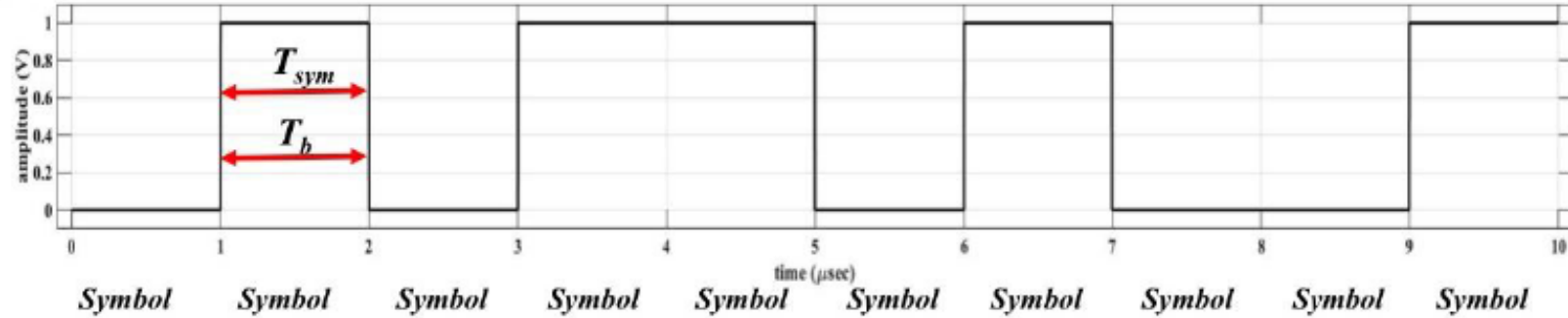


Information (bits):

0 1 0 1 1 0 1 0 0 1

Baseband signal  
(voltage pulses):



$$R_{sym} = \frac{1}{T_{sym}} \qquad R_b = \frac{1}{T_b}$$

In this example, since  $T_{sym}$  is 1 μsec, then  $R_{sym} = 1/10^{-6} = 1 \times 10^6$  symbols/sec. And, since each symbol carries 1 bit of information (that is, 1 bit/symbol), the bit rate is  $1 \times 10^6$  bits/sec = 1 Mbps. In general, the symbol rate and bit rate are related by:

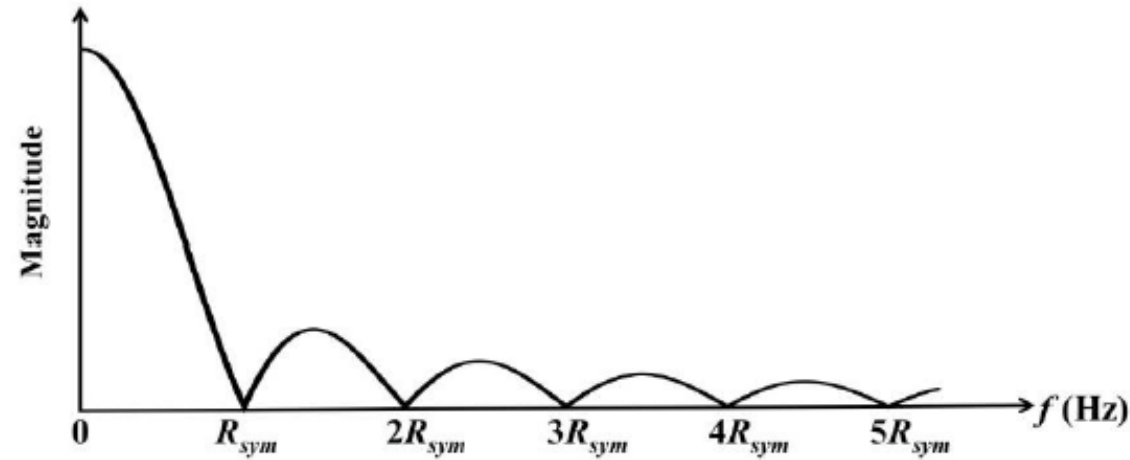
$$R_b = NR_{sym}$$

$$N = \log_2 M$$

N=number of bits per symbol

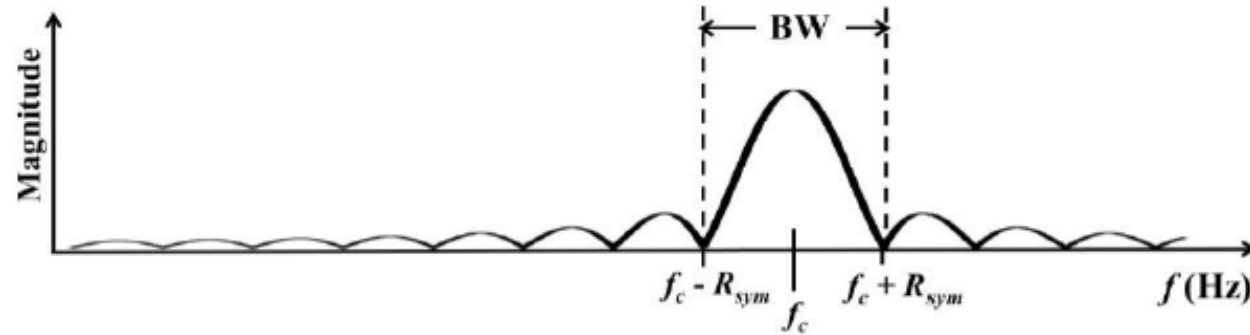
M= number of symbols

Frequency spectrum for random voltage pulses (a baseband signal—primarily low frequencies)



For example, if the symbol rate was 500 symbols/sec (so the bit rate was 500 bps), then the frequency content's magnitude would be zero at 500 Hz, 1000 Hz, etc. This plot of frequency content is much different than that of a signal composed of sinusoids! There are no spikes! Nevertheless, most of the frequency content is at very low frequencies. The frequency content does continue out to an infinite frequency, although the magnitude drops dramatically at higher frequencies. In a perfect world, we'd say the bandwidth of voltage pulses approaches  $\infty$  Hz, but for digital signals, we'll use the *null-bandwidth* as our calculated bandwidth. The null-bandwidth is defined as the amount of the frequency spectrum (in Hz) from the maximum magnitude (which here occurs at 0 Hz) to where the spectrum first goes to a magnitude of 0 (called a *null*, here at  $R_{sym}$  Hz). The bandwidth is given by:

$$BW = f_2 - f_1 = R_{sym} - 0 = R_{sym} [Hz]$$

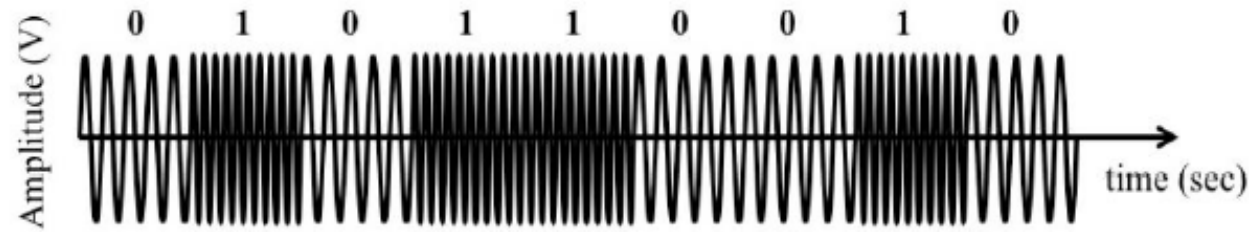


Frequency Spectrum for modulated voltage pulses  
(now a band pass signal—primary frequency content at a much higher frequency than the voltage pulses, centered at the carrier frequency)

