Motivation

Noise caused by the channel in digital communication can be catastrophic to a signal. In particular, noise can destroy an image and make it indistinguishable. As a result, some endeavor to reduce noise is necessary when transmitting digital images.

We investigated the effects of upsampling with and without filtering as well as the effects of using different filters on the noise level of an image after being transmitted through a simulated channel.

The problem we will tackle in regard to this is the transmission of a gray-scale image. This image will be filtered before and after subjection to a noisy channel and the result will be analyzed.

Quadrature Amplitude Modulation (QAM)

Quadrature Amplitude Modulation (QAM)

All signal communications must adhere to frequency restrictions so that they can be received without interference. This gives rise to the notion of carrier modulation, where a baseband signal is moved to an unoccupied section of the frequency domain before transmission. This is also known as frequency modulation. This also simultaneously addresses the issue that low-frequency signals suffer greatly from attenuation during transmission through a medium.

In order to transmit digital information, symbols are needed to represent the bits. The simplest set is known as BPSK, which consists of just two symbols; one represents 0, while the other represents 1. The baud rate in this case is only one; more complicated methods are necessary if we wish to improve upon this. While there are many types of modulation of varying complexity, we will focus on one of the popular methods known as 16-Quadrature Amplitude Modulation (16-QAM).

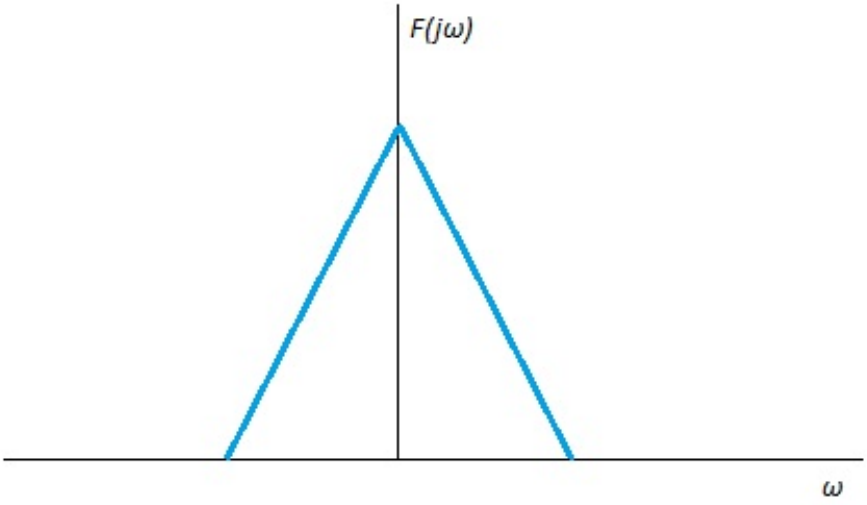
16-QAM utilizes both amplitude and phase alterations in conjunction with frequency modulation in a way that allows each symbol to represent four bits rather than just one. This increase in baud rate comes at the cost of design complexity and cost. The transmitter must send two signals simultaneously; in order to do this in a way that the signals can be separated by the receiver, the two signals must be orthogonal to each other. This is implemented via the frequency modulation, except one signal is modulated by a cosine and the other a sine. Thus, the output s(t) can be defined as
Equation:

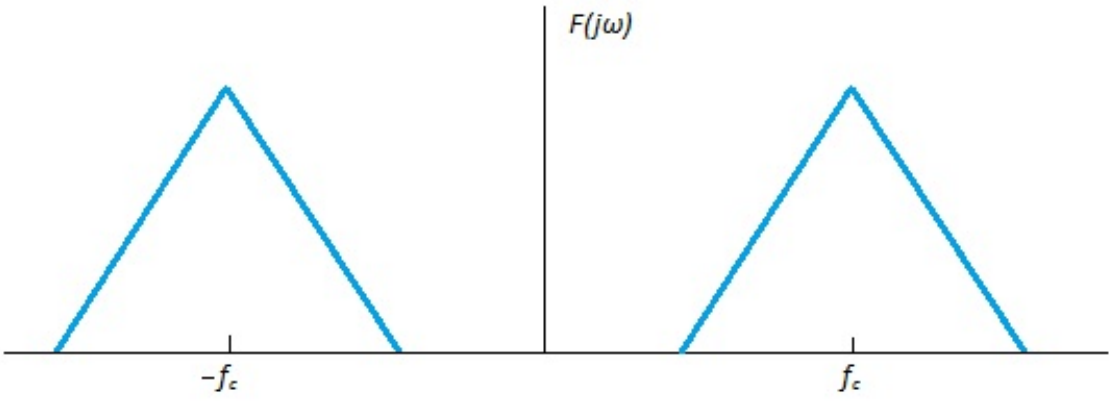
$$s(t) = I(t)\cos(2\pi \mathrm{f}_c t) - Q(t)\sin(2\pi \mathrm{f}_c t)$$

The first signal is known as the in-phase component, while the other is known as the quadrature component. The fact that

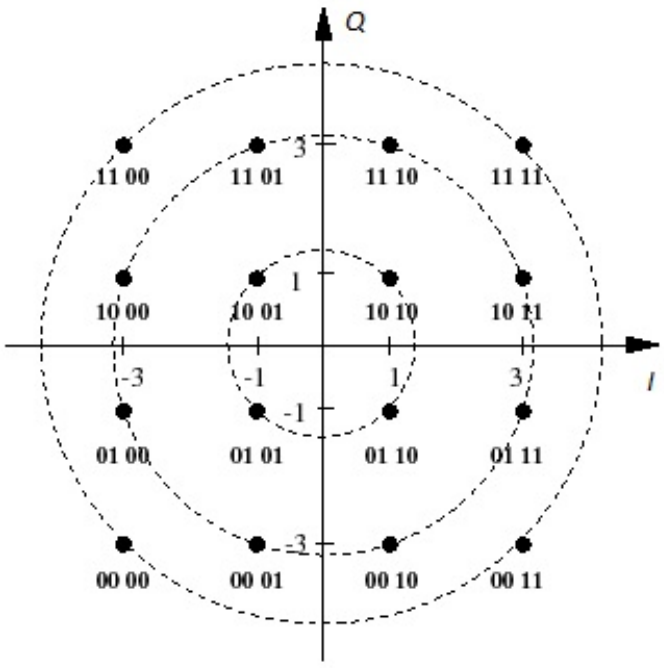
$$\cos(\omega t)\cdot\sin(\omega t) = \int_0^{2\pi k} \cos(\omega t)\sin(\omega t) * dt = 0, k \in Z$$

implies the signals' orthogonality. Multiplication by these sinusoids, via properties of the Fourier Transform, centers the frequency representation of the signal around plus and minus f_c rather than at baseband.





The various amplitudes paired with the two phases can be succinctly represented by a constellation map as shown below.



Each point corresponds to a particular pair of amplitudes of the two signals. To combat the effects of noise, the points of the constellation are placed as far away from each other as possible so avoid misinterpretation. Many constellation configurations can be used; ours is described below:

Bits	**I(t)**	**Q(t)**
0001	1	1
0010	3	1

0011	1	3
0100	3	3
0101	1	-1
0110	1	-3
0111	3	-1
1000	3	-3
1001	-1	1
1010	-1	3
1011	-3	1
1101	-3	3
1110	-1	-1
1111	-3	-1
1110	-1	-3
1111	-3	-3

In order to correctly interpret the data from r(t), the received signal s(t) with the addition of white noise after it passes through the channel, the receiver must recover I(t) and Q(t). I(t) is obtained by modulating s(t) by a cosine of identical frequency and phase as the original modulation, while Q(t) is obtained in the same way but with a sine instead. A low-pass filter will then yield the original signal, as the following equations illustrate:
Equation:

$$r(t) = I(t)\cos(\omega t) + Q(t)\sin(\omega t)$$

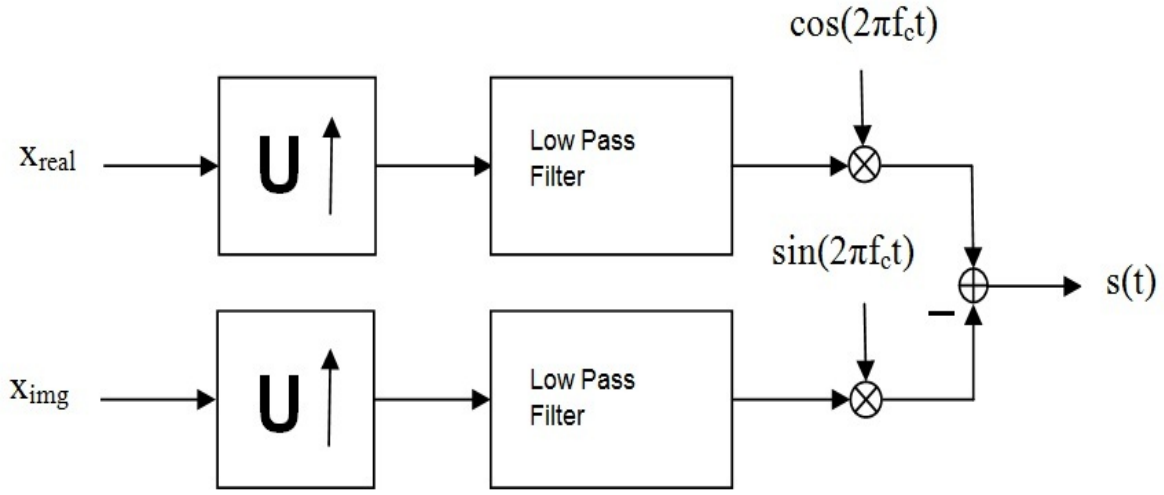
Equation:

$$I_{rcvd}(t) = LPF[r(t)\cos(\omega t)]$$
$$I_{rcvd}(t) = LPF[I(t)\cos^2(\omega t) + Q(t)\sin(\omega t)\cos(\omega t)]$$
$$I_{rcvd}(t) = LPF[\tfrac{1}{2}I(t)(1 + \cos(2\omega \mathrm{t})) + \tfrac{1}{2}Q(t)\sin(2\omega \mathrm{t})]$$
$$I_{rcvd}(t) = \tfrac{I(t)}{2}$$

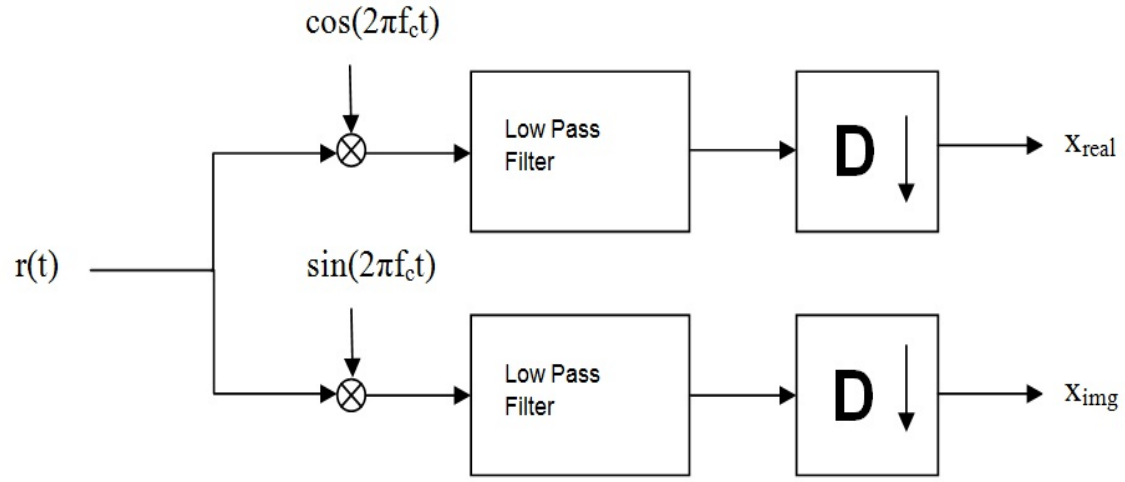
The low-pass filter removes the components of frequency 2ω, leaving only a baseband signal. A similar approach shows that indeed
Equation:

$$Q_{rcvd}(t) = LPF[r(t)\sin(\omega t)] = \frac{Q(t)}{2}$$

Thus, both signals I(t) and Q(t) can successfully be recovered at the receiver. Below is the block diagram implementation of a transmitter using 16 QAM:



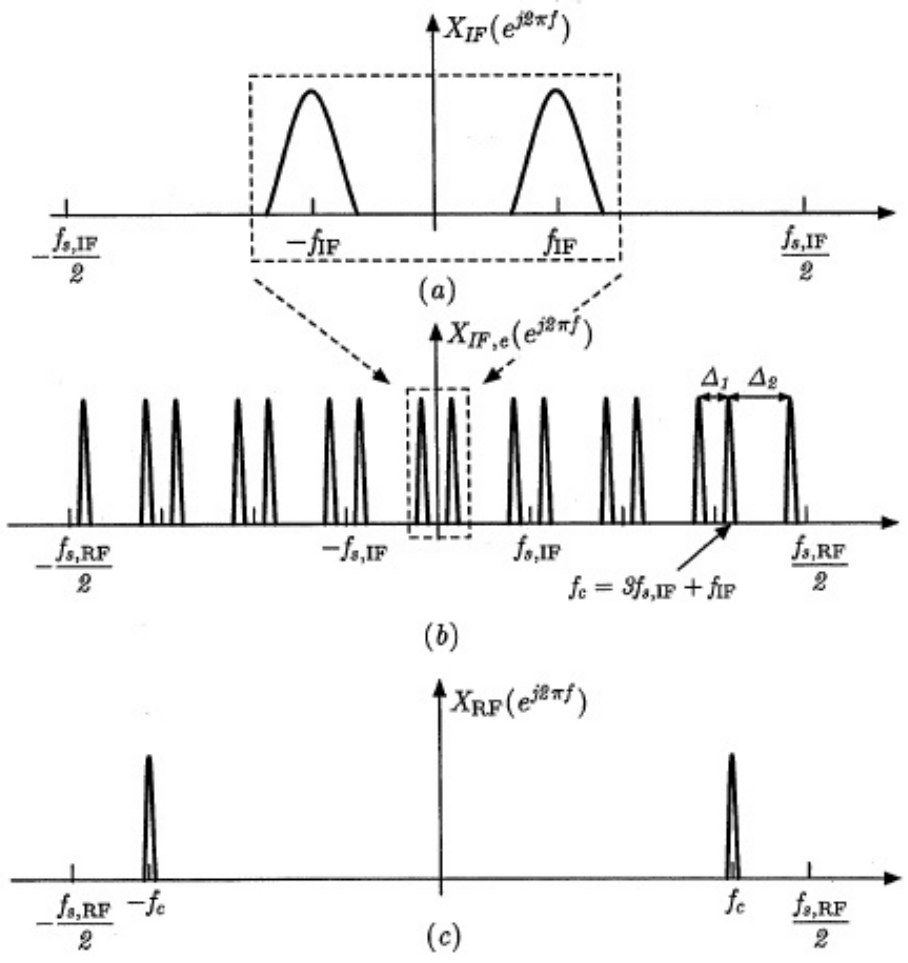
Below is the block diagram implementation of a receiver using 16 QAM:



Digital Filtering

Purpose of Upsampling in Digital Filters in Communication

In digital communications filters are important in the process of upsampling. Upsampling is required for transmission because we want the signal's frequency representation to be narrow and confined to frequencies around the carrier frequency. By upsampling the signal, the frequency response of the signal to be transmitted gets compressed and becomes band limited to a significantly smaller range of frequencies, which is necessary for transmission.



