POWDER Training Lecture Notes

Dr. Neal Patwari ©2025
University of Utah
Kahlert School of Computing

Summary: Here's an eight word summary how digital communications systems communicate: *Send and receive linear combinations of orthogonal waveforms.* That's it! The choice of the waveforms, the number M of (and values of the) different linear combinations is about *the design of the digital communications system*, which we also discuss in these notes.

Some of you have had the introductory material in prior courses. I include the full material in these notes for reference. Here is a summary of the main takeaways in each section. First, Section 1 (Orthogonality):

1. We use a K dimensional basis to represent all the possible symbols we could send.
2. We make a list of M possible symbols within this basis.
3. Each symbol becomes a K-dimensional vector.
4. The received signal is separated into the K possible components (bases), and becomes a K-dimensional vector.

For Section 2 (Pulse Shapes):

1. Pulse shapes are designed to be orthogonal to the other copies of the pulse shape, at integer multiples of T_s, the symbol period.
2. There is a systematic way to find pulse shapes called the Nyquist **Filtering** Theorem.
3. The square-root raised cosine (SRRC) is the most common example of a pulse shape, and the formula is given in (10).

For Section 3 (QAM):

1. The most common (for narrowband systems) modulation is M-QAM, which uses $K = 2$, where the two orthogonal waveforms are a cosine and a sine, multiplied by a pulse shape.
2. The 2D symbol vector is equivalent to a complex value. We use I for the horizontal / cosine / real / in-phase component, and Q for the vertical / sine / imaginary / quadrature component.
3. Most commonly, QAM uses a square grid of possible symbol vectors, and thus $M = 4^n$ for positive integer n.
4. The most common exception to the above is BPSK, which uses $K = 1$, i.e., only the cosine multiplied by a pulse shape.

For Section 4 (Optimal Detection):

1. With some reasonable assumptions we can show that the optimal Bayesian detector, given the received signal vector, picks the symbol vector (of the M possible symbols) that is closest in Euclidean distance to the received signal vector. That's it – the symbol decision is picking the closest point.

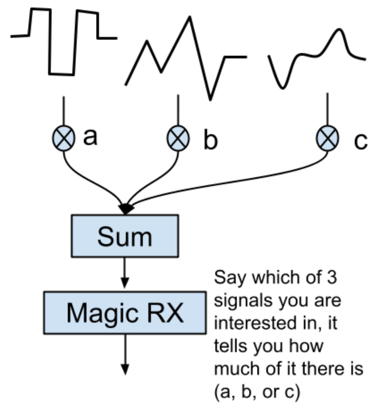

Figure 1: A receiver does the 'magic' of separating multiple (orthonormal) signals received in some linear combination, and telling you how much of each was in that combination.

Section 5 (SDR Implementation) introduces an overview of the digital communications Lab we'll do in the next class. There are some implementation details we cover including digital-to-analog conversion and analog-to-digital conversion; gain control; Shout; phase, frequency, and symbol time synchronization.

For Section 6 (Probability of Bit Error):

1. We have analytical formulas that, given the signal power and noise power (or energy), we can find the probability of bit error. Alternatively, for a given probability of bit error, we can determine the required $\frac{\mathcal{E}_b}{N_0}$ or C/N.

1 Orthogonality

Orthogonal signals make digital communications systems work, and work together. This is because

1. *Multiple Access*: Multiple users can access the same medium by using orthogonal signals, and a receiver can separate one user's signal from the rest because the others can be zeroed out.

2. *Increasing Signal Dimension*: A single user can send information along multiple dimensions at the same time, which is useful for increasing the bit rate or fidelity.

3. *Multiple Symbols over Time*: We want not just to send one symbol (1 or more bits sent at one time), we want to send symbol after symbol over time. By choosing a symbol that is orthogonal to itself at different multiples of T_s, the "symbol period", the receiver can separate the info sent at subsequent symbol periods.

Also, I can give an "engineering" definition of a set of orthogonal waveforms: *They are waveforms that can be separated.*

Here's a graphical argument. Consider Figure 1. Imagine that you have three random-looking signals that you know. But you've accidentally multiplied them by different numbers a, b, and c, and then added them together. Now you need a magic machine to tell you how much of the first signal there was. It would be hard in general, but when these signals are mutually orthonormal, then the machine is easy. It would just multiply with the first signal, and integrate. The number out of the integral is the value a you were looking for.

Now, let's provide the mathematical definition.

<u>Def'n:</u> *Orthogonal*
Two real-valued waveforms $\phi_0(t)$ and $\phi_1(t)$ are *orthogonal* if

$$\int_{-\infty}^{\infty} \phi_0(t)\phi_1(t)dt = 0.$$

What does this integral remind you of?

Example: Sine and Cosine
Let

$$\begin{aligned}
\phi_0(t) &= \begin{cases} \cos(2\pi t), & 0 < t \leq 1 \\ 0, & o.w. \end{cases} \\
\phi_1(t) &= \begin{cases} \sin(2\pi t), & 0 < t \leq 1 \\ 0, & o.w. \end{cases}
\end{aligned}$$

Are $\phi_0(t)$ and $\phi_1(t)$ orthogonal?
Solution: Using $\sin 2x = 2\cos x \sin x$,

$$\begin{aligned}
\int_{-\infty}^{\infty} \phi_0(t)\phi_1(t)dt &= \int_0^1 \cos(2\pi t)\sin(2\pi t)dt \\
&= \int_0^1 \frac{1}{2}\sin(4\pi t)dt \\
&= \frac{-1}{8\pi}\cos(4\pi t)\Big|_0^1 = \frac{-1}{8\pi}(1-1) = 0
\end{aligned}$$

So, yes, $\phi_0(t)$ and $\phi_1(t)$ are orthogonal.

Example: Frequency Shift Keying
Assume $T_s >> 1/f_c$, and show that these two are orthogonal.

$$\begin{aligned}
\phi_0(t) &= \begin{cases} \cos(2\pi f_c t), & 0 \leq t \leq T_s \\ 0, & o.w. \end{cases} \\
\phi_1(t) &= \begin{cases} \cos\left(2\pi\left[f_c + \frac{1}{T_s}\right]t\right), & 0 \leq t \leq T_s \\ 0, & o.w. \end{cases}
\end{aligned}$$

Solution: The integral of the product of the two must be zero. Checking, and using the identity for the product of two cosines,

$$\begin{aligned}
&\int_0^{T_s} \cos(2\pi f_c t)\cos\left(2\pi\left[f_c + \frac{1}{T_s}\right]t\right)dt \\
&= \frac{1}{2}\int_0^{T_s} \cos(2\pi t/T_s)\,dt + \frac{1}{2}\int_0^{T_s} \cos(4\pi f_c t + 2\pi t/T_s)\,dt \\
&= 0 + \frac{1}{2}\left[\frac{1}{2\pi(2f_c + 1/T_s)}\sin(2\pi(2f_c + 1/T_s)t)\Big|_0^{T_s}\right.
\end{aligned}$$

The remaining term has a $\frac{1}{2\pi(2f_c+1/T_s)}$ constant out front. Because f_c is very very high (typically on the order of 10^9 in wireless applications) this term will be very very low. The sine term is limited to between -1 and +1 so it will not cause the second term to be large. (Alternatively, one can set f_cT_s to be some integer multiple of 0.5, which would give exact orthogonality.)

