OVERVIEW

Quadrature Amplitude Modulation, or QAM, is an everyday modulation scheme used in today’s telecommunications world. While most of us don’t see it, this modulation technique is used in DSL, Cable Modem, wireless, HDTV, fiber-optic communications and more. QAM is also used in Bluetooth applications—it’s everywhere and you may not even know it! This article will focus on QAM, or QPSK because once you understand this baseline, you’ll understand the principles behind high-order QAM such as 16, 64, and 256QAM and how amazing that technology really is!

UNDERSTANDING THE QUADRATURE CONCEPT

The first step to understanding QAM starts with the “Q” which is our ‘Quad’ and in the end this involves understanding waves. A sine wave is comprised of both a horizontal and vertical component that can be expressed in two common ways. The first way is to graph a sinusoidal wave using the “x” axis as your phase (in degrees from 0 to 360) and the “y” axis as the amplitude of that signal. The second method converts the 360 degree wave into a polar graph with QUADRANTS. This will be detailed towards the end of the article.

To the left is a familiar sine wave with no phase shift. The wave begins at an amplitude of zero and at zero degrees. It peaks at 90 degrees and returns to zero at 180 degrees. It then reaches the minimum 90 degrees later before returning where it started. Note that every wave’s period completes a 360 degree cycle—this will be key to understanding QAM a bit later in this article.

In QAM systems, the Sine, -Sine, Cosine, and –Cosine comprise the four phase shifts that occur within a QAM system and electronics use this phase shifting to understand what bits are being sent and received. In a QAM system, 2 bits are represented in each phase shift using two of these phase shifts at a given moment in time. Let’s take a step back and review bits; then we’ll assign a relationship between those bits and the phase shift associated with them.

In order to represent 4-decimal numbers we’ll need a 2-bit binary system. In this situation, the Most Significant Bit (MSB) will be a “2” or a “0” depending on whether it is on or off. The Least Significant Bit (LSB) is a “1” or a “0” depending on whether it is turned on or off. The combination of both can provide decimal numbers “0” through “3” for a total of “4”. Understanding MSB and LSB is important since these phase shifts are exactly how QAM modulation works to depict each set of bits.

DECIMAL TO 2-BIT BINARY CONVERSION

<table>
<thead>
<tr>
<th>Decimal</th>
<th>MSB</th>
<th>LSB</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Now that we have the basics of bits and waveforms down we need to take a look at the different waveforms... in reality, they are the same waveforms but all of them are phase-shifted. When two of these waveforms are combined, variances in phase and amplitude become a new waveform. These new waveforms are measured by the receiving device as a 00, 01, 10, or 11 as shown in the previous table. Below are the different waves and assignments to their MSB and LSB roles.

You can see that we’re only detailing two different types of waves in the sense that the Sine and –Sine waves are simply the opposite of each other. The same is true for the Cosine wave. Because Sine and - Sine are exactly opposite of each other, they can never be used together because they would cancel each other out when combined to represent the LSB. In the end it makes perfect sense that Sine waves, whether positive or negative, are assign to the LSB since you can’t have two LSBs at the same time... nor can you have two opposing MSBs at the same time because they would simply cancel each other out given that Cosine is assigned to the 0 MSB and –Cosine is assigned to the MSB of 1.

**CALCULATING THE BIT REPRESENTATION OF A WAVEFORM**

You can create the representative waveform for 2-bits by adding the two original waveforms that correspond to the MSB and LSB. The resulting waveform will have differences in phase and amplitude. The phase will indicate what sector, or angle, the wave is at and thus the corresponding pair of bits from one waveform. To better depict this, we’ll run through a few calculations so you can see how two waves look once they are added together.
ADDING OUR WAVES FOR THE RESULT

Below are calculations that show the result of the MSB wave and LSB wave summed together for each combination.

