ECE 5325/6325: Wireless Communication Systems
Lecture Notes, Spring 2013

Lecture 17

Today: Spread Spectrum: (1) Frequency Hopping, (2) Direct Sequence

- Reading: Today – Molisch 18.1, 18.2. Thu: MUSE Channel Coding Video, Mol 14.1, 14.2.
- HW 7 is due today at noon on Canvas or in the locker.
- HW 8 is due Tue March 26
- Exam 3 is Thu April 4
- IEEE Com/SP Utah Section talk: "An Introduction to the Physical Layer of LTE Systems", by Claudio da Silva, 6:30-8pm in WEB 1250, Wed March 20. This talk is directly on 5325/6325 course material, and Dr. da Silva is the expert. We don't cover 4G in this course, so this talk is your chance.

1 Spread Spectrum

"Spread spectrum" is the use of a much wider bandwidth than necessary in a radio communications system, in order to achieve other objectives.

For example, frequency modulation (FM) historically was considered to be widely spread in spectrum (broadcast FM radio uses 200 kHz to send audio, which is inherently a 3-10 kHz bandwidth signal). However, FM was used because it could mitigate the effects of multipath fading, and could achieve better signal quality at low SNR.

The two major types of spread spectrum modulation are:

1. Frequency Hopping Spread Spectrum (FH-SS)
2. Direct-Sequence Spread Spectrum (DS-SS)

Both types of SS use *pseudo-random codes*, periodic sequences of zeros and ones. These are not actually random – a computer generates them using (deterministic) binary feedback logic. But, an average person would look at the sequence and judge them to be random. So they are called pseudo-random.

1.1 FH-SS

FH-SS pseudo-randomly changes center frequency each "hopping period", T_h. Bluetooth is a FH-SS system, which achieves a (coded) bit rate of 1 Mbps (potentially up to 3 Mbps), but uses 80 MHz of spectrum, in 79 different center frequencies, with a hopping period $T_h = 1/1600$ s/hop. While at each center frequency, the modulation is similar to things we have been talking about, *e.g.*, FSK, DPSK, QPSK, etc.

Three benefits of FH-SS are:

1. *Interference avoidance*: There may be significant interference at a few of the center frequencies. But even if we totally lose all bits during hops to those few frequencies, we will be able to recover using the bits received during successful (non-interfered) hops. We also avoid being an interferer to someone else's signal for too long.

2. *Multiple Access*: Two devices can occupy the same spectrum and operate without coordinating medium access at all. Their transmissions will "collide" some small fraction of the time, but (hopefully) not often enough to cause failure.

3. *Stealth*: There is an advantage to switching randomly among frequencies when an eavesdropper doesn't know your hopping pattern – they will not be able to easily follow your signal. This was the original reason for the first use of FH-SS (it was discovered by actor and inventor Hedy Lamarr, in 1940 [1] as a means for covert military communication. In the patent, she describes FH-SS by analogy to piano playing).

1.2 DS-SS

We will see that DS-SS has the same three advantages as FH-SS, but for different reasons.

Direct-sequence spread spectrum simply uses a pulse shape that has a wide bandwidth. This pulse shape is known as a pseudo-noise signal, which is, essentially, itself a BPSK-modulated signal, but which "pseudo-random" data that is known to both transmitter and receiver.

We can describe the modulated signal, sent by user k, as:

$$s_k(t) = a_k(t)p_k(t)\cos(2\pi f_c t)$$

where $a_k(t)$ is the bit signal, +1 or -1, which indicates the data bits to be sent. For example, the Barker code uses bits $[+1, -1, +1, +1, -1, +1, +1, +1, -1, -1, -1]$. As before, we have a pulse shape $p_k(t)$, however, in this case, $p_k(t)$ is not simply a SRRC pulse. It is a high bandwidth pseudo-noise signal. Essentially, $p_k(t)$ is a BPSK-modulated signal itself. The "bits" of the PN signal are called "chips" to distinguish them from bits. We denote the number of chips in $p_k(t)$ as M_C, it is the number of chips per bit. This

M_C is also called the processing gain. The period of each chip is denoted T_c, so

$$M_C = \frac{T_s}{T_c} = \frac{R_c}{R_s}$$

Where $R_c = 1/T_c$ is the chip rate. For example, 802.11b has a chip rate of 11 M (chips per second) and a symbol rate of 1 M (symbols per second).