Like in analog amplitude modulation, the information signal's frequency spectrum is shifted up by  $f_c$  Hz, and there is a mirror image of the frequency content on the left side of  $f_c$ . The transmission bandwidth (using the null-bandwidth definition along with the fact that there is now a null to the left and right of the carrier frequency) is

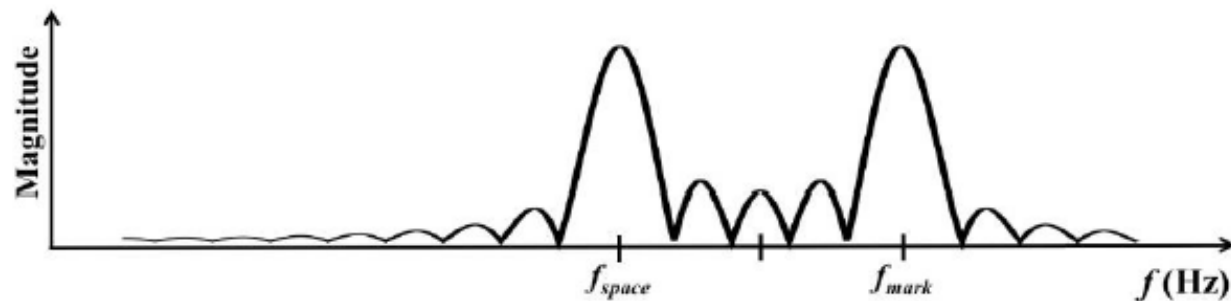
$$BW = f_2 - f_1 = (f_c + R_{sym}) - (f_c - R_{sym}) = 2R_{sym} [Hz]$$

**3.1 Frequency Shift Keying (FSK)** Frequency-shift keying (FSK) is a frequency modulation scheme in which digital information is transmitted through discrete frequency changes (shifts) of a carrier wave. The simplest form of FSK is Binary FSK (BFSK), in which a carrier's frequency is shifted to a low frequency or a high frequency to transmit 0s and 1s. The plot below shows a sample FSK signal along with the associated bits.

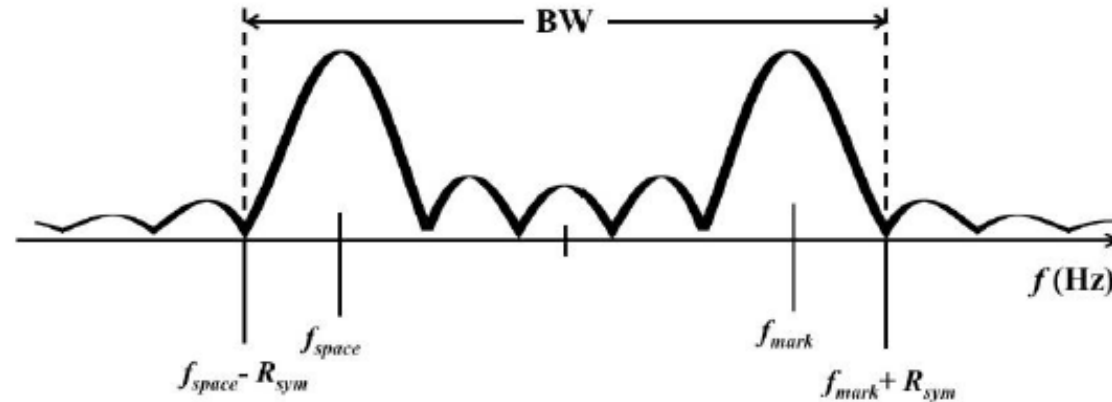


FSK was used “back in the day” with dial-up modems to connect your home computer to your Internet service provider over your analog phone. With a modem, a 0-bit was represented with a lower frequency carrier of 1070 Hz and a 1-bit was represented with a higher carrier frequency of 1270 Hz. The lower frequency, binary 0, was called the “space” frequency while the higher frequency, binary 1, was called the “mark” frequency. The terms mark/space were a throwback to the days of Morse code or flashing light communications.

In the frequency domain, we use two carrier frequencies and consider FSK to be two different digital transmissions, one at the mark frequency (the higher frequency) and one at the space frequency (lower frequency). The resulting FSK frequency plot would look like the following. This figure is two copies of the frequency plot on the previous page, one centered at  $f_{mark}$  and one centered at  $f_{space}$ .



To determine the bandwidth for FSK modulation, we take a closer look at the frequency spectrum around the mark and space frequencies. We use the null-bandwidth definition to compute the bandwidth as shown below.



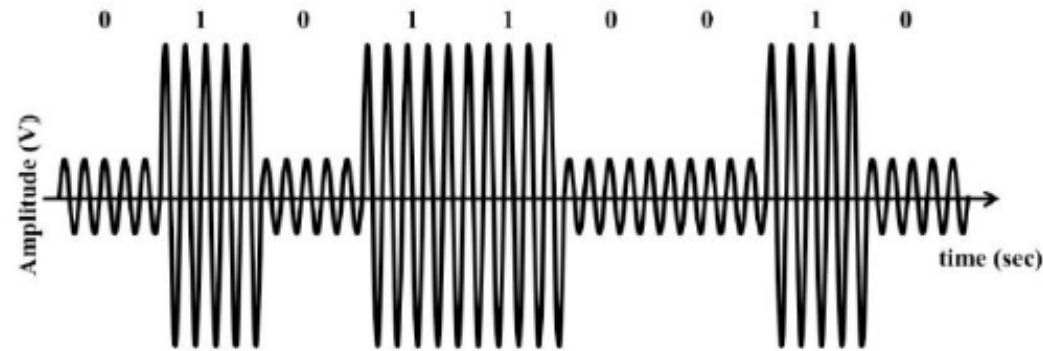
In the figure, the bandwidth effectively runs from the first null to the left of  $f_{space}$  to the first null to the right of  $f_{mark}$ . Mathematically, we can compute the FSK bandwidth as:

$$BW = f_2 - f_1 = (f_{mark} + R_{sym}) - (f_{space} - R_{sym}) = f_{mark} - f_{space} + 2R_{sym} [Hz]$$