Thus,

$$\int_{-\infty}^{\infty} \phi_0(t)\phi_1(t)dt \leq \frac{1}{\pi(2f_c + 1/T_s)} \quad \approx \quad 0$$

Thus the two different frequency waveforms are orthogonal.

1.1 Orthonormal Basis

Taking orthogonality one step further, we define an orthonormal basis:

<u>Def'n:</u> *Orthonormal Basis*
A set of functions $\phi_0(t), \ldots, \phi_{K-1}(t)$ are *orthonormal* (the set is referred to as an *orthonormal basis*) if they are mutually orthogonal and each has unit energy, *i.e.*, $\int_{t=-\infty}^{\infty} |\phi_k(t)|^2 dt = 1$ for $k = 0, \ldots, K-1$.

You might remember in a linear algebra course talking about a *vector space* having a *basis* which is a set of vectors which can build (from some linear combination of basis vectors) any vector in the space. This is analogous, with functions rather than vectors. The word "basis" also implies you can build something out of a linear combination of functions in the orthonormal basis.

For shorthand, we use a set name to describe the orthonormal basis. The Rice book [2] uses $\mathcal{B}$:

$$\mathcal{B} = \{\phi_0(t), \phi_1(t), \ldots, \phi_{K-1}(t)\}$$

Each function in $\mathcal{B}$ is called a basis function. You can think of $\mathcal{B}$ as an unambiguous and useful language.

What are some common orthonormal bases?

- Nyquist sampling: sinc functions centered at sampling times.
- Fourier series: complex sinusoids at different frequencies
- Sine and cosine at the same frequency
- Wavelets
- N-dimensional vectors, where each vector is a different unit vector along an orthogonal axis.

And, we will come up with others. Each one has a limitation – only a certain set of signals can be exactly represented as a linear combination of its waveforms. Essentially, we must limit the set of possible signals to a set. That is, some subset of all possible signals.

1.2 Linear Combinations

What is a linear combination of orthonormal waveforms? Well, consider the orthonormal set $\phi_0(t), \ldots, \phi_{K-1}(t)$. A linear combination $s_m(t)$ is

$$s_m(t) = a_{m,0}\phi_0(t) + a_{m,1}\phi_1(t) + \cdots + a_{m,K-1}\phi_{K-1}(t) = \sum_{m=k}^{K-1} a_{m,k}\phi_k(t)$$

We also call the a linear combination a *symbol*. We use subscript m to indicate that it's not the only possible linear combination (or symbol). In fact, we will use M different symbols, so $i = 0, \ldots, M-1$, and we will use $s_0(t), \ldots, s_{M-1}(t)$.

We represent the mth symbol (linear combination of the orthonormal waveforms), $s_m(t)$, as a vector for ease of notation:

$$\mathbf{s}_m = [a_{m,0}, a_{m,1}, \ldots, a_{m,K-1}]^T$$

The superscript T is for transpose – $\mathbf{s}_m$ is a column vector. Vectors are easy to deal with because they can be plotted in vector space, to show graphically what is going on. We call the plot of all possible $\mathbf{s}_m$, that is, for $m = 0, \ldots M-1$, the *constellation diagram*. Some examples are shown in Figure 2.

1.3 Span of an Orthonormal Basis

<u>Def'n:</u> *Span*
The span of the set $\mathcal{B}$ is the set of all functions which are linear combinations of the functions in $\mathcal{B}$. The span is referred to as $\text{Span}\{\mathcal{B}\}$ and is

$$\text{Span}\{\mathcal{B}\} = \left\{ \sum_{k=0}^{K-1} a_k\phi_k(t) \text{ such that } a_0, \ldots, a_{K-1} \in \mathbb{R} \right\}$$

Another way to define this set is to say that a function $f(t) \in \text{Span}\{\mathcal{B}\}$ if and only if there exists $a_0, \ldots, a_{K-1} \in \mathbb{R}$ such that

$$f(t) = \sum_{k=0}^{K-1} a_k\phi_k(t)$$

In short, the Span of $\mathcal{B}$ is the space of possible symbol waveforms. The symbols $s_m(t)$ for $m = 0, \ldots M-1$ are the ones we choose to use to convey information between the transmitter and receiver. We call the set of the M possible symbols $\mathcal{S}$. Since every element of $\mathcal{S} \in \text{Span}\{\mathcal{B}\}$, it is clear that $\mathcal{S} \subset \text{Span}\{\mathcal{B}\}$.

1.4 Using M Different Linear Combinations

Here is how a transmitter *uses* the different linear combinations to convey digital bits to the receiver. First, consider that there are M different symbols for the TX to chose from. Each symbol is described by a $(\log_2 M)$-length bit sequence. For example, if there are 8 possible combinations, we would label them 000, 001, 011, 010, 110, 111, 101, 100.

The transmitter knows which $\log_2 M$-bit sequence it wants to send. It picks the symbol that corresponds to that bit sequence, let's call it symbol m. Then it transmits $s_m(t)$.

If the receiver is able to determine that symbol m was sent, it will correctly receive those $\log_2 M$ bits of information. In this example, it will receive three bits of information.

Next we will talk about how a receiver is able to separate the received signal into components, each corresponding to one of the orthonormal waveforms $\phi_k(t)$. From this separation, it will be able to decide which symbol was sent.

1.5 How to Choose a Modulation

A *digital modulation* is the choice of: (1) the linear combinations $\mathbf{s}_0, \ldots, \mathbf{s}_{M-1}$ and, (2) the orthonormal waveforms $\phi_0(t), \ldots, \phi_{K-1}(t)$. We will choose a digital modulation as a tradeoff between the following characteristics:

1. Bandwidth efficiency: How many bits per second (bps) can be sent per Hertz of signal bandwidth. Thus the bandwidth efficiency has units of bps/Hz.
2. Power efficiency: How much energy per bit is required at the receiver in order to achieve a desired fidelity (low bit error rate). We typically use S/N or E_s/N_0 or E_b/N_0 as our figure of merit. This will be a major topic of the 2nd part of this course.
3. Cost of implementation: Things like symbol and carrier synchronization, and linear transmitters, require additional device cost, which might be unacceptable in a particular system.

1.6 Analysis

At a receiver, our job will be to analyze the received signal (a function) and to decide which of the M possible signals was sent. This is the task of *analysis*. It turns out that an orthonormal bases makes our analysis very straightforward and elegant.

We won't receive exactly what we sent - there will be additional functions added to the signal function we sent due to noise and interference. We might say that if we send signal m, *i.e.*, $s_m(t)$ from our signal set, then we would receive

$$r(t) = s_m(t) + w(t)$$

where the $w(t)$ is the sum of all of the additive signals that we did not intend to receive. But $w(t)$ would generally not be totally in the span of our basis $\mathcal{B}$, so $r(t)$ would not be in Span$\{\mathcal{B}\}$ either.