Note that the chips do not necessarily need to be the same each time. In IS-95 (also called 2G code-division multiple access (CDMA) by Qualcomm), the "short code" has length $2^{15} = 32768$. There are not 32768 chips per bit, though – there are 64 chips per bit. The PN code generator just provides a source of chips that are taken 64 at a time to produce the pulse shape for each data symbol. In the IS-95 case, $M_C = 64$.

Incidentally, also in IS-95 is a long code that has length $2^{42} - 1$. The long code is different for every mobile. The output chips are xor-ed with the long code to make the signal hard to eavesdrop and makes it unique to the particular mobile.

The bandwidth of a DS-SS symbol, when chips have the SRRC shape, is

$$B = (1 + \alpha)R_c$$

Which is then M_C times the bandwidth of the BPSK signal would have been as a narrowband signal.

Recall that the SNR required to demodulate a signal is given by:

$$SNR = \frac{\mathcal{E}_b}{N_0}\frac{R_b}{B}$$

So with DS-SS, the SNR is lowered by a factor of M_C

$$SNR = \frac{\mathcal{E}_b}{N_0}\frac{R_b}{(1+\alpha)R_c} = \frac{\mathcal{E}_b}{N_0}\frac{1}{(1+\alpha)M_C}$$

However, If you thought this signal was just a plain-old BPSK signal, *i.e.*, didn't know the PN signal, you'd need the regular SNR $\frac{\mathcal{E}_b}{N_0}\frac{1}{1+\alpha}$, which is M_C times higher. This makes us understand advantage #3 of DS-SS: Stealth. Knowing the PN signal allows one to demodulate the signal with M_C times less SNR than an eavesdropper could. If the M_C is high enough, the signal would be extremely difficult to detect at all, but could still be used by the intended receiver.

Advantage #1: Reception of DS-SS uses the same principle as discussed earlier – the received signal is correlated with the known PN signal. What if a narrowband interference signal was also in the received signal? Well, this narrowband signal would effectively be spread, when it is multiplied by the PN signal in the receiver. In contrast, the desired signal is de-spread (becomes narrowband again) due to correlation with the PN signal. The spread interference can then be partially filtered out using a narrowband filter.

Advantage #2: Further, DS-SS can be designed for some particular benefits for multiple access. These relate to the near-orthogonality of the particular PN codes used in DS-SS. In short, some sets of PN signals are nearly orthogonal or completely orthogonal *to each other*; and some PN signals have the property that the PN signal is orthogonal *to itself* at a different time delay. This is where the term, code-division multiple access (CDMA) comes from.

First, consider sets of PN signals orthogonal to each other. One example is the set of Gold codes (used in the GPS network). Gold code signals are nearly orthogonal with each other. Another example is the Kasami sequences. See 18.2.6 in Molisch.

Another example is the Walsh-Hadamard (WH) sequence set, used in IS-95 (CDMA). The WH-64 sequences, shown in Figure 1(c), are used in IS-95. The 64 WH signals are exactly orthogonal to each other. These signals provide a means on the downlink to send 64 simultaneous user signals, and yet have each mobile perform a correlation operation that completely zeros out the signal sent from the BS to all 63 other mobiles. When one mobile correlates with its signal, $p_k(t)$, it has zero correlation with the other 63 WH signals.

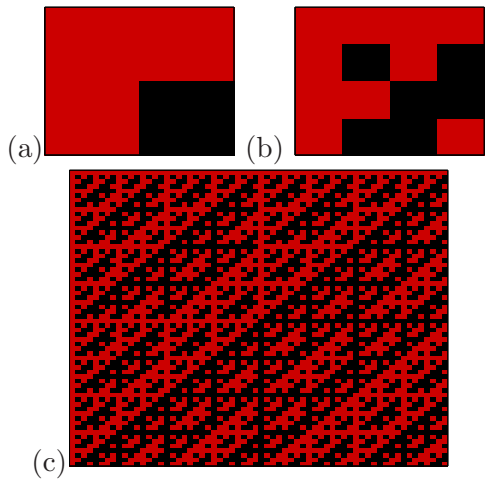

Figure 1: Walsh Hadamard sequences are in the rows of the above images. The signal is +1 during black squares, and -1 during red squares: (a) two WH-2 sequences in two rows, (b) four WH-4 sequences in four rows, and (c) sixty-four WH-64 sequences in 64 rows.