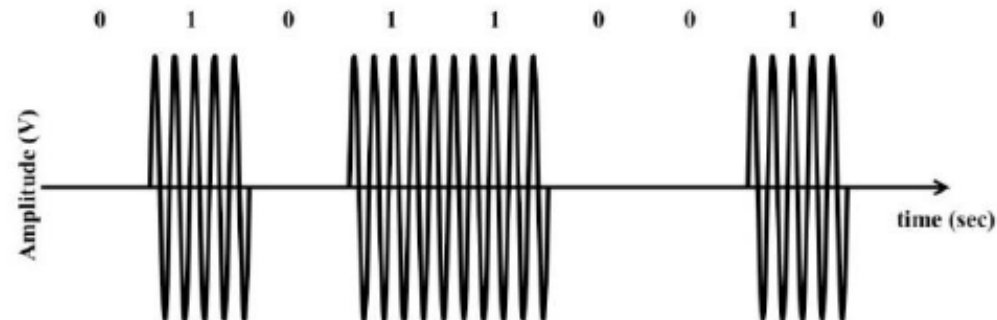
Or, for FSK, since the symbol rate is equal to the bit rate, we could also use the equation:

$$BW = f_2 - f_1 = (f_{mark} + R_{sym}) - (f_{space} - R_{sym}) = f_{mark} - f_{space} + 2R_b [Hz]$$

**3.2 Amplitude Shift Keying (ASK) and On-Off Keying (OOK)** Amplitude Shift Keying is a form of amplitude modulation that represents digital data as shifts in the amplitude of a carrier wave: for example, small amplitude for a 0-bit, and larger amplitude for a 1-bit. We have seen what an ASK signal looks like in a previous chapter, repeated below.



The simplest digital modulation scheme is a form of ASK called *On-Off keying* (OOK). This is analogous to flashing light communication. In OOK, a carrier is transmitted for a 1-bit and nothing is transmitted for a 0-bit; this is the same as saying that the smaller ASK amplitude is 0.



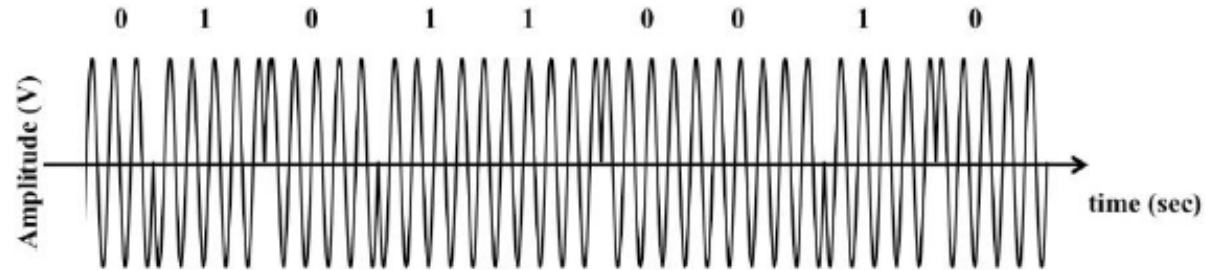
Note that in all forms of ASK, the frequency and phase of the carrier are the same for both symbols; it is the amplitude that changes.

**3.3 Binary Phase Shift Keying (BPSK)** Phase shift keying (PSK) is a form of phase modulation where the carrier's phase shifts to one of a finite set of possible phases based on the bits that are input. For binary phase shift keying (BPSK), the carrier phase is shifted between one of only two phases (typically  $0^\circ$  and  $180^\circ$ ) depending on whether a 0-bit or a 1-bit is being transmitted. For example:

0-bit: the symbol transmitted is  $V_c \cos(2\pi f_c t)$ .

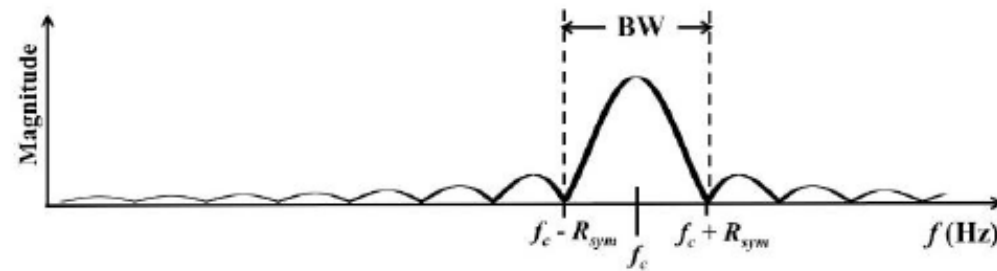
1-bit: the symbol transmitted is  $V_c \cos(2\pi f_c t + 180^\circ) = -V_c \cos(2\pi f_c t)$

Using  $\pm \cos(2\pi f_c t)$  as the symbols is not the only type of BPSK modulation there is, as long as the phases are  $180^\circ$  apart...for example, you could also use  $\pm \sin(2\pi f_c t)$ . A sample BPSK transmission is shown in the following figure. As you look at the figure, you'll notice that for us, it is probably harder to see the transitions in phase for 0-bits and 1-bits, but rest assured that a BPSK receiver can do it easily.