1.6.1 Symbol Closest to Received Signal

One main question we ask at the receiver is, what is the symbol $s_m(t)$ that is closest to the received signal? The term "closest" here is somewhat qualitative until we define it. Without proof (in this lecture) we are going to use squared error to measure "closeness":

$$\mathcal{E}_m = \int_{-\infty}^{\infty} |r(t) - s_m(t)|^2 dt \tag{1}$$

That is, for any given m, we would integrate across time the squared difference between $r(t)$ and $s_m(t)$. Our decision will be the m that has minimum $\mathcal{E}_m$:

$$\hat{m} = \arg \min_{m \in \{0 \ldots M-1\}} \left\{ \int_{-\infty}^{\infty} |r(t) - s_m(t)|^2 dt \right\} \tag{2}$$

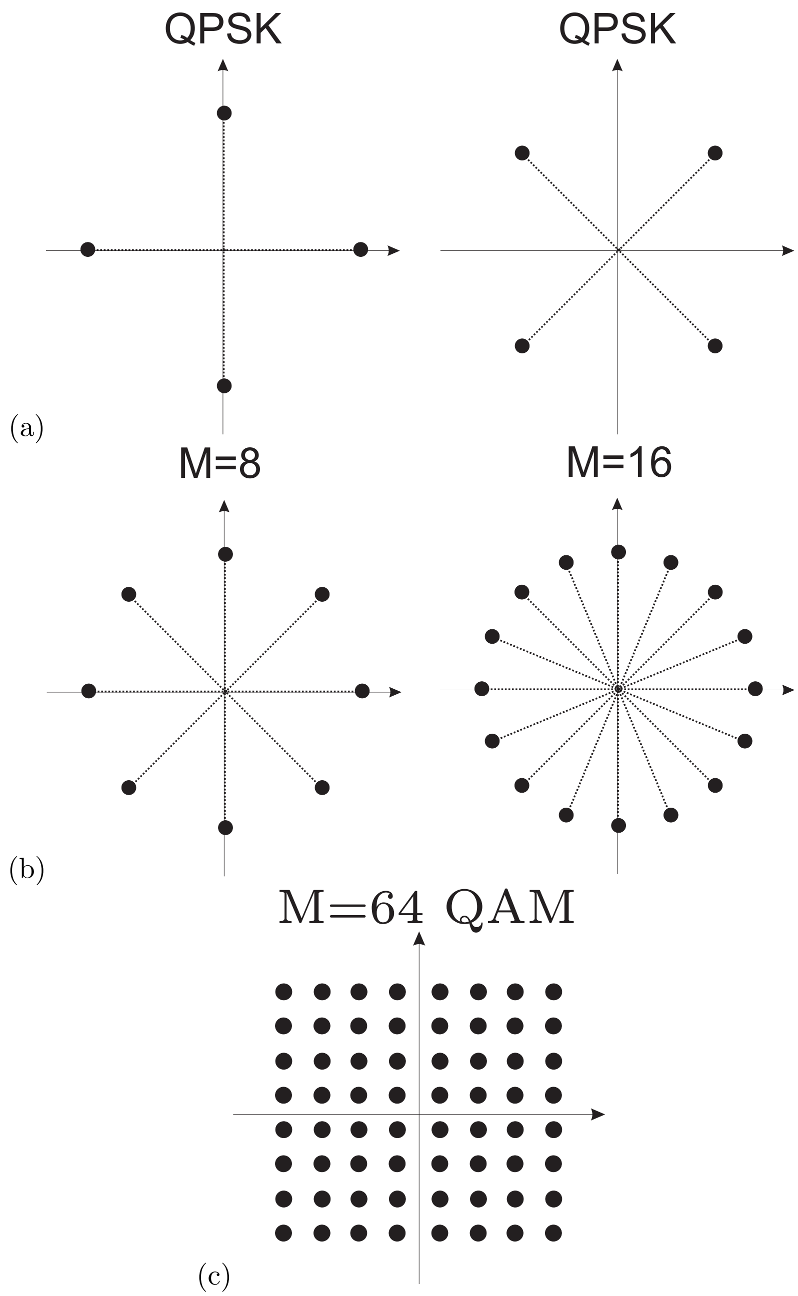

Figure 2: Signal constellations for (a) $M = 4$ PSK (a.k.a. BPSK), (b) $M = 8$ and $M = 16$ PSK, and (c) 64-QAM.

The term $\hat{s}(t)$ is the result, the receiver's guess of the transmitted symbol. Because $s_m(t) \in \mathcal{S}$, we know that $s_m(t) = \sum_{k=0}^{K-1} a_{m,k}\phi_{m,k}$. So,

$$\begin{aligned}
\hat{m} &= \arg \min_{m\in\{0\ldots M-1\}} \left\{ \int_{t=-\infty}^{\infty} r^2(t)dt \right. \\
&\quad -2\sum_{k=0}^{K-1} a_{m,k} \int_{t=-\infty}^{\infty} r(t)\phi_k(t)dt \\
&\quad \left. + \sum_{k=0}^{K-1}\sum_{k'=0}^{K-1} a_{m,k}a_{m,k'} \int_{t=-\infty}^{\infty} \phi_k(t)\phi_{k'}(t)dt \right\}.
\end{aligned} \tag{3}$$

Although there are a lot of terms, this begins to simplify. The first term is not a function of m. The integral in the second term is the inner product $\langle r(t), \phi_k(t)\rangle$. Let's call this x_k. The integral in the third term is one when $k = k'$ and zero otherwise, so we can just consider the terms when $k = k'$,

$$\arg \min_{m\in\{0\ldots M-1\}} \left\{ -2\sum_{k=0}^{K-1} a_{m,k}x_k + \sum_{k=0}^{K-1} a_{m,k}^2 \right\}. \tag{4}$$

We can make this easier for us by including a constant $\sum_{k=0}^{K-1} x_k^2$. Since this constant is not a function of m, it doesn't affect the arg min.

$$\begin{aligned}
&= \arg \min_{m\in\{0\ldots M-1\}} \left\{ \sum_{k=0}^{K-1} \left(x_k^2 - 2a_{m,k}x_k + a_{m,k}^2\right) \right\}. \\
&= \arg \min_{m\in\{0\ldots M-1\}} \left\{ \sum_{k=0}^{K-1} \left(x_k - a_{m,k}\right)^2 \right\}.
\end{aligned} \tag{5}$$

To find the m that makes this the smallest, then, we should calculate the x_k values by finding the inner product between the received signal and $\phi_k(t)$, and forming the vector

$$\mathbf{x} = [x_0, x_1, \ldots, x_{K-1}]^T$$

Keep a list of the symbol vectors $\mathbf{s}_m$ for each m. The final expression above translates to

$$\hat{m} = \arg\min_m \|\mathbf{x} - \mathbf{s}_m\|^2 \tag{6}$$

where $\|\cdot\|^2$ is the squared norm of a vector. Thus, just find the squared Euclidean distance between each $\mathbf{s}_m$ and $\mathbf{x}$. This lowest-squared distance vector corresponds to the m that is closest to the received signal.

Note that $\mathbf{x}$, that is, the inner products

$$x_k = \int_{t=-\infty}^{\infty} r(t)\phi_k(t)dt$$

are all that matters in the decision about which symbol was sent.

2 Pulse Shapes

Consider two options for a pulse shape:

1. Consider the spectrum of using a pulse that is a rect function, ie., constant between 0 and T_s, the symbol period.
2. Consider the time-domain of the signal which is close to a rect function in the frequency domain.

We don't want either: (1) the pulse occupies too much spectrum by having high sidelobes in the frequency domain, and (2) the pulse occupies too much time.

1. If we try (1) above and use FDMA (frequency division multiple access) then the interference is *out-of-band interference*.
2. If we try (2) above and put symbols right next to each other in time, our own symbols can experience interference called *inter-symbol interference* whenever we don't sample the output of the pulse shape filter at exactly the right time (or when the multipath in the radio channel make multiple pulses arrive at small time delays).