Note that WH sequences are not used on the uplink, because they are only orthogonal if time-synchronized. Mobiles aren't able to time-synchronize with each other very well.

Second, consider the autocorrelation of a PN signal (the correlation of a signal with itself at different time delays). The autocorrelation is defined as

$$R_p(\tau) = \int_0^{T_s} a_k(t) p_k(t) p_k(t-\tau) dt$$

Note this is called $ACF(\tau)$ in Molisch. First assume that the data signal $a_k(t) = 1$. If $\tau = 0$, the value of $R_p(\tau)$ is simply the energy in the PN signal $p_k(t)$ over a duration T_s. For τ a multiple of T_s, that is, the period of $p_k(t)$, we get the same value, *i.e.*, $R_p(nT_s) = R_p(0)$. For in between values of τ, (say, $T_c < \tau < T_s - T_c$, PN signals have a nearly (but not quite) zero autocorrelation. Generally, for these τ, $R_p(\tau) \approx -R_p(0)/M_C$. An example is plotted in 18.1 in Molisch.

1.3 PN code generation

PN code sequences are generated by binary logic gates called "linear feedback shift registers" (LFSR). A properly designed k-state LFSR produces a $2^k - 1$ length PN code sequence. Figure 2(a) and (b) show 4 and 5 state LFSRs. Each state of $s_1, \ldots, s_k$ is either zero or one. Note that we should never have all zero states. Assume that we start out with states that are not all zero. Then, at each time, we compute the mod-2 addition specified, and the output of the adder goes into s_1 in the next time step. Similarly, we shift states, so that state s_i always takes the value that state s_{i-1} had in the previous time step. The output is always drawn from s_k, the rightmost state.

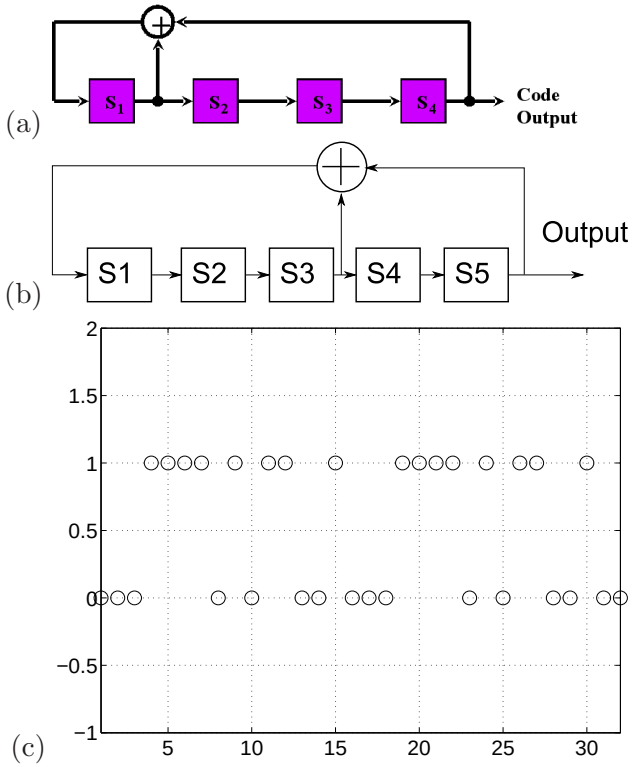

Figure 2: Linear feedback shift register (LFSR) generation of an (a) 4-stage, or (b) 5-stage, maximal-length PN code; and (c) the generated code sequence for the 4-stage LFSR.

Example: 4-state LFSR
Calculate the first 6 outputs of the LFSR given that the initial states $[s_1, s_2, s_3, s_4]$ are [0, 1, 0, 1]. Solution:

Step	s_1	s_2	s_3	s_4	Output
0	0	1	0	1	n/a
1	1	0	1	0	1
2	1	1	0	1	0
3	0	1	1	0	1
4	0	0	1	1	0
5	1	0	0	1	1
6	0	1	0	0	1
7	0	0	1	0	0
8	0	0	0	1	0
9	1	0	0	0	1
10	1	1	0	0	0
11	1	1	1	0	0
12	1	1	1	1	0
13	0	1	1	1	1
14	1	0	1	1	1
15	0	1	0	1	1

References

[1] R. Malik. Spread spectrum - secret military technology to 3G. *IEEE History of Telecommunications Contest*, pages 1–5, 2001.