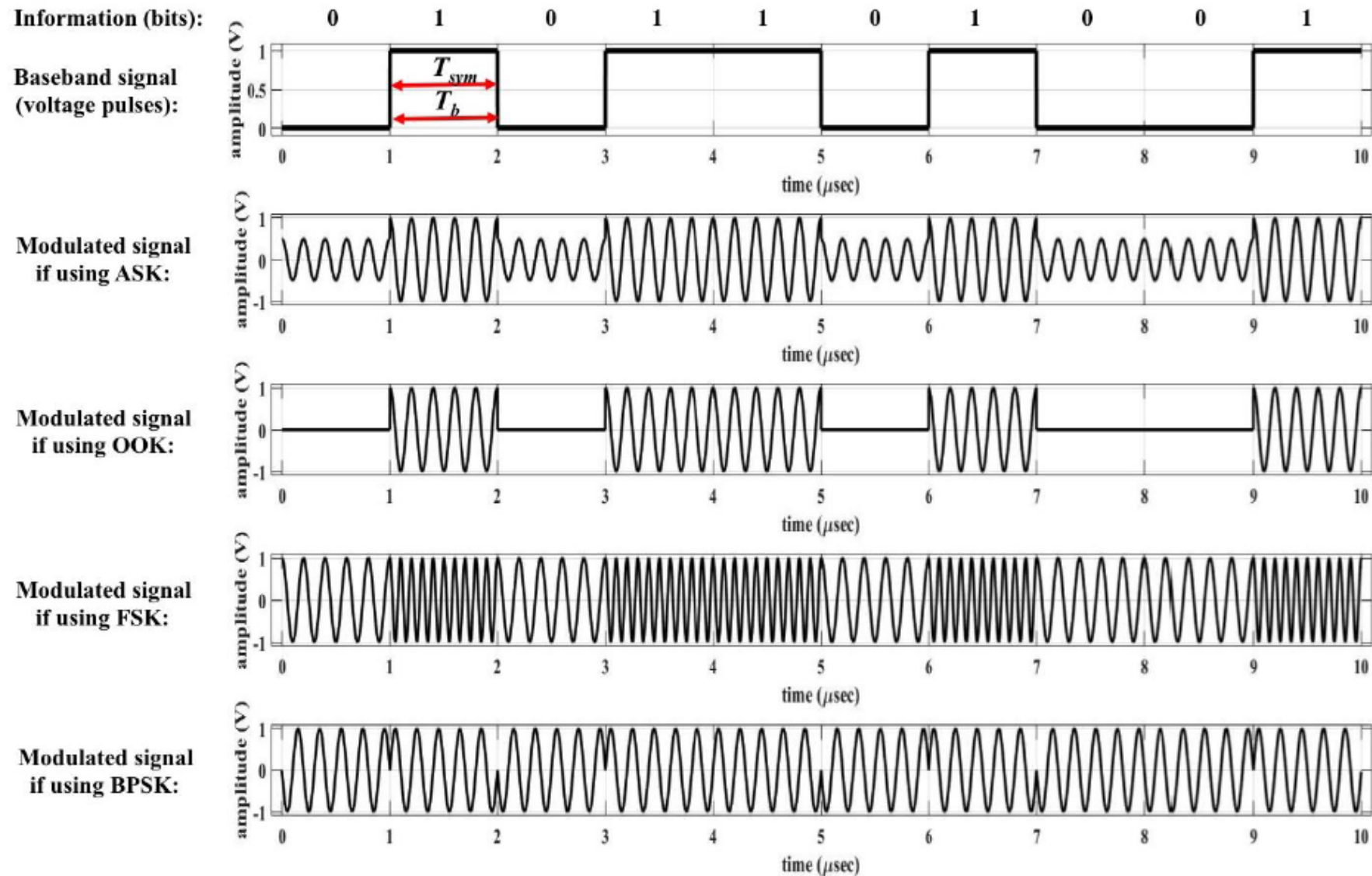
It is important to point out that in PSK, the amplitude of all output symbols is the same; it is the phase of the output symbols that are different, while in ASK, both symbols have the same phase, but different amplitudes.

The bandwidth associated with ASK, OOK, or BPSK is what we have seen before,  $BW = 2R_{sym}$ , as shown in the figure below. In the case of ASK, OOK, or BPSK, since  $N = 1$  bits/symbol,  $BW = 2R_{sym} = 2R_b$ . For example, for ASK, OOK, or BPSK, if the symbol rate is 600,000 symbols/sec, the bitrate is 600 kbps, and so the bandwidth is  $2(600,000) = 1.2$  MHz.



Let us now summarize the types of binary digital modulation that have been introduced so far. On a wire, the symbols take the form of voltage pulses, which are, for example, a high pulse for a 1-bit and a low (or no) pulse for a 0-bit. In FSK, ASK, OOK or BPSK, the symbols take the form of a high frequency carrier that has its frequency or amplitude or phase altered based on whether a 0-bit or a 1-bit is being transmitted. In binary modulation, the number of symbols that can be transmitted ( $M$ ) is two ( $M = 2$ ) and each symbol represents one bit of data ( $N = 1$  bit per symbol). For binary modulation, the time duration of a bit is the same as the time duration of a symbol ( $T_b = T_{sym}$ ). The following figure depicts the relationship between bits and symbols for voltage pulses (the baseband signal), and FSK, ASK, OOK and BPSK (the modulated signals).



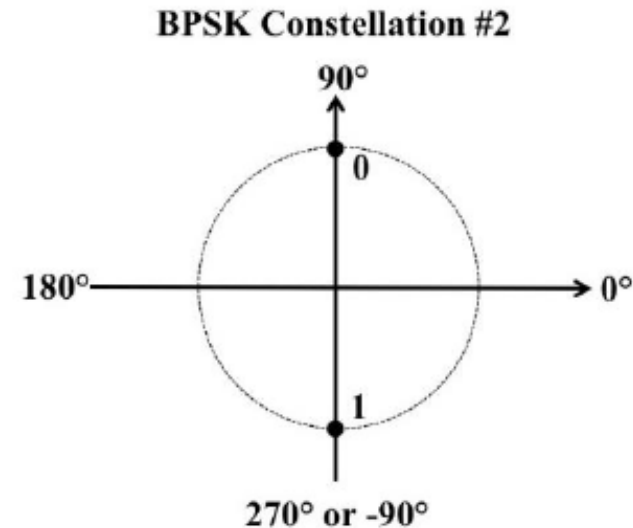
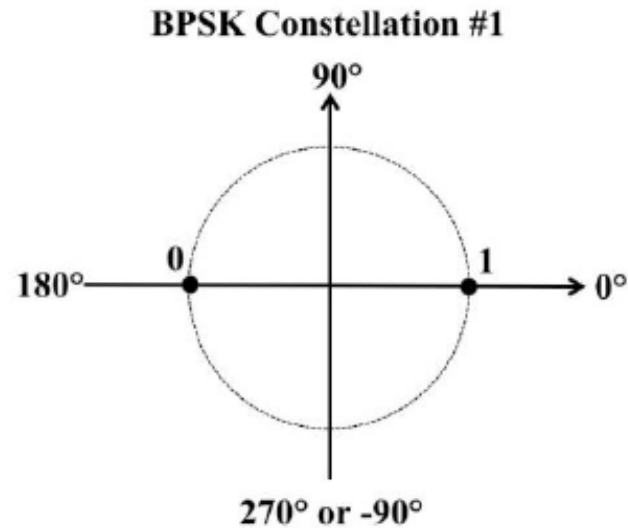


Up to this point we have discussed binary digital modulation ( $M=2$ ) with one bit per symbol ( $N=1$ ), which means that at any time, only one of two possible symbols would be transmitted. But it is possible to have a modulation scheme with more than two symbols. This means that with each symbol, more than one bit is transmitted at a time. These types of modulation are referred to as  $M$ -ary digital modulation.

#### 4. M-ary Digital Modulation

Before launching into more complicated digital modulation, we'll introduce a graphical way to relate what the output symbols are, and the bits that each symbol represents. This is called a *constellation diagram*. A constellation diagram is a plot of the phase and relative amplitude of the output symbols for a digital modulation system, in polar coordinates. In terms of the symbol's phase,  $0^\circ$  is along the positive x-axis, and phase increases as you move counterclockwise around the x-y plane. The symbol's relative amplitude is measured as distance from the origin of the plot. The possible output symbols are represented with large dots, and adjacent to them are the bits they represent. Symbols that have the same amplitude are the same distance from the origin (you can think of them as laying on the same circle around the origin). All symbols with the same phase would fall on the same line segment that originates at the origin and goes out at a certain angle.