In reality, we want to compromise between (1) and (2) and experience only a small amount of both.

2.1 Nyquist Filtering

We want to extend the pulse $p(t)$ to be longer in duration than only between zero and T_s, but we want to do so in a way that preserves the property that $p(t)$ and $p(t-T_s)$, and for that matter, $p(t-nT_s)$ for integers $n \neq 0$, are orthogonal.

Key insight: we don't need $p(t)$ and $p(t-\tau)$ to be orthogonal for all real-valued times τ – only for $\tau = nT_s$, for integer n. In other words, the set:

$$\ldots, p(t+2T_s), p(t+T_s), p(t), p(t-T_s), p(t-2T_s), \ldots$$

form an orthonormal set. The Nyquist filtering condition is a frequency-domain rule that can be used to design arbitrary pulse shapes $p(t)$ that meet this condition.

I'm going to use frequency in Hertz f, rather than radial frequency ω, for this discussion, as it makes expressions a little easier.

Theorem: Nyquist Filtering
Proof: Let $r_p(t) = \int_{\tau=-\infty}^{\infty} p(\tau)p(\tau-t)d\tau$ be the autocorrelation function of pulse shape $p(t)$. A necessary and sufficient condition for $r_p(t)$ to satisfy

$$r_p(nT_s) = \begin{cases} 1, & n=0 \\ 0, & \text{other integer } n \end{cases}$$

is that its Fourier transform $R_p(f)$ satisfy

$$\sum_{m=-\infty}^{\infty} R_p\left(f + \frac{m}{T_s}\right) = T_s$$

Proof: on page 677, Appendix A, of Rice book [2].

Note that $r_p(t)$ doesn't need to be any particular value at any real-valued t other than nT_s.

Some comments on what $R_p(f)$ could be:

- $R_p(f) = \text{rect}(fT_s)$, *i.e.*, exactly constant within $-\frac{1}{2T_s} < f < \frac{1}{2T_s}$ band and zero outside.
- It may bleed into the next 'channel' but the sum of

$$\cdots + R_p\left(f - \frac{1}{T_s}\right) + R_p(f) + R_p\left(f + \frac{1}{T_s}\right) + \cdots$$

must be constant across all f. But the neighboring frequency-shifted copy of $R_p(f)$ must be lower s.t. the sum is constant. See Figure 3.

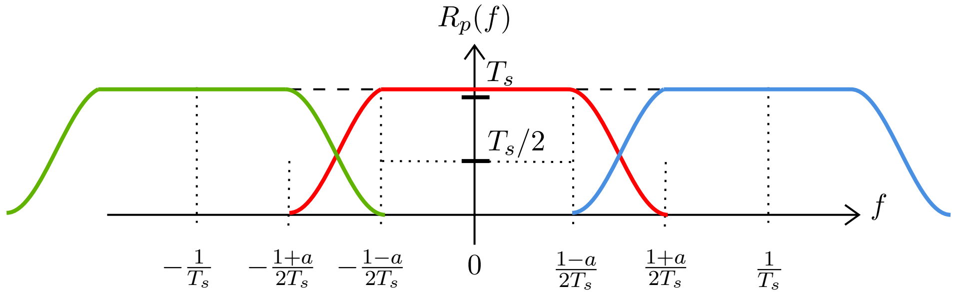

Figure 3: The Nyquist filtering criterion: $1/T_s$-frequency-shifted copies of $R_p(f)$ must add up to a constant (T_s). This plot shows $R_p(f)$ for the "square root raised cosine" pulse shape.

If $R_p(f)$ only bleeds into one neighboring channel (that is, $R_p(f) = 0$ for all $|f| > \frac{1}{T_s}$), we denote the difference between the ideal rect function and our $R_p(f)$ as $\Delta(f)$,

$$\Delta(f) = |R_p(f) - T_s\text{rect}(fT_s)|$$

then we can rewrite the Nyquist Filtering condition as,

$$\Delta\left(\frac{1}{2T_s} - f\right) = \Delta\left(\frac{1}{2T_s} + f\right), \quad \text{for all } -\frac{1}{T_s} \leq f < \frac{1}{T_s}$$

Essentially, $R_p(f)$ is symmetric about $f = \frac{1}{2T_s}$ and $R_p(f) = \frac{T_s}{2}$, i.e., if it was folded twice at those lines it would line up.

Andy Bateman (*Digital Communications: Design for the Real World*, 1998) presents this condition as "A Nyquist channel response is characterized by the transfer function having a transition band that is symmetrical about a frequency equal to $0.5 \times 1/T_s$."

Activity: Do-it-yourself Nyquist filter. Take a sheet of paper and fold it in half on the longest side, and then in half the other direction. Cut a function in the thickest side (the edge that you just folded). Leave a tiny bit so that it is not completely disconnected into two halves. Unfold. Drawing a horizontal line for the frequency f axis, the middle is $0.5/T_s$, and the vertical axis it $R_p(f)$. One half (bottom or top) will be a plot of $R_p(f)$, and the other half will be an (inverse) plot of $R_p(f - 1/T_s)$.

2.2 How to get $p(t)$ from $R_p(f)$

Assuming now $R_p(f)$ meets the Nyquist filtering condition, how do we go from that to $p(t)$? The pulse shape $p(t)$ must be such that $r_p(t)$, the autocorrelation function, has the Fourier transform

$R_p(f)$. Recall that convolution in the time domain is multiplication in the frequency domain. And autocorrelation (correlation with the same function itself) is convolution with a time-reversed version of itself: $r_p(t) = p(t) \star p(-t)$. Let's assume that the pulse shape is real-valued and symmetric about $t = 0$ since that's the only kind of pulse shape I've ever seen used. Thus

$$R_p(f) = \mathfrak{F}\left\{p(t) \star p(-t)\right\} = P(f)P(f) = |P(f)|^2.$$

This means that $|P(f)| = |R_p(f)|^{1/2}$ or

$$p(t) = \mathfrak{F}^{-1}\left\{|R_p(f)|^{1/2}\right\}. \tag{7}$$

Technically we could have $|P(f)| = \pm|R_p(f)|^{1/2}$, but the sign is arbitrary, as long as the transmitter and receiver agree on it.

2.3 Square Root Raised Cosine Pulse Shapes

The raised-cosine function has an autocorrelation function of:

$$R_p(f) = \begin{cases} T_s, & 0 \le |f| \le \frac{1-\alpha}{2T_s} \\ \frac{T_s}{2}\left\{1 + \cos\left[\frac{\pi T_s}{\alpha}\left(|f| - \frac{1-\alpha}{2T_s}\right)\right]\right\}, & \frac{1-\alpha}{2T_s} \le |f| \le \frac{1+\alpha}{2T_s} \\ 0, & o.w. \end{cases} \tag{8}$$

where α is a parameter between 0 and 1 that indicates how quickly (α close to 0) or how slowly (α close to 1) the pulse shape's frequency content transitions from its maximum T_s to zero.

