Today: (1) OFDM, (2) Fading Demo

- OH Thu 10-noon (me), Fri 9-noon (J. Kemp).
- Today’s reading: Goldsmith handout from WebCT. For Thu: Watch MUSE Channel Coding video
- Exam 2 is Tuesday March 9.

1 Fading Demo

To run the demo I am running, you need:

- Two wireless sensors programmed with SPIN, and one programmed with BaseStation, two TinyOS programs. You may borrow these from me.
- A computer with Python, SciPy, and MatPlotLib
- The java package net.tinyos.tools

A Ubuntu OS makes the installation easy as adding packages. Other OSes, I haven’t tried.

My sensors are 2.4-2.48 GHz radios, using DS-SS, with OQPSK modulation, at a data rate of 250 kbps, with a BW of about 5 MHz. The antennas are inverted F on-board antennas. They measure received power in dBm. Both directions of the link are measured.

The questions to answer while participating in the demo are:

1. What type of fading is measured while one transceiver is moving?
2. What type of fading causes the measured $P_r$ to differ when the transceiver has changed positions slightly?
3. What would I see if I could change the center frequency?
4. What is the effect of a person walking near either node, or in the line between them?
5. Can you tell whether a link is in a deep fade or not?
6. What is the dB difference in power between a deep fade and an anti-fade (opposite of a fade)?


2 Multi-carrier Modulation

The basic idea for multi-carrier modulation is to divide the total available bandwidth $B$ into $N$ different subchannels. In each subchannel, we’ll use some particular modulation, and send all signals on all subchannels simultaneously. Considering orthogonal waveforms, we’ll use both the sin and the cosine function at many different frequencies.

The benefit is that with narrower bandwidth subchannels, we can achieve flat fading on each subchannel. Remember, if the bandwidth is low enough, we don’t need an equalizer, because the channel experiences flat fading. (Q: What was the rule given in the Rappaport book given $T_s$ and $\sigma_\tau$ to determine if fading is flat or frequency selective?)

So we will set $N$ high enough so that the symbol rate in each subchannel (which is $B/N$ wide) will be high enough to “qualify” as flat fading. How do you find $T_s$ to fit in $B/N$? Recall the bandwidth is given by $\frac{1}{T_s}(1 + \alpha)$ for rolloff factor $\alpha$ so now $T_s = \frac{N}{B}(1 + \alpha)$.

If we use $T_s \geq 10\sigma_\tau$ then now we can choose $N$ as

$$N \geq 10\sigma_\tau B \frac{1}{1 + \alpha}$$

Now, bit errors are not an ‘all or nothing’ game. In frequency multiplexing, there are $N$ parallel bitstreams, each of rate $\log_2(M)/T_s$, where $\log_2(M)/T_s$ is the bit rate on each subchannel. As a first order approximation, a subchannel either experiences a high SNR and makes no errors; or is in a severe fade, has a very low SNR, and experiences a BER of 0.5 (the worst bit error rate!). If $\beta$ is the probability that a subchannel experiences a severe fade, the overall probability of error will be $0.5\beta$.

Frequency multiplexing is typically combined with channel coding designed to correct a small percentage of bit errors.

Note that the handout uses $T_N$ where we use $T_s$. They reserve $T_s$ to be $T_N/N$. But the symbol period, the duration of the symbol, is in fact what we call $T_s$. We know $N$ symbols are sent simultaneously, but each one still has duration $T_s$. Please make a note.

2.1 OFDM

In the previous section we made no restriction on the frequencies of the subcarriers. Well, we know that to have orthogonal sinusoids at different frequencies, we need a particular condition on $\Delta f$ between the two frequencies.