For example, here are two possible BPSK systems' constellation diagrams. In BPSK, the output symbols both have the same amplitude (both of the symbols are equidistant from the origin), but their phases are  $180^\circ$  apart. There are other possible combinations of two carrier phases that might be used (such as  $+90^\circ$  and  $-90^\circ$ ), but the actual constellation used is not important, as long as the transmitter and receiver use the same constellation.



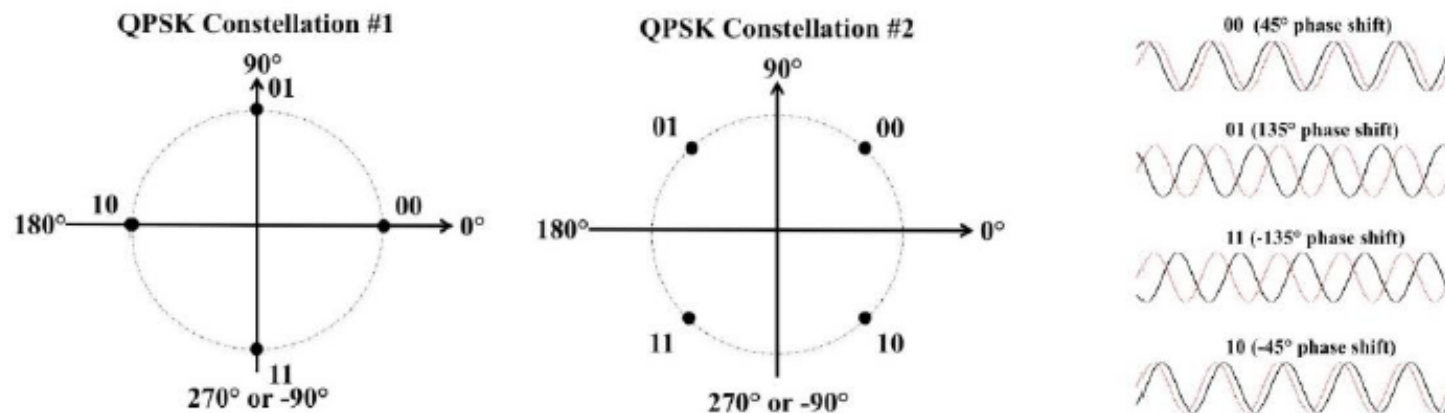
Note that BPSK transmits 1 bit per symbol, so only one bit value is placed next to each symbol.

If it is desired to get the information from the transmitter to the receiver faster, we could increase the number of bits per second (bps) that are transmitted. The cost of increasing the bitrate (besides requiring more complex components) is that it increases the transmission bandwidth: recall that for ASK, OOK or BPSK,  $BW = 2R_{sym} = 2R_b$ , and from Chapter 19, that bandwidth can be expensive! Is there a way to transmit a higher bitrate but using a smaller transmission bandwidth? The answer is yes, using  $M$ -ary digital modulation.

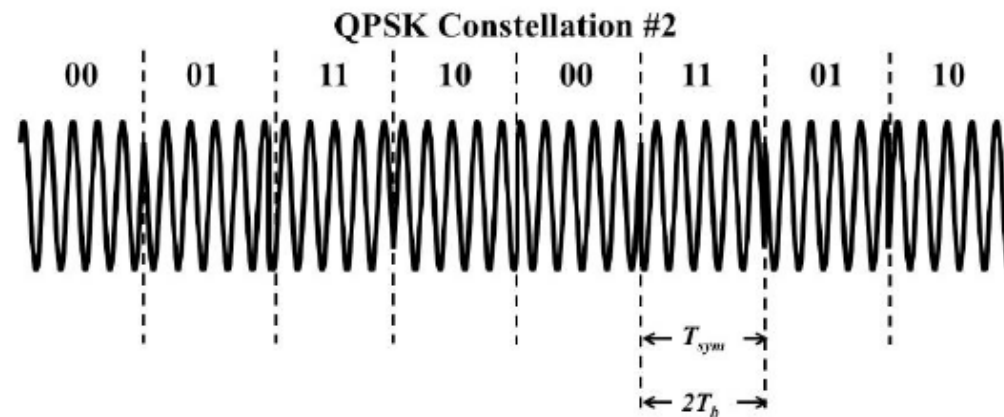
In  $M$ -ary modulation, we can preserve bandwidth if we **keep the symbol rate the same and increase the number of bits per symbol**. For example, instead of transmitting just 2 possible phase shifts ( $0^\circ$  and  $180^\circ$ ), we could transmit one of 4 possible phase shifts per symbol. This is called quadrature phase shift keying (QPSK).

**4.1 Quadrature Phase Shift Keying (QPSK)** In QPSK, there are **4 symbols** ( $M = 4$ ) and there are **2 bits per symbol** ( $N = 2 = \log_2 M$ ). Two of the possible constellation diagrams for QPSK are shown in the following figure<sup>1</sup>, and the four symbols from QPSK Constellation #2 are shown to the right of this constellation. The carrier with a phase of  $0^\circ$  is plotted in a dashed line with each symbol for reference. The four symbols in the right-hand constellation are:

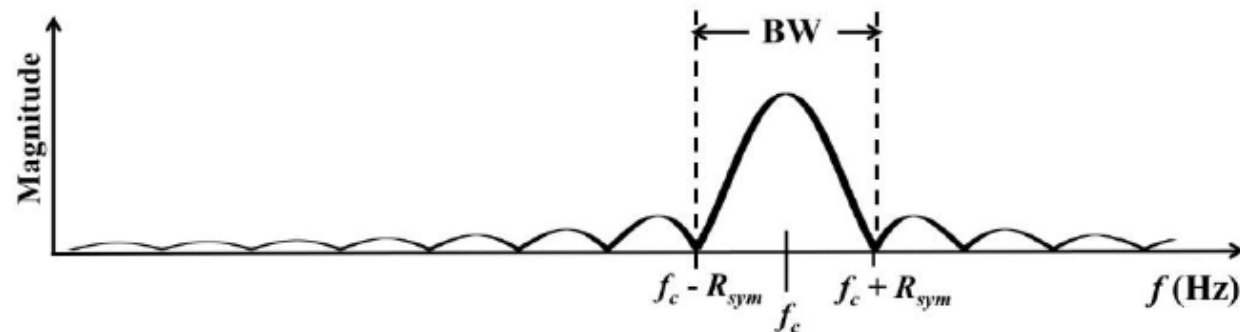
$V_c \cos(2\pi f_c t + 45^\circ)$ ,  $V_c \cos(2\pi f_c t + 135^\circ)$ ,  $V_c \cos(2\pi f_c t - 135^\circ)$  and  $V_c \cos(2\pi f_c t - 45^\circ)$ .