From (7) we need the square root of $R_p(f)$, which is this pulse shape is called the square root raised cosine (SRRC):

$$|R_p(f)|^{1/2} = \begin{cases} \sqrt{T_s}, & 0 \le |f| \le \frac{1-\alpha}{2T_s} \\ \sqrt{\frac{T_s}{2}\left\{1 + \cos\left[\frac{\pi T_s}{\alpha}\left(|f| - \frac{1-\alpha}{2T_s}\right)\right]\right\}}, & \frac{1-\alpha}{2T_s} \le |f| \le \frac{1+\alpha}{2T_s} \\ 0, & o.w. \end{cases} \tag{9}$$

Finally, taking the inverse Fourier transform of this, with some manipulation and use of our favorite Fourier transform properties and pairs, we get the $p(t)$ result in the Rice book (A.30) [2]:

$$p(t) = \frac{1}{\sqrt{T_s}} \frac{\sin\left(\frac{\pi(1-\alpha)t}{T_s}\right) + \frac{4\alpha t}{T_s}\cos\left(\frac{\pi(1+\alpha)t}{T_s}\right)}{\frac{\pi t}{T_s}\left[1 - \left(\frac{4\alpha t}{T_s}\right)^2\right]}, \tag{10}$$

where t is the time variable, and α and T_s are constants. Incidentally, this would be a good proof for a motivated student. Please note that the function is 0/0 for $t = 0$ which one can avoid by calculating it for some very small $t > 0$ (the engineering solution) or by using L'Hôpital's rule (the math solution).

3 Quadrature Amplitude Modulation (QAM)

Quadrature amplitude modulation (QAM) is a two-dimensional bandpass signaling method which uses the in-phase and quadrature (cosine and sine, respectively) at the same frequency as the two basis functions. In other words, QAM's two basis functions are:

$$\begin{aligned} \phi_0(t) &= \sqrt{2}p(t)\cos(\omega_0 t) \\ \phi_1(t) &= -\sqrt{2}p(t)\sin(\omega_0 t) \end{aligned} \tag{11}$$

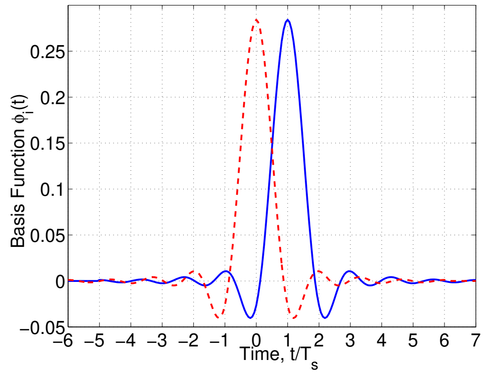

Figure 4: Two SRRC Pulses, delayed in time by nT_s for any integer n, are orthogonal to each other.

where $p(t)$ is a pulse shape that meets the Nyquist filtering theorem with support on $T_1 \leq t \leq T_2$ for some real constants $T_1 < T_2$. That is, $p(t)$ is only non-zero within that window, and $T_1 < 0$ and $T_2 > T_s$. (We remove the assumption that it is non-zero for all time, as this is not a realistic assumption.)

In most flow chart-type drawings of transmitters and receivers, frequency up-conversion and down-conversion are separate from pulse shaping, which is largely true to how the device operates. The definition of the orthonormal basis in (11) specifically considers a pulse shape at a frequency ω_0. We include it here because it is *critical* to see how, with the same pulse shape $p(t)$, we can have two orthogonal basis functions. (This is not intuitive!)

We have two restrictions on $p(t)$ that makes these two basis functions orthonormal:

- $p(t)$ is unit-energy.
- $p(t)$ is 'low pass'; that is, it has low frequency content compared to ω_0.

3.1 Showing Orthogonality

The two basis functions in (11) are (approximately) orthogonal, and one could show this mathematically by multiplying and integrating them. The short story is that the product creates a double-frequency sinusoid centered at 0. Because its frequency is so high compared to the slow changes in $p(t)$, each cycle approximately cancels itself out.

3.2 Complex Baseband

With these two basis functions, M-ary QAM is defined as an arbitrary signal set $\mathbf{s}_0, \ldots, \mathbf{s}_{M-1}$, where each signal space vector $\mathbf{s}_m$ is two-dimensional:

$$\mathbf{s}_m = \begin{bmatrix} a_{m,0} \\ a_{m,1} \end{bmatrix}$$

The signal corresponding to symbol m in (M-ary) QAM is thus

$$\begin{aligned} s_m(t) &= a_{m,0}\phi_0(t) + a_{m,1}\phi_1(t) \\ &= a_{m,0}\sqrt{2}p(t)\cos(\omega_0 t) - a_{m,1}\sqrt{2}p(t)\sin(\omega_0 t) \\ &= \sqrt{2}p(t)\left[a_{m,0}\cos(\omega_0 t) - a_{m,1}\sin(\omega_0 t)\right] \end{aligned} \tag{12}$$

Note that we could also write the signal $s(t)$ as

$$\begin{aligned} s_m(t) &= \sqrt{2}p(t)\mathbb{R}\left\{a_{m,0}e^{j\omega_0 t} + ja_{m,1}e^{j\omega_0 t}\right\} \\ &= \sqrt{2}p(t)\mathbb{R}\left\{e^{j\omega_0 t}\left(a_{m,0} + ja_{m,1}\right)\right\} \end{aligned} \tag{13}$$

In many textbooks, you will see them write a QAM signal in shorthand as

$$s_m^{CB}(t) = p(t)(a_{m,0} + ja_{m,1}) \tag{14}$$

This is called *complex baseband*. If you do the following operation you can recover the real signal $s_m(t)$ as

$$s_m(t) = \sqrt{2}\mathbb{R}\left\{e^{j\omega_0 t}s_m^{CB}(t)\right\} \tag{15}$$

In this notation symbol m is represented with a complex number

$$s_m^{CB} = a_{m,0} + ja_{m,1}$$

instead of a vector $\mathbf{s}_m = [s_{m,0}, s_{m,1}]^T$. You can think of the two versions (complex value, 2-D vector) as being equivalent. We will sometimes say the two components are real and the imaginary components. We also sometimes call the components the in-phase and quadrature components.

Many other books use complex baseband notation. Software defined radios also use complex baseband, in the sense that samples are represented as complex valued. We interpret the samples as $s_m^{CB}(nT)$ from the complex baseband signal in (14).

3.3 Signal Constellations

The signal space representation $\mathbf{s}_m$ is given by

$$\mathbf{s}_m = [a_{m,0}, a_{m,1}]^T$$

for $m = 0, \ldots, M-1$.

- See Figure 5.3.3 in the Rice book for examples of square QAM [2]. These constellations use $M = 2^a$ for some **even** integer a, and arrange the points in a grid. One such diagram for $M = 64$ square QAM is also given in Figure 2(c).
- Figure 5.3.4 shows examples of constellations which use $M = 2^a$ for some **odd** integer a, and arrange the points in a grid. These are either rectangular grids, or squares with the corners cut out, or more hexagonal grids.

3.4 Angle and Magnitude Representation

You can plot $\mathbf{s}_m$ in signal space and see that it has a magnitude (distance from the origin) of $|\mathbf{s}_m| = \sqrt{a_{m,0}^2 + a_{m,1}^2}$ and angle of $\angle\mathbf{s}_m = \tan^{-1}\frac{a_{m,1}}{a_{m,0}}$. In the continuous time signal $s(t)$ this is

$$s(t) = \sqrt{2}p(t)|\mathbf{s}_m|\cos(\omega_0 t + \angle\mathbf{s}_m)$$

3.5 Phase-Shift Keying

Some implementations of QAM limit the constellation to include only signal space vectors with equal magnitude, *i.e.*,

$$|\mathbf{s}_0| = |\mathbf{s}_1| = \cdots = |\mathbf{s}_{M-1}|$$

The points $\mathbf{s}_m$ for $m = 0, \ldots, M-1$ are uniformly spaced on the unit circle. Some examples are shown in Figure 2(a) and (b).