In FSK, we use a single basis function at each of different frequencies. In QAM, we use two basis functions at the same frequency.
OFDM is the combination:

\[ \phi_0, I(t) = \begin{cases} \cos(2\pi f_c t), & 0 \leq t \leq T_s \\ 0, & \text{o.w.} \end{cases} \]

\[ \phi_0, Q(t) = \begin{cases} \sin(2\pi f_c t), & 0 \leq t \leq T_s \\ 0, & \text{o.w.} \end{cases} \]

\[ \phi_1, I(t) = \begin{cases} \cos(2\pi f_c t + 2\pi \Delta f t), & 0 \leq t \leq T_s \\ 0, & \text{o.w.} \end{cases} \]

\[ \phi_1, Q(t) = \begin{cases} \sin(2\pi f_c t + 2\pi \Delta f t), & 0 \leq t \leq T_s \\ 0, & \text{o.w.} \end{cases} \]

\[ \vdots \]

\[ \phi_{M-1}, I(t) = \begin{cases} \cos(2\pi f_c t + 2\pi (M-1) \Delta f t), & 0 \leq t \leq T_s \\ 0, & \text{o.w.} \end{cases} \]

\[ \phi_{M-1}, Q(t) = \begin{cases} \sin(2\pi f_c t + 2\pi (M-1) \Delta f t), & 0 \leq t \leq T_s \\ 0, & \text{o.w.} \end{cases} \]

where \( \Delta f = \frac{1}{T_s} \). These are all orthogonal functions! We can transmit much more information than possible in \( M \)-ary FSK. (Note we have \( 2M \) basis functions here!)

The signal on subchannel \( k \) might be represented as:

\[ x_k(t) = a_{k,I}(t) \cos(2\pi f_k t) + a_{k,Q}(t) \sin(2\pi f_k t) \]

The complex baseband signal of the sum of all \( K \) signals might then be represented as

\[ x_l(t) = \Re \left\{ \sum_{k=1}^{N} (a_{k,I}(t) + ja_{k,Q}(t))e^{j2\pi k \Delta f t} \right\} \]

\[ x_l(t) = \Re \left\{ \sum_{k=1}^{N} A_k(t)e^{j2\pi k \Delta f t} \right\} \quad (1) \]

where \( A_k(t) = a_{k,I}(t) + ja_{k,Q}(t) \). Does this look like an inverse discrete Fourier transform? If yes, than you can see why it is possible to use an IFFT and FFT to generate the complex baseband signal.

1. FFT implementation: There is a particular implementation of the transmitter and receiver that use FFT/IFFT operations. This avoids having \( N \) independent transmitter chains and receiver chains. The FFT implementation (and the speed and ease of implementation of the FFT in hardware) is why OFDM is popular.

Since the \( N \) carriers are orthogonal, the signal is like \( N \)-ary FSK. But, rather than transmitting on one of the \( K \) carriers at a given time (like FSK) we transmit information in parallel on all
Figure 1: Signal space diagram for $N = 3$ subchannel OFDM with 4-PAM on each channel.

$N$ channels simultaneously. An example state space diagram for $N = 3$ and PAM on each channel is shown in Figure 1.

However, because the FFT implementation assumes that the data is periodic, and it is not, we must add a cyclic prefix at the start of each block of $N$ symbols. The cyclic prefix is length $\mu$ and is an overhead (doesn’t contain data) and thus reduces our overall data rate. The actual data rate will be the total bit rate multiplied by $\frac{N}{N+\mu}$. We determine $\mu$ to be longer than the expected delay spread $\sigma_T$. So, $\mu = \sigma_T/T_s$.

Example: 802.11a
IEEE 802.11a uses OFDM with 52 subcarriers. Four of the subcarriers are reserved for pilot tones, so effectively 48 subcarriers are used for data. Each data subcarrier can be modulated in different ways. One example is to use 16 square QAM on each subcarrier (which is 4 bits per symbol per subcarrier). The symbol rate in 802.11a is 250k/sec. Thus the bit rate is

$$250 \times 10^3 \frac{\text{OFDM symbols}}{\text{sec}} \times 48 \frac{\text{subcarriers}}{\text{OFDM symbol}} \times 4 \frac{\text{coded bits}}{\text{subcarrier}} = 48 \frac{\text{Mb}}{\text{sec}}$$