<table>
<thead>
<tr>
<th>Decimal (O), MSB (O) and LSB (O), Cosine + Sine</th>
<th>Decimal (1), MSB (O) and LSB (1), Cosine + (-)Sine</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Cosine = 0 MSB" /></td>
<td><img src="image" alt="Cosine = 0 MSB" /></td>
</tr>
<tr>
<td><img src="image" alt="Sine = 0 LSB" /></td>
<td><img src="image" alt="Sine = 1 LSB" /></td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td><strong>Equals</strong></td>
</tr>
<tr>
<td><img src="image" alt="Cosine + Sine (MSB O and LSB O) = Decimal O" /></td>
<td><img src="image" alt="Cosine + (-)Sine (MSB O and LSB 1) = Decimal 1" /></td>
</tr>
</tbody>
</table>
Decimal (2), MSB (1) and LSB (0), (-)Cosine + Sine

- Cosine = 1 MSB

+ 

Sine = 0 LSB

Equals

- Cosine + Sine (MSB 1 and LSB 0) =

Decimal 2


Decimal (3), MSB (1) and LSB (1), (-)Cosine + (-)Sine

- Cosine = 1 MSB

+ 

- Sine = 1 LSB

Equals

- Cosine + (-)Sine (MSB 1 and LSB 1) =

Decimal 3
WAVEFORM RESULTS

If you look at each resultant waveform you will see similarities between them. For example, adding Cosine and Sine results in the inverse of adding –Cosine and –Sine. Makes perfect sense. Same goes for adding Cosine and –Sine compared to –Cosine and Sine. The real concept is that 2-bits are represented in one composite waveform...or more accurately a phase shift to represent the data.

If you look at the calculations you can confirm the accuracy of adding these waveforms. By looking at the first calculation for Decimal number (0) the amplitude of Cosine at 0 degrees to be 1. Simultaneously, the value for Sine is 0. The summation of the combined waveform is 1. You can add the Cosine and Sine to calculate the value of the amplitude for any phase-point to calculate the resultant waveform which ends up in a different phase than either of the two original waveforms.

LOOKING AT THIS THROUGH A POLAR LENSE

Another way to evaluate waveforms is by use of polar graphs. Because sinusoidal waveforms are comprised of 360 degrees they can be easily analyzed using methods such as polar graphs. For QAM it improves understanding greatly because you can depict the value of phase, amplitude, and data all at the same time. The image below is probably familiar to you from a past math class.

In this example, there is a given amplitude to a point that has a phase of 45 degrees and represents an MSB of 0 and a LSB of 0. There are four QUADRANTS that consist of phase, amplitude, and ultimately the representation of the MSB and LSB. In QAM the value of phase and amplitude are measured to determine the bit value. In the example shown here, each 90 degree phase represents one of the 2-bit binary values. In 16, 64, 256QAM systems the amplitude becomes an important part of representing bit values.

EXTENDING BEYOND QAM/QPSK

So now we’ve explored the basics of QAM – what’s next? Well the short answer is a lot. This article provided an insight into what QAM is - 4QAM/QPSK. This technology has already been further advanced to 16QAM, 64QAM, and 256QAM. We’ll overview that in the next section but it would require a large article to truly embrace the technology it takes to understand the workings of higher-order QAM modulation techniques. Below is a depiction of what higher-order QAM technologies look like using polar graphs – and given the knowledge you now have it’s easy to appreciate what the latest technology has to offer...it is very impressive when you understand what is happening!

<table>
<thead>
<tr>
<th>10</th>
<th>00</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>01</td>
</tr>
</tbody>
</table>

This is a polar graph of what we did for this article. It uses 2-bits per phase and amplitude to represent decimal numbers up to four.
This is a polar graph for a QAM system capable of sending a decimal numbers up to 16 using 4-bit representation for the phase and amplitude of a wave. 16QAM uses two more bits than QAM resulting in 16 decimal digits and “squares” since each number needs a unique location in phase and amplitude.

The polar graph for 64QAM. This modulation technique uses two more bits than 16QAM, resulting in four times as many combinations – 64.

The most recent is 256QAM – a very high performance modulation scheme. Consideration must be made to signal strength and quality such as signal to noise ratios. Higher order QAM modulation schemes must be able to accurately receive the phase AND amplitude with precision; noise or poor signal quality can cause significant problems.

Most devices experiencing problems using a higher-order QAM (and other modulation techniques) will commonly “downshift” to the next lower technique until a reliable connection is established. For example, the device would attempt a connection at 256QAM but if bit errors become problematic it would seek to change the connection to 64QAM.

**FINAL THOUGHTS…**

So this was a pretty deep article but something interesting, as a concept, to learn. There are other QAM techniques I didn’t mention such as 8QAM and 32QAM which have the same theory of application in communication system pertaining to the use of phase angle and amplitude to represent a set of bits.

The next time you are reviewing a product you might see a listing for the modulation type and know a little more than bigger numbers are better – you’ll know the concept behind why!