QPSK $M = 4$ PSK is also called quadrature phase shift keying (QPSK), and is shown in Figure 2(a). Note that the rotation of the signal space diagram doesn't matter, so both 'versions' are identical in concept (although would be a slightly different implementation). Note how QPSK is the same as $M = 4$ square QAM.

3.6 Systems which use QAM

See the Couch book [1], numerous Wikipedia pages, and the Rice book [2]:

- 100 Gbps Ethernet (optical over fiber): NRZ (rectangular) pulses and M-PAM with $M = 2$ or 4.
- Digital Microwave Relay, various manufacturer-specific protocols. 6 GHz, and 11 GHz.
- 5G. Each subchannel uses QAM modulation from QPSK to 256 QAM.
- DSL. G.DMT uses multicarrier (up to 256 carriers) methods (OFDM), and on each narrowband (4.3kHz) carrier, it can send up to 2^{15} QAM (32,768 QAM). G.Lite uses up to 128 carriers, each with up to $2^8 = 256$ QAM.
- Cable modems. Upstream: 6 MHz bandwidth channel, with 64 QAM or 256 QAM. Downstream: QPSK or 16 QAM.
- 802.11a, 802.11g: Adaptive modulation methods, use up to 64 QAM.
- 802.11ac, 11ax: uses up to 1024 QAM.
- Digital Video Broadcast (DVB): APSK used in ETSI standard.

Some gains in bit rate have come from higher M modulations; but there is a limit since the bits/symbol are proportional to $\log_2 M$, which increases only slowly with M. A lot of gains thus have come just from increasing the bandwidth of the signals, or using MIMO with many antennas (which we cover later).

3.7 Bandwidth of QAM, PAM, PSK

The bandwidth of all QAM modulations are all determined by the bandwidth of the pulse used. For square root-raised cosine (SRRC) pulses, the null to null bandwidth is

$$B = \frac{1+\alpha}{T_s} \tag{16}$$

where α is the "rolloff" factor or "excess bandwidth" parameter of the SRRC pulse. This is as given in the definition of SRRC pulses in (9). Recall if $\alpha = 0$ then the pulse shape is a rect in the frequency domain and a sinc in the time domain, and has the smallest bandwidth.

4 M-ary Detection Theory with K-Dimensional Signals

We are going to start to talk about QAM, PSK, and FSK, modulations with $K \geq 2$ basis functions. Our setup:

- Transmit: one of M possible symbols, $\mathbf{s}_0, \ldots, \mathbf{s}_{M-1}$. Recall these $\mathbf{s}_i$ vectors are length K, we write its elements as:
$$\mathbf{s}_i = \begin{bmatrix} a_{i,0} \\ \vdots \\ a_{i,K-1} \end{bmatrix}.$$
- Receive: the symbol vector plus noise, after the matched filter and downsampler:
$$\begin{aligned} H_0: &\quad \mathbf{X} = \mathbf{s}_0 + \mathbf{W} \\ \cdots &\quad \cdots \\ H_{M-1}: &\quad \mathbf{X} = \mathbf{s}_{M-1} + \mathbf{W} \end{aligned}$$
- Assume: $\mathbf{W}$ is multivariate Gaussian, each of K components W_k are independent with zero mean and variance $\sigma_W^2 = N_0/2$.
- Assume: Symbols are equally likely.
- Question: What are the optimal decision regions?

4.1 Optimal Detection Receiver

The measurement $\mathbf{X}$ has a different joint probability density under each hypothesis:

$$\begin{aligned} H_0: &\quad f_{\mathbf{X}|H_0}(\mathbf{x}|H_0)P[H_0] \\ H_1: &\quad f_{\mathbf{X}|H_1}(\mathbf{x}|H_1)P[H_1] \\ \cdots &\quad \cdots \\ H_{M-1}: &\quad f_{\mathbf{X}|H_{M-1}}(\mathbf{x}|H_{M-1})P[H_{M-1}] \end{aligned}$$

Given that a particular $\mathbf{x}$ is measured, what is our method for deciding which hypothesis (about which symbol was sent) is true, in a way that minimizes the error? We can look to our previous lecture on binary decision. We minimized error by finding which joint probability of $\mathbf{x}$ and H_i for $i \in \{0, 1\}$ was highest. A similar derivation to that one would show that the probability of error is minimized by finding the i which joint probability of $\mathbf{x}$ and H_i for $i \in \{0, \ldots, M-1\}$ is highest. That is, we decide symbol i was sent if

$$f_{\mathbf{X}|H_i}(\mathbf{x}|H_i)P[H_i] > f_{\mathbf{X}|H_j}(\mathbf{x}|H_j)P[H_j] \text{ for all } j \neq i.$$

For this class, we'll usually consider the case of Gaussian additive noise and equally probable symbols. While symbols are sometimes not equally probable for $M = 2$ binary detection, it is very rare in higher M communication systems because it is easy for communications systems designers to encode the data so that each symbol is equally likely to be transmitted. If $P[H_0] = \cdots = P[H_{M-1}]$ then we only need to find the i that makes the likelihood $f_{\mathbf{X}|H_i}(\mathbf{x}|H_i)$ maximum, that is, maximum likelihood detection:

$$\text{Symbol Decision} = \arg\max_i f_{\mathbf{X}|H_i}(\mathbf{x}|H_i) \tag{17}$$

Here we have multivariate Gaussian measurement. For reasons we have not covered, the elements of vector $\mathbf{X}$ are uncorrelated and each have the same variance σ_W^2. This means that

$$f_{\mathbf{X}|H_i}(\mathbf{x}|H_i) = \frac{1}{(2\pi\sigma_W^2)^{K/2}} \exp\left[-\sum_{i=0}^{K-1} \frac{(x_k - a_{i,k})^2}{2\sigma_W^2}\right].$$

Since $\sum_i (x_k - a_{i,k})^2$ can be written as the squared Euclidean distance between two vectors $\mathbf{x}$ and $\mathbf{s}_i$, we can simplify by writing:

$$f_{\mathbf{X}|H_i}(\mathbf{x}|H_i) = \frac{1}{(2\pi\sigma_W^2)^{K/2}} \exp\left[-\frac{\|\mathbf{x} - \mathbf{s}_i\|^2}{2\sigma_W^2}\right].$$

When we want to solve (17) we can simplify, as we did in the binary decision case: 1) using the natural log to remove the exp; 2) removing any additive terms and multiplying out any terms that are not a function of i, and 3) using the square root. Note that the log and the $\sqrt{\cdot}$ are monotonically increasing functions and thus don't change the output of the argmin.

$$\begin{aligned}
\hat{i} &= \operatorname*{argmax}_i \left\{ \ln \frac{1}{(2\pi\sigma_W^2)^{K/2}} - \frac{\|\mathbf{x} - \mathbf{s}_i\|^2}{2\sigma_W^2} \right\} \\
\hat{i} &= \operatorname*{argmax}_i - \frac{\|\mathbf{x} - \mathbf{s}_i\|^2}{2\sigma_W^2} \\
\hat{i} &= \operatorname*{argmin}_i \|\mathbf{x} - \mathbf{s}_i\|^2 \\
\hat{i} &= \operatorname*{argmin}_i \|\mathbf{x} - \mathbf{s}_i\| \qquad (18)
\end{aligned}$$

Again: The short story is that we just find the $\mathbf{s}_i$ in the signal space diagram which is closest to $\mathbf{x}$.