The following figure is a plot of the use of QPSK constellation #2 in the previous figure to transmit the bit stream 0001111000110110. Also shown is the bit duration, and the symbol duration for QPSK.



The frequency spectrum for  $M$ -ary modulation schemes is shown in the figure below, which is the same one that appeared earlier in the chapter for ASK, OOK and BPSK. But if the bit rate is constant, the benefit of transmitting more than one bit in a symbol can be seen in the fact that the nulls are closer to the carrier frequency, since  $R_{sym} = R_b / N$ .



From the figure, it is seen that the bandwidth for QPSK ( $N = 2$ ) is given by

$$BW = 2R_{sym} = \frac{2R_b}{N} = 2\left(\frac{R_b}{2}\right) = R_b \text{ Hz.}$$

For example, if bitrate is 600 kbps,  $BW = 2(600,000)/2 = 600$  kHz, half that of ASK, OOK or BPSK!

## 4.2 M-ary PSK

We can further increase the number of bits per symbol by increasing the number of possible phase shifts. The  $M$  in  $M$ -ary refers to the number of symbols. Consider the 8-PSK constellation to the right (one of many possible 8-PSK constellations<sup>2</sup>).

How many bits per symbol are transmitted? There are 8 symbols ( $M = 8$ ), so  $N = \log_2 M = \log_2 8 = 3$  bits/symbol. This is also evident from the diagram because the three bits associated with each symbol appear next to the symbol.

What is the bandwidth for 8-PSK? Since  $N=3$  bits/symbol, bandwidth is given by

$$BW = 2R_{sym} = \frac{2R_b}{N} = \frac{2R_b}{3}.$$

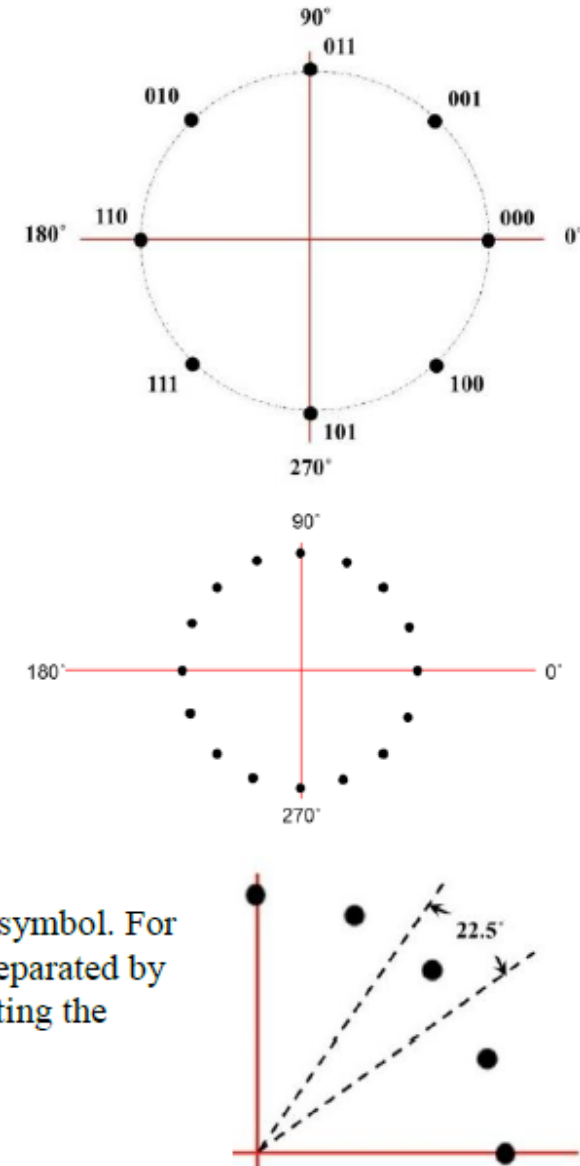
For example, if the bitrate is 600 kbps, bandwidth for 8-PSK is  $BW = 2(600,000)/3 = 400$  kHz, even less than for QPSK!

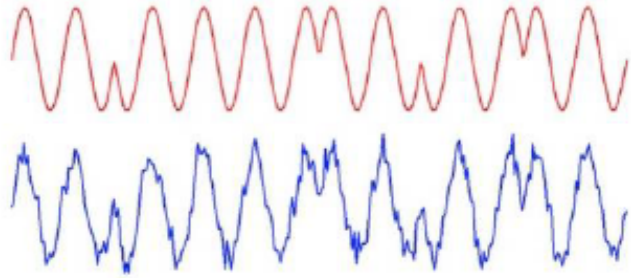
We could further increase to 4 bits/symbol using 16-PSK. Here,  $M = 16$  and  $N = 4$  bits/symbol. A 16-PSK constellation is shown to the right, where each phase is separated by  $360^\circ/16 = 22.5^\circ$ . More complex  $M$ -ary PSK modulation is possible: 32-PSK, 64-PSK, etc., but it becomes more susceptible to noise as the symbols get closer together. As a reminder, for PSK, all of the symbols have the same carrier frequency and amplitude; it is their phase that is different. For that reason, on a PSK constellation diagram, all of the symbols appear on a circle about the origin.

To demodulate any type of PSK, a receiver must determine the phase of the received symbol. For 16-PSK, the receiver must determine the phase within  $\pm 11.25^\circ$ , since the phases are separated by  $22.5^\circ$ . A portion of the constellation diagram for 16-PSK is shown to the right, indicating the wedge of phase values that separates one of the symbols from the adjacent symbols.

### Noise Effects

Recall that the number one most limiting factor in communication systems is noise. In all transmissions, the received signal will be degraded by noise. The following figure shows a BPSK signal and the same signal corrupted by noise. You might imagine that it is harder for a receiver to determine the correct phase (correct symbol) that was transmitted for the noisy signal.

