitted signal in 3a, and dividing it by the number of bits in the payload (i.e., excluding the first frame). Use this to set the value of N_0 for generating the noise samples.
4b) Pass the (rate $4/T_s$) noise samples through the receive filter, and add the result to the output of part 3a.
4c) Consider first a noiseless channel estimate, in which you carry out Step 3b (estimating the channel based on frame 1) *before* you add noise to the output of 3a. Now add the noise and carry out Step 3c (demodulating the other frames). Estimate the BER and compare with the analytical value for ideal QPSK. Show the scatter plots of the decision statistics.
4d) Repeat 4c, except that you now do the channel estimate based on frame 1 after adding noise. Discuss how the BER degrades. Compare the channel estimates from parts 4c and 4d on the same plot.
Note: You will notice a significant BER degradation, but that is because the channel estimation technique is a bit naive (the channel coefficients for different subcarriers are highly correlated, and we are not exploiting this). Exploring better channel estimation techniques is beyond the scope of this lab, but you are encouraged to browse the literature on OFDM channel estimation to dig deeper.

Step 5 (Consolidation): Once you are happy with your code, plot the BER (log scale) as a function of E_b/N_0 (dB) for the channel in Lab 3. Plot three curves: ideal QPSK, OFDM with noiseless channel estimation, OFDM with noisy channel estimation. Comment on the relation between the curves.

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.