5 Experimental SDR Transmitter and Receiver on POWDER

In this section we walk through and use a digital transmitter and receiver in Python. We use the python code to run a known packet transmission over the air on POWDER, and record the received signal at one or more receivers, and then to demodulate the received signal to (hopefully) correctly decode the transmitted data.

I would say that the most important thing to know about digital communications is that the main problem it solves is that while the data may be digital, the medium of transmission is real-valued and continuous-time, and worse – noisy and subject to interference. Our engineering solution of digital communications is largely a way to match the digital signal to this analog medium.

This section assumes the modulation is PSK/QAM.

5.1 Encoder

- Serial to Parallel (S/P). Each symbol conveys $\log_2 M$ bits, where M is the number of distinct symbols on the constellation diagram. For QPSK, $M = 4$. For 16-QAM, $M = 16$.
- Constellation. Each $\log_2 M$ bit string is mapped to a complex value.
- $N - 1$ zeros are inserted between each complex symbol value.

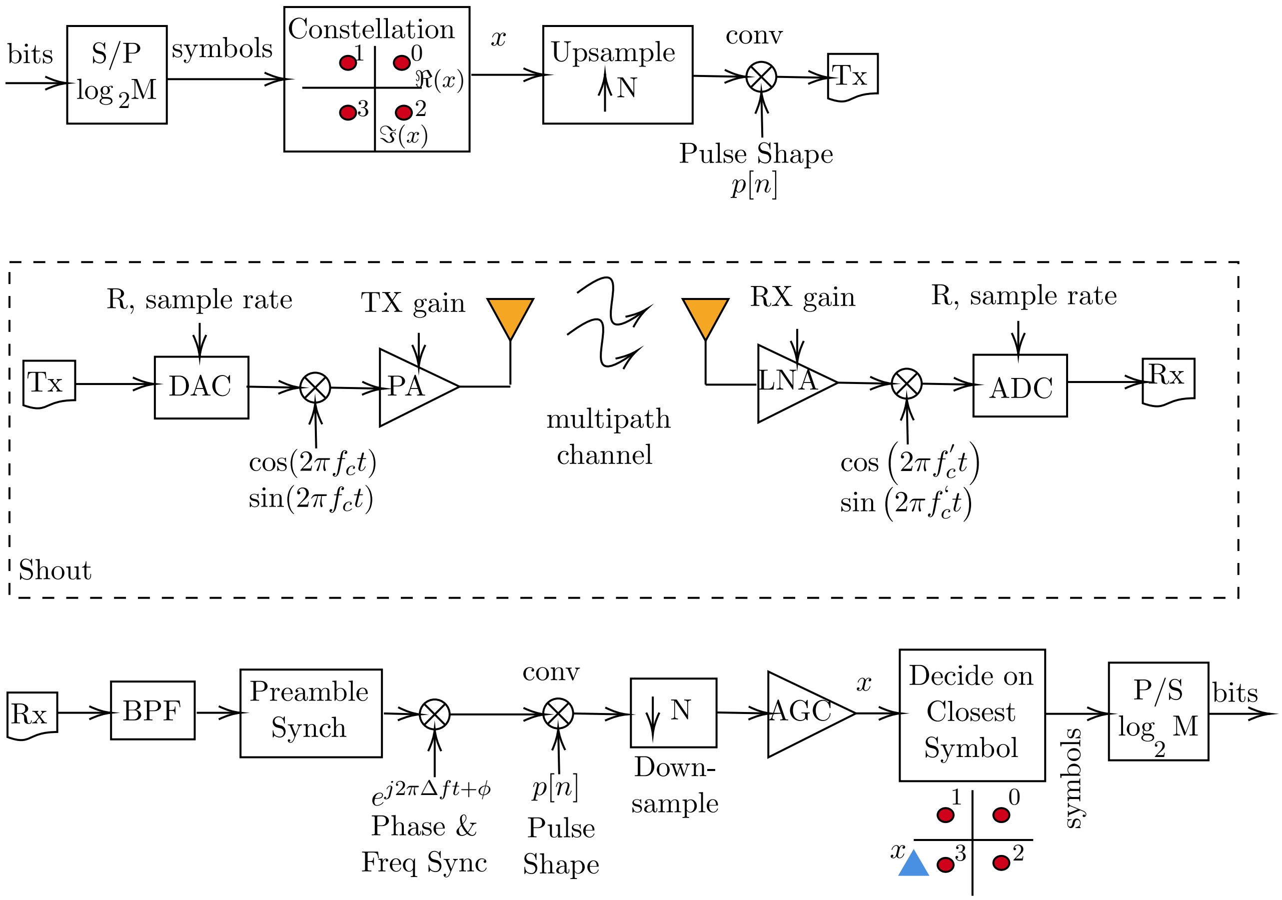

Figure 5: (Top row) Digital channel encoding, and (middle row) transmitting and receiving the modulated signal centered at a center frequency f_c, and (bottom row) receiver synchronization, demodulation / channel decoding.

- We convolve the upsampled signal with the pulse shape. In combination, the upsample and convolution makes the output the sum of complex-amplitude scaled pulse shapes, each time-delayed by a different multiple of the symbol period.
- We write the complex baseband signal to a .iq file.

5.2 Shout and Channel

Shout uses the UHD driver to convert the complex baseband digital data in the .iq file to transmit it from a NI SDR. It does this by converting each digital complex sample value to two analog value (one I and one Q). It then multiplies the analog I signal with a cosine at the carrier frequency f_c. It also multiplies the analog Q signal with a sine at the carrier frequency f_c. It sends this signal to the antenna. By having a voltage signal in the antenna conductor, it creates an electromagnetic wave that is propagated from it. The electromagnetic wave that passes by the receive antenna induces a current in its conductor, within the band it is efficient at receiving. The voltage signal is amplified (multiplied) by a low noise amplifier (LNA). Then by multiplying the LNA output by a cosine at the carrier frequency f_c, it effectively downconverts the incoming signal to the I baseband signal. by multiplying the LNA output by a sine at the carrier frequency f_c, it effectively downconverts the incoming signal to the Q baseband signal. These I and Q signals are both sampled and saved to file as complex values.

5.3 Synch and Decoder

In our python receiver, we operate many of steps a digital receiver must do.

- Bandpass filter (BPF): We sharply filter out noise from out of the frequency band so that it doesn't impact our synchronization operations.
- We do preamble synchronization in time, frequency, and phase. The received signal is time delayed. It shows up at a frequency shift of $\Delta f = f_c - f_c'$, in the realistic situation that the TX and RX oscillators are not perfectly aligned. (Ours are, typically! But this wouldn't be true in real life, or if our white rabbit system is not working.) I do this in this code by

  1. creating a copy of the known preamble signal
  2. cross-correlating the RX signal with the preamble signal; and
  3. looking at the max magnitude peak in that cross-correlation.
  4. The phase of that peak is the phase offset.

  The frequency offset of the preamble is measured by an Fourier transform of the preamble, and I leave it to the code to provide more specifics.