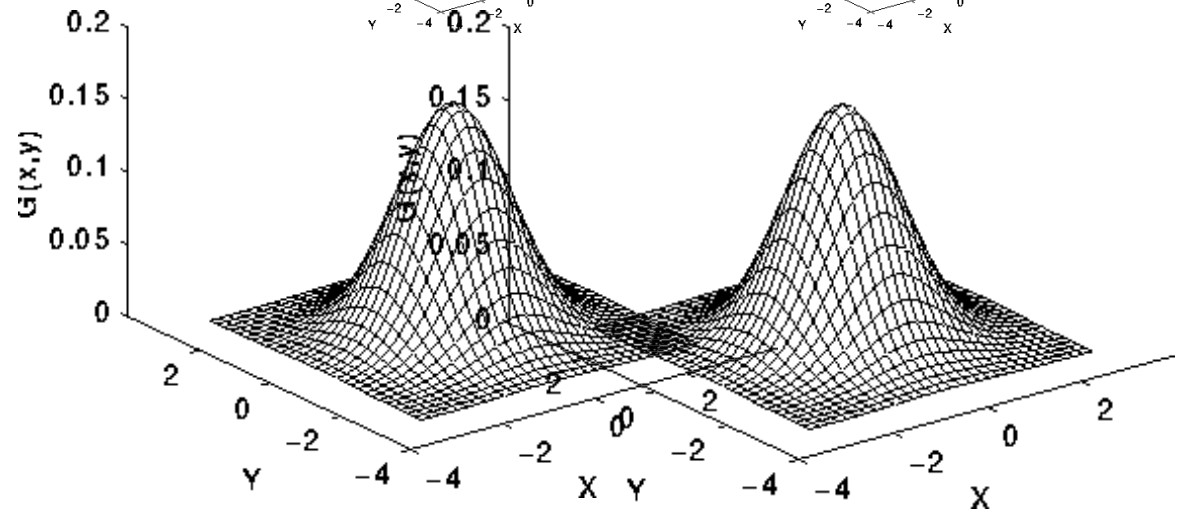
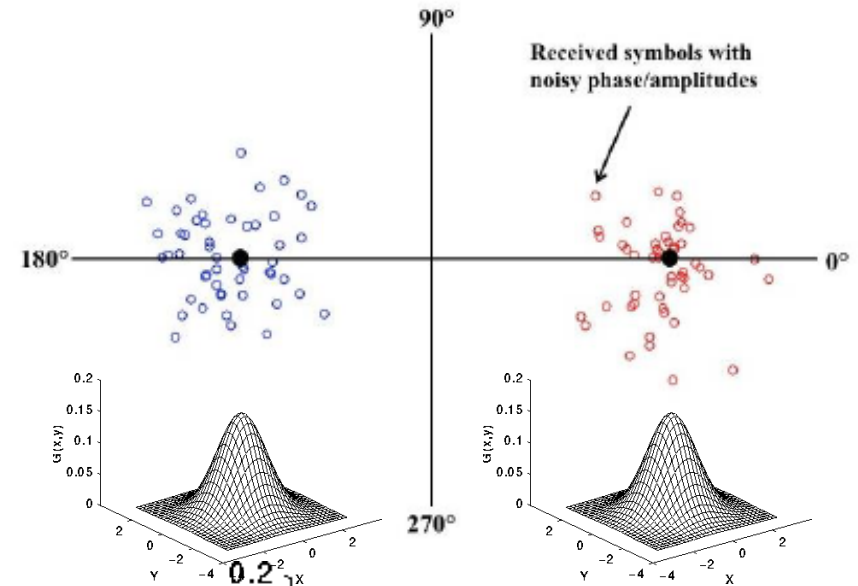




**BPSK Signal**

**BPSK Signal + Noise**

This noise corruption can be depicted in the constellation diagram to the right, where the two transmitted BPSK symbols are indicated in the two large black circles (phase =  $0^\circ$  and phase =  $180^\circ$ ), and some noisy received symbols are the smaller red and blue circles.

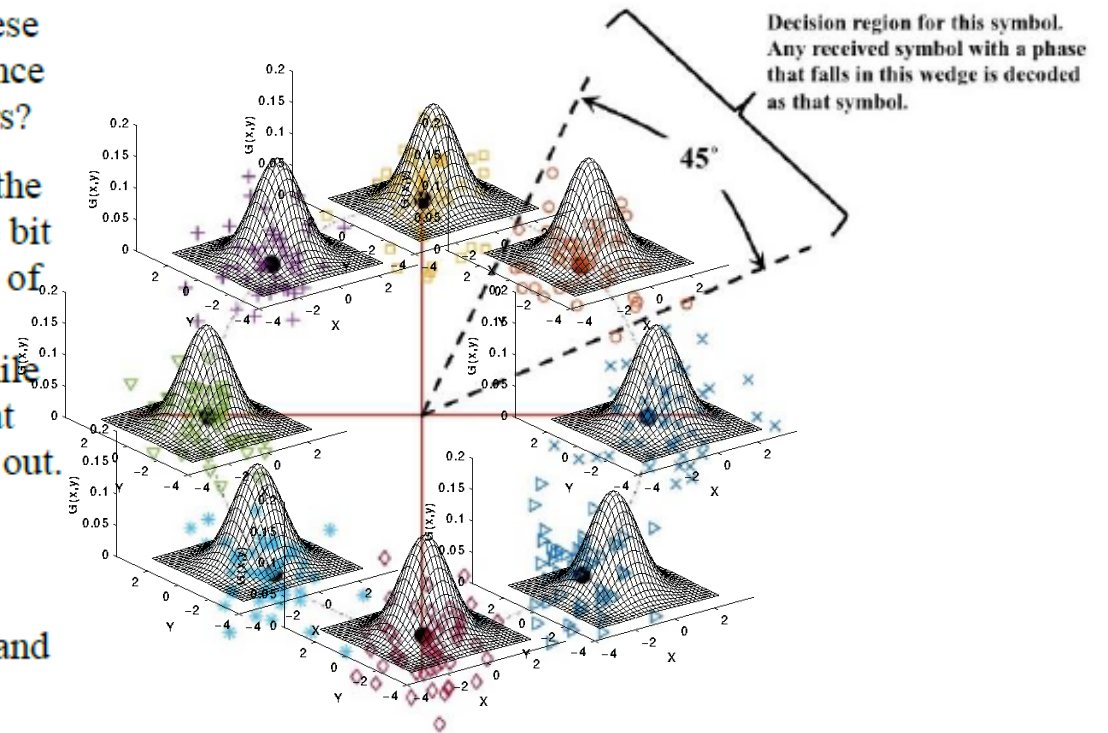


A BPSK receiver must make a decision to determine the phase of a received signal to determine the corresponding bit. If the noise is severe enough, a receiver might make a mistake and decide that it had received a 0-bit when it actually received a 1-bit. These are called *bit errors*. Now, consider the same noise in the presence of an 8-PSK signal. Is it easier for the receiver to make bit errors?