As can be seen from the figure above, (a) shows the frequency domain representation of the signal to be transmitted, (b) shows the upsampled version of the frequency response, and upon low pass filtering the upsampled signal we get only two spikes. When the signal is modulated to the carrier frequency, both spikes appear at the corresponding carrier frequency in (c). This is important because we don't want information to be spread across the frequency spectrum, rather we want to transmit the signal at a specific carrier frequency.

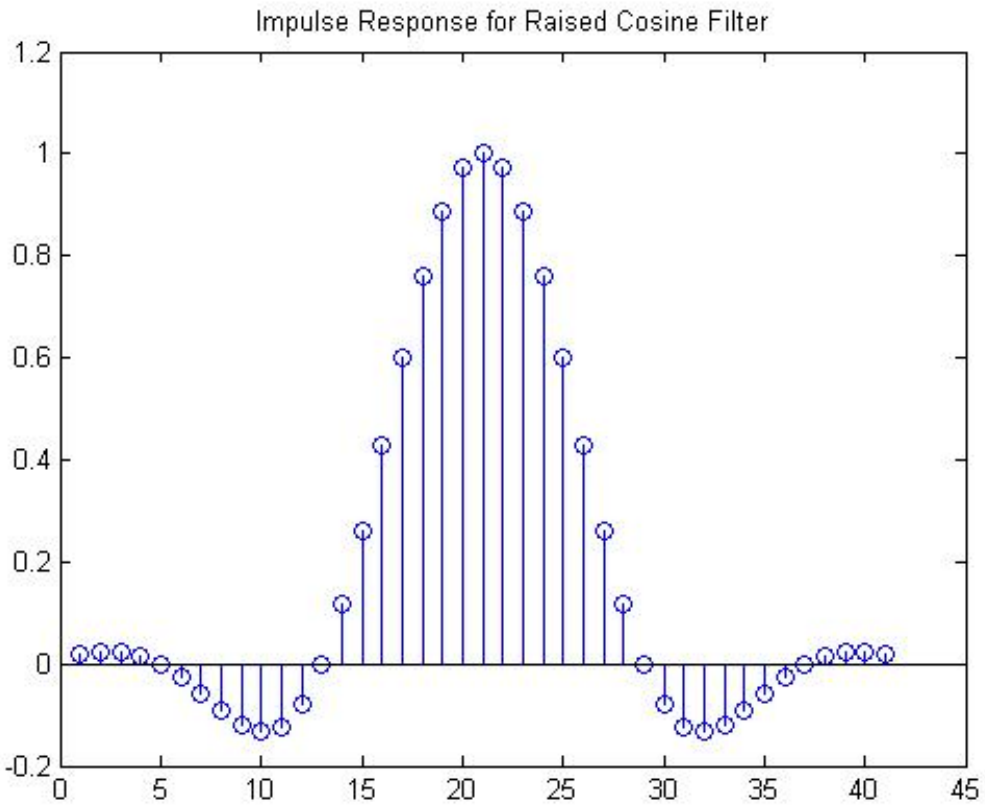
The upsampling process is accomplished by taking a signal, inserting L zeros between each sample and then low pass filtering the result. Below is a description of possible low pass filters that can be used to achieve this result.

Raised Cosine Filter

The raised cosine filter is a type of low pass filter that accomplishes the interpolation necessary after inserting the L zeros between each sample. It's frequency response is given by:

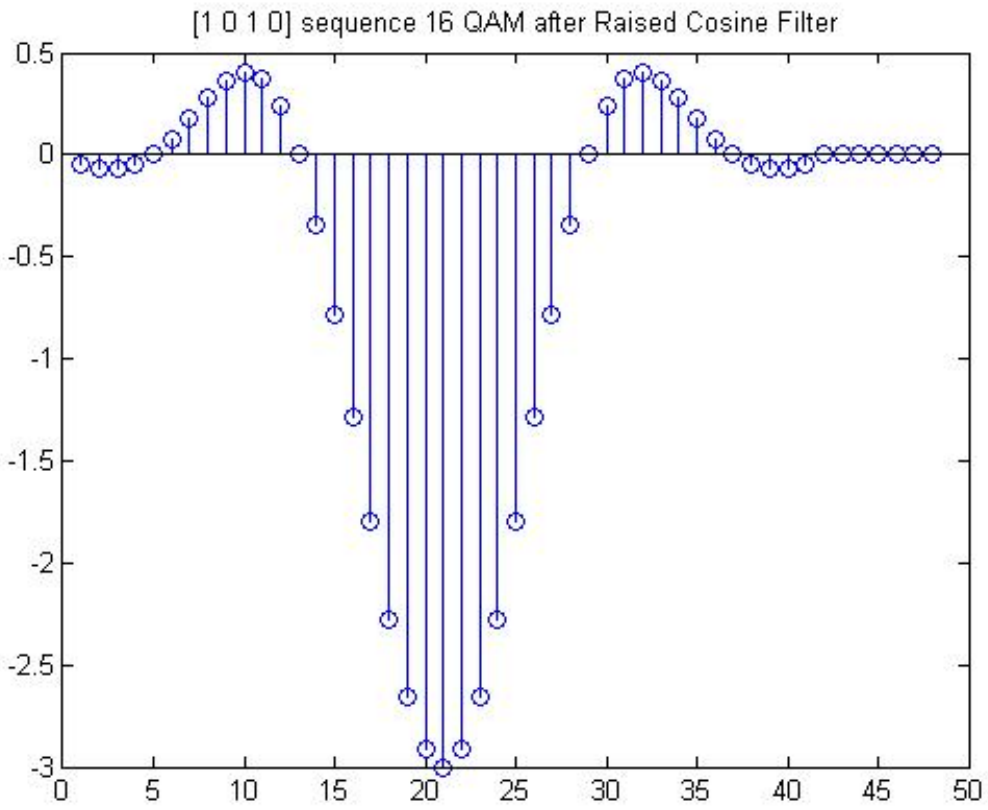
$$\mathrm{P}_{rc}(f) = \begin{cases} T_b & \text{for } 0 \leq |f| \leq \dfrac{1-\alpha}{2T_b} \\ \frac{T_b}{2}\left\{1 + cos\left[\frac{T_b\pi}{\alpha}\left(|f| - \frac{1-\alpha}{2T_b}\right)\right]\right\} & \text{for } \dfrac{1-\alpha}{2T_b} \leq |f| \leq \frac{1+\alpha}{2T_b} \\ 0 & \text{otherwise} \end{cases}$$

In the filter α is a parameter which is between 0 and 1 and is called the rolloff factor. The larger α is, the wider the bandwidth of the filter. As α approaches zero the filter will become a brick wall and will look like a box in the frequency domain. The Impulse response of the filter is shown in the figure below:

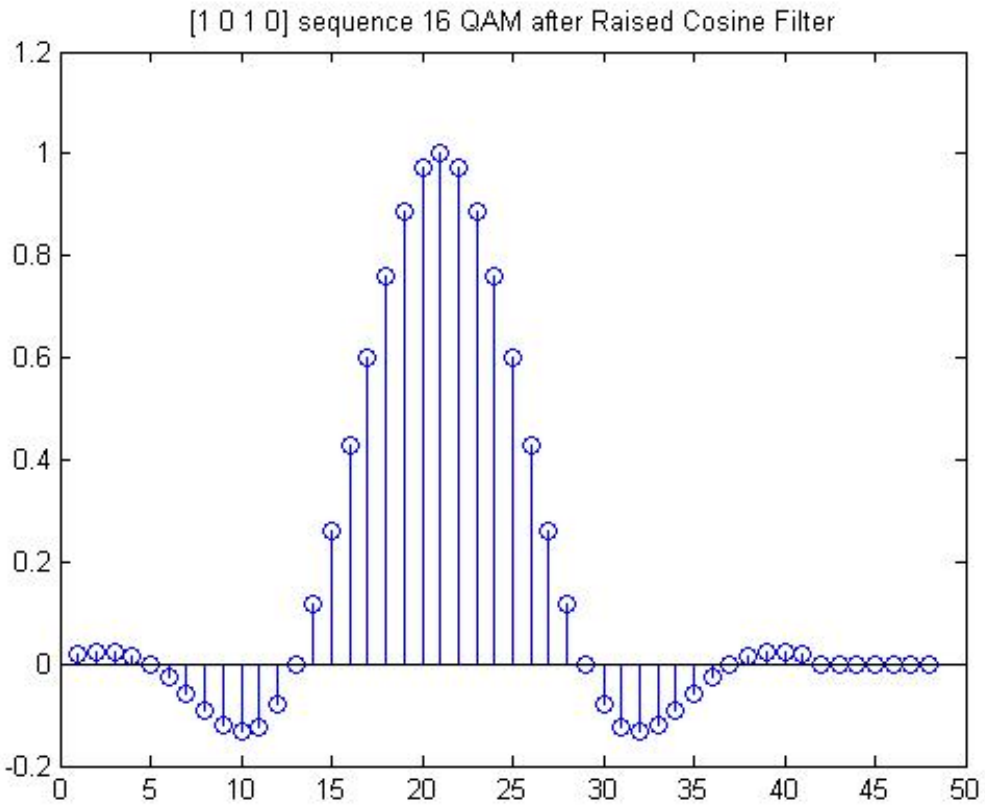


As can be seen above, the time domain representation is given by a sinc and so in reality the raised cosine filter would extend to plus and minus infinity. However, above the impulse response was truncated and selected to have a length of 41 for the purposes of our digital communication scheme. The raised cosine filter is the best filter for digital communication, specifically 16 QAM, because it removes interference that may occur from one symbol to the next. This means that the waveform can be recovered perfectly at the receiver and this is why the raised cosine filter is typically used in digital communication.

In order to complete the upsampling process it is necessary to convolve the impulse response of the raised cosine filter and the vector that contains the signal with zeros inserted between the samples. The output of this convolution will be the upsampled signal. Below is the output of the real part of a [1 0 1 0] sequence convolved with the raised cosine filter.



As can be seen above, the absolute value of the maximum value is -3. This is because the first two bits, [1 0] correspond to the real part of the sequence and they get mapped to a value of I=-3. Below is the imaginary part of the signal, and as we can see the absolute value of the maximum value is 1. This is because the last to bits, [1 0] correspond to the imaginary part of the sequence and they get mapped to a value of Q=1.



The raised cosine filter is used because it limits the bandwidth of the signal and decays quickly in the time domain. The advantages of this is that it allows for data transmission in specific frequency ranges with an insignificant amount of information spread out across large frequencies.