- We then convolve with the same pulse shape used in the transmitter. Technically, the matched filter or optimum filter, has an impulse response that is the time-reversed pulse shape, but our pulse shape is symmetric about zero, so it makes no difference whether we time-reverse it or not.
- Now, every Nth sample has all the information we need, it is the scaled amplitude of the complex voltage sent to convey the transmitted symbol (corresponding to one of the symbols on the constellation diagram)

- Since the channel sharply attenuated the signal, we use an AGC to multiply the signal amplitudes back to a range that matches the constellation diagram.
- For each sample $\mathbf{x}$, we find the closest symbol on the constellation diagram.
- That closest symbol corresponds to k bits, we add them to the bits out vector.

6 Modulation Probability of Bit Error

Given that we have a link with a certain bandwidth and SNR, what modulation (and coding scheme) can be used to maximize bit rate but still have reliable communications? This is a key question we discuss in this section. In general, this is a large topic, and we can't cover it fully in this class. It relates to many system tradeoffs of which a system designer should be aware and knowledgeable about, between data rate, bandwidth efficiency, and fidelity (data correctness).

This introductory coverage of the topic will focus on a few common modulation types. We are not covering error correction coding schemes, because of the added complexity.

6.1 Q Function

The probability that a unit-variance, zero mean Gaussian random variable X exceeds some value x is so common in digital communications, it is given its own name, $Q(x)$,

$$Q(x) = P\left[X > x\right] = \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-w^2/2} dw$$

More familiar functions (that exists on most calculators and function libraries) are the $\mathrm{erf}(x)$ and the $\mathrm{erfc}(x) = 1 - \mathrm{erf}(x)$ functions:

$$Q(y) = \frac{1}{2}\mathrm{erfc}\left(\frac{y}{\sqrt{2}}\right). \tag{19}$$

You should keep handy this Python function definition of $Q(y)$ and its inverse, $\mathrm{Q}^{-1}(y)$:

```
from scipy.special import erfinv, erfc
from math import sqrt

def Q(y):
    return 0.5*erfc(y/sqrt(2))

def Qinv(y):
    return sqrt(2)*erfinv(1-2*y)
```

6.2 Summary of Probability of Bit Error Formulas

Using multiple methods, one can derive formulas for the probability of bit error as a function of the energy per bit and noise energy. The dependence is always a function of the ratio of the two, $\frac{\mathcal{E}_b}{N_0}$. See Table 6.2. There are exact expressions for BPSK and QPSK, but the remaining expressions are approximations.

Recall in our last lecture, we derived that

$$\frac{\mathcal{E}_b}{N_0} = \frac{C}{N} \cdot \frac{B}{R_b}. \tag{20}$$

Name	$P\,[\textbf{bit error}]$
BPSK	$= \mathrm{Q}\left(\sqrt{\frac{2\mathcal{E}_b}{N_0}}\right)$
QPSK	$= \mathrm{Q}\left(\sqrt{\frac{2\mathcal{E}_b}{N_0}}\right)$
8-PSK	$\approx \frac{2}{3}\mathrm{Q}\left(\sqrt{0.8787\frac{\mathcal{E}_b}{N_0}}\right)$
Square 16-QAM	$\approx 0.75\mathrm{Q}\left(\sqrt{0.8\frac{\mathcal{E}_b}{N_0}}\right)$
Square 64-QAM	$\approx \frac{7}{12}\mathrm{Q}\left(\sqrt{\frac{2}{7}\frac{\mathcal{E}_b}{N_0}}\right)$
Square 256-QAM	$\approx \frac{15}{32}\mathrm{Q}\left(\sqrt{\frac{24}{255}\frac{\mathcal{E}_b}{N_0}}\right)$
Square M-QAM	$\approx \frac{4}{\log_2 M}\frac{(\sqrt{M}-1)}{\sqrt{M}}\mathrm{Q}\left(\sqrt{\frac{3\log_2 M}{M-1}\frac{\mathcal{E}_b}{N_0}}\right)$

Table 1: Exact or approximate probability of bit error formulas for several modulations.

The bandwidth is a function of both the pulse shape and the symbol rate R_s. The bandwidth of the SRRC pulse shape was given in (16), and we can write it as:

$$B = (1+\alpha)R_s = (1+\alpha)\frac{R_b}{\log_2 M} \tag{21}$$

In other words,

$$\frac{B}{R_b} = \frac{1+\alpha}{\log_2 M} \tag{22}$$

Thus

$$\frac{\mathcal{E}_b}{N_0} = \frac{C}{N}\frac{1+\alpha}{\log_2 M}. \tag{23}$$

In the big picture:

- We want a very low probability of bit error, depending on the application, the desired $P\,[\text{bit error}]$ might be from 10^{-2} to 10^{-6}.
- The Q function is a decreasing function: as the argument gets larger, the probability of bit error decreases.
- A higher $\frac{\mathcal{E}_b}{N_0}$ results in a lower $P\,[\text{bit error}]$.
- A higher constant (like 2 for BPSK and QPSK) multiplying $\frac{\mathcal{E}_b}{N_0}$ means that the $P\,[\text{bit error}]$ will reduce faster as a $\frac{\mathcal{E}_b}{N_0}$ increases. But as M goes up in M-QAM, the constant multiplier decreases.
- In addition, as M goes up, R_b increases, so $\frac{\mathcal{E}_b}{N_0} = \frac{C}{N}\cdot\frac{B}{R_b}$ decreases.

In short, if we use more bandwidth-efficient modulation to increase the bit rate on our given bandwidth, we need more energy per bit, and thus more received power, in order to receive that data at the same level of reliability.

Example: Using the $P\,[\textbf{bit error}]$ formulas

1. What is the probability of bit error using 16 QAM when $\frac{\mathcal{E}_b}{N_0} = 20$ (linear)?
2. What is the $\frac{\mathcal{E}_b}{N_0}$ required to achieve a probability of error of 10^{-6} in QPSK?
3. For the problem in (2.), what is the SNR required? Assume SRRC pulses with $\alpha = 0.2$.

Solution:

1. What is the probability of bit error using 16 QAM when $\frac{\mathcal{E}_b}{N_0} = 20$ (linear)?

$$\begin{aligned} P\,[\text{bit error}] &\approx 0.75\mathrm{Q}\left(\sqrt{0.8(20)}\right) = 0.75\mathrm{Q}\left(\sqrt{16}\right) \\ P\,[\text{bit error}] &\approx 0.75(3.167\times 10^{-5}) = 2.375\times 10^{-5} \end{aligned}$$

2. What is the $\frac{\mathcal{E}_b}{N_0}$ required to achieve a probability of error of 10^{-6} in QPSK?

$$\begin{aligned} 10^{-6} &= \mathrm{Q}\left(\sqrt{\frac{2\mathcal{E}_b}{N_0}}\right) \\ \frac{\mathcal{E}_b}{N_0} &= \frac{1}{2}\left[\mathrm{Q}^{-1}\left(10^{-6}\right)\right]^2 \\ \frac{\mathcal{E}_b}{N_0} &= \frac{1}{2}\left[4.75\right]^2 = 11.29 \end{aligned}$$

3. For the problem in (2.), what is the SNR required? From (23),

$$\frac{C}{N} = \frac{\mathcal{E}_b}{N_0}\frac{\log_2 M}{1+\alpha} = 11.29\frac{2}{1+0.2} = 18.8$$

References

[1] L. W. Couch. *Digital and Analog Communication Systems.* Pearson, 7th edition, 2007.

[2] M. Rice. *Digital Communications: a Discrete-Time Approach.* Pearson Prentice Hall, 2009.