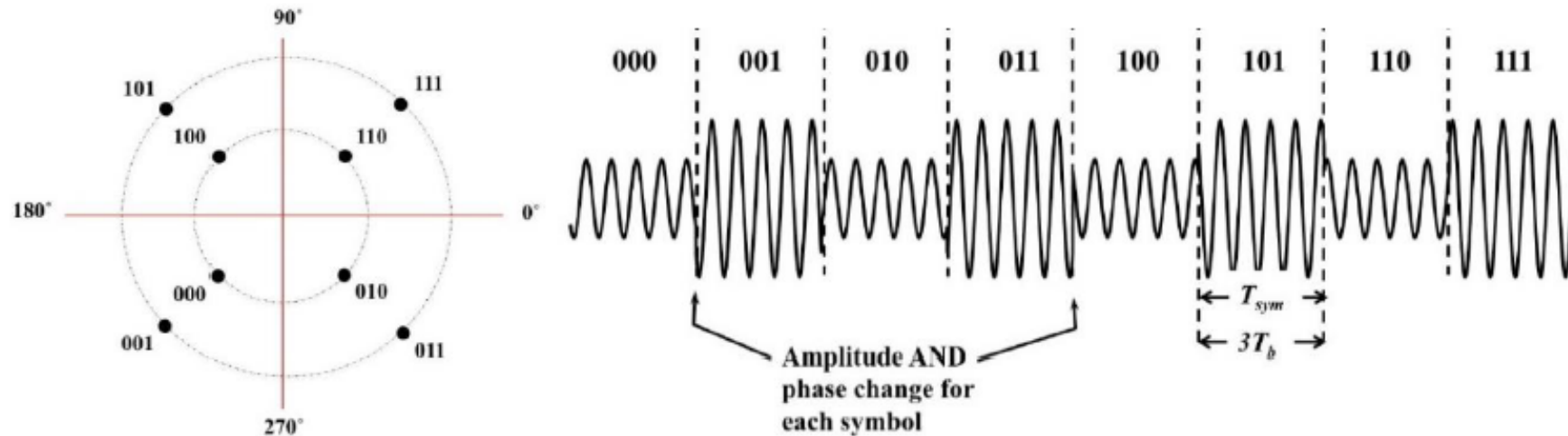
The answer is yes, since more phases are used in PSK, making the symbols closer together makes it easier for the receiver to make bit errors (see the figure to the right). But, of course, the advantage of more symbols is a narrower bandwidth, if the bit rate is held constant. There is a way to use more symbols in modulation while reducing the chances of making bit errors; by using symbols that have different amplitudes AND phases so they are more spread out.

### 4.3 Quadrature Amplitude Modulation (QAM)

In order to increase the distance between symbols in a constellation, another option is to modulate both the amplitude and the phase of the carrier. This is called *Quadrature Amplitude Modulation (QAM)*.



**4.3.1 8-QAM** An 8-QAM constellation is shown below (one of many possible 8-QAM constellations). The eight symbols along with the 3-bit digital words corresponding to each are shown to the right of the constellation. This system uses 2 possible amplitudes and 4 possible phases. In 8-QAM, the duration of a symbol is three times the duration of a bit (since each symbol carries 3 bits). Note that there are both phase and/or amplitude changes for each symbol. For the system with the constellation shown below, the eight output symbols might be  $2 \cos(2\pi f_c t \pm 45^\circ)$ ,  $2 \cos(2\pi f_c t \pm 135^\circ)$ ,  $4 \cos(2\pi f_c t \pm 45^\circ)$ , and  $4 \cos(2\pi f_c t \pm 135^\circ)$ .



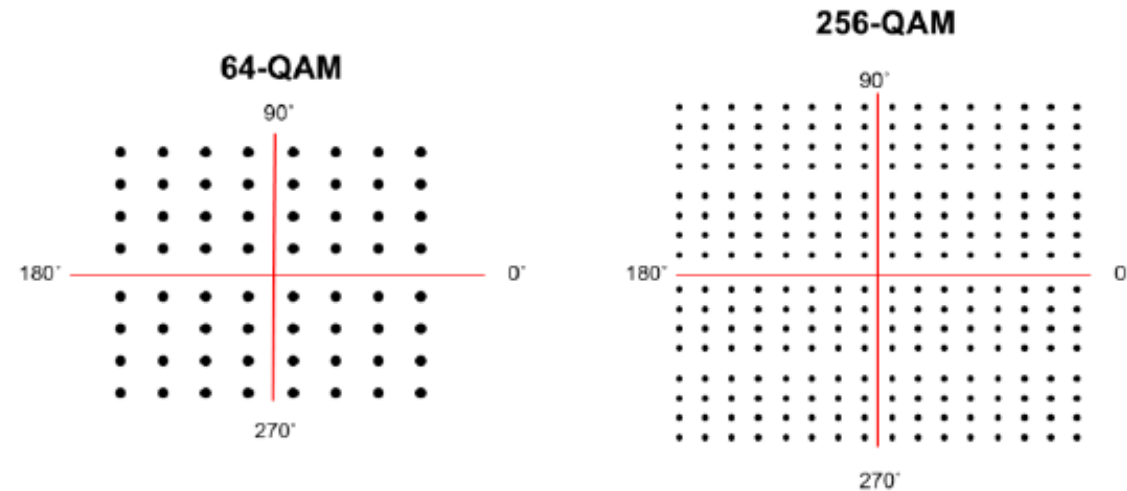
What is the bandwidth for 8-QAM? The same as for 8-PSK, since the bandwidth for all digital modulation types (except for FSK) is given by

$$BW = 2R_{sym} = \frac{2R_b}{N} = \frac{2R_b}{3}$$

And it doesn't stop there.



**4.3.2 Higher level QAM signals** QAM signaling can be extended to have a larger number of symbols, which then allows a much higher bit rate in the same bandwidth (because there are more bits per symbol). 64-QAM and 256-QAM are common in cable modems, satellites, and high-speed fixed broadband wireless. Some possible constellations are in the following figure.



In 256-QAM, you find that for each symbol you are transmitting (there are 256 symbols), there are 8 bits of information. Assuming the symbol rate remains constant, then for the same bandwidth you are sending 8 times more information when you use 256-QAM than when you use OOK, ASK or BPSK. For 256-QAM, if the bitrate is 600 kbps, the bandwidth is  $2(600,000)/8 = 150$  kHz.

Now that's powerful!

## 5. Summary of Types of Digital Modulation

The following table summarizes some of the various types of digital modulations introduced. With  $M$ -ary digital modulation, even more symbols are used (so there are more bits/symbol). This means there are PSK and QAM modulations with  $M=32$ , 64, 128, etc, although usually  $M=32$  is the maximum number of symbols used with PSK due to the noise effects.

<b>Modulation</b>	<b># Symbols (<math>M</math>)</b>	<b># Bits/symbol (<math>N=\log_2 M</math>)</b>	<b>Bandwidth (<math>BW</math>)</b>
<b>FSK</b>	2	1	$f_{mark} - f_{space} + 2R_b$
<b>ASK</b>	2	1	$2R_{sym} = 2R_b/N = 2R_b$
<b>OOK</b>	2	1	$2R_{sym} = 2R_b/N = 2R_b$
<b>BPSK</b>	2	1	$2R_{sym} = 2R_b/N = 2R_b$

<b>QPSK</b>	4	2	$2R_{sym} = 2R_b/N = R_b$
<b>8PSK</b>	8	3	$2R_{sym} = 2R_b/N = 2R_b/3$
<b>8QAM</b>	8	3	$2R_{sym} = 2R_b/N = 2R_b/3$
<b>16PSK</b>	16	4	$2R_{sym} = 2R_b/N = R_b/2$
<b>16QAM</b>	16	4	$2R_{sym} = 2R_b/N = R_b/2$
⋮	⋮	⋮	⋮