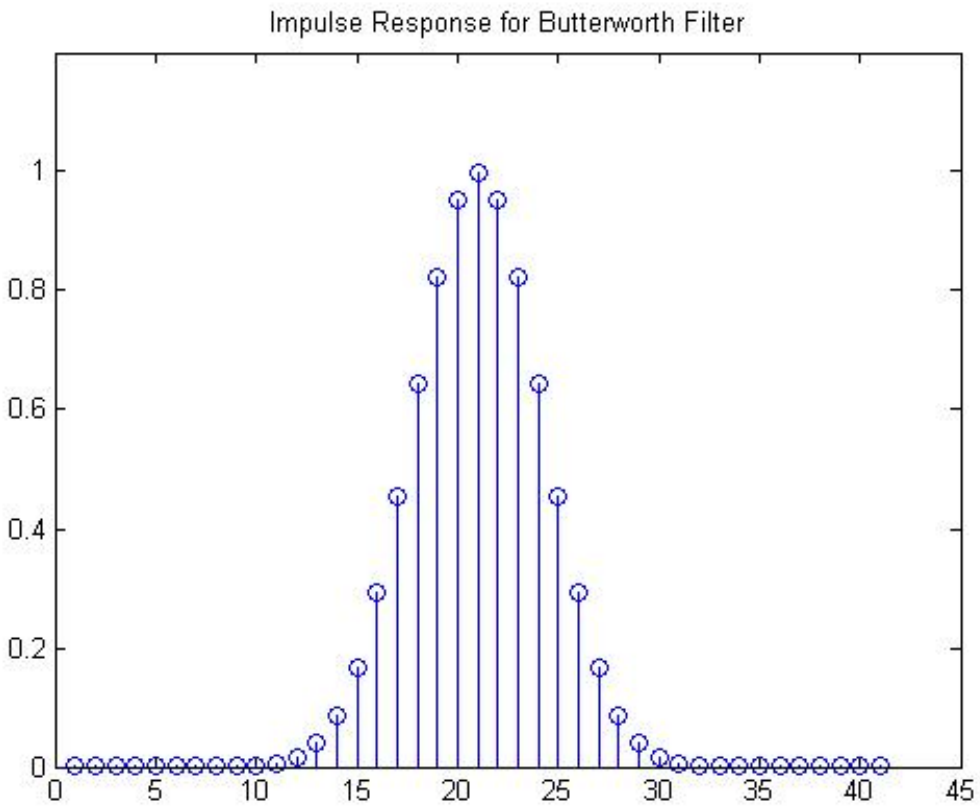
Butterworth Filter

Butterworth filters are another type of low pass filter which can be used to complete the upsampling process. The frequency response of the Butterworth filter is given by:

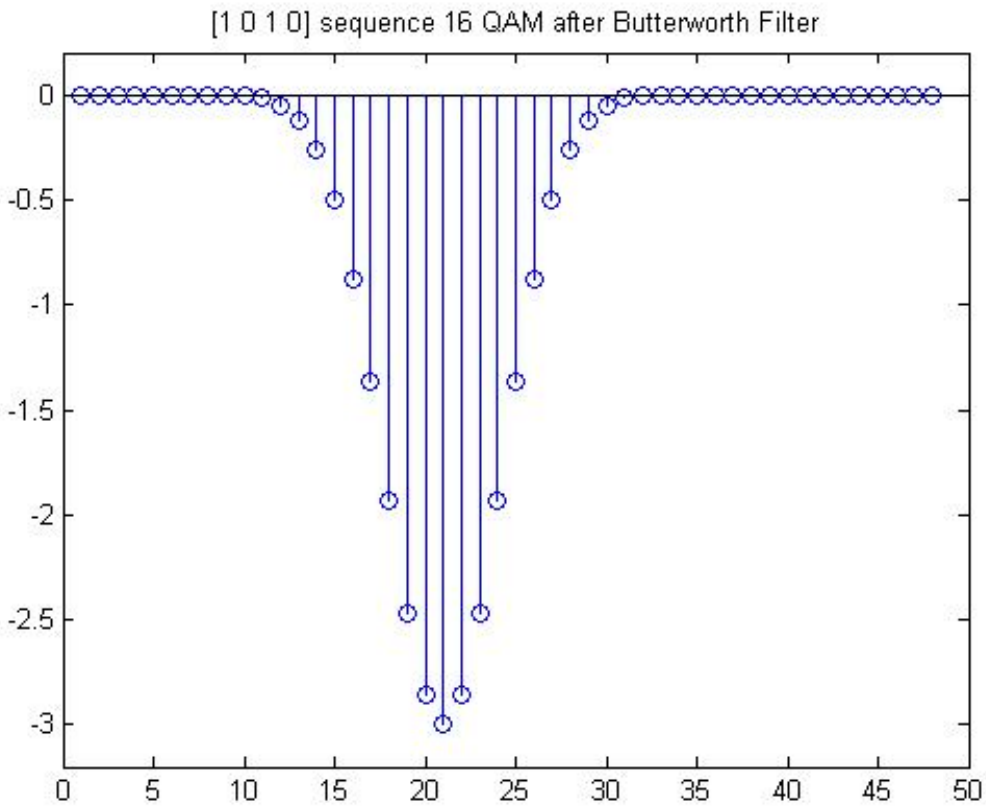
Equation:

$$H(j\omega) = \frac{1}{\sqrt{1+(\frac{\omega}{\omega_c})^{2N}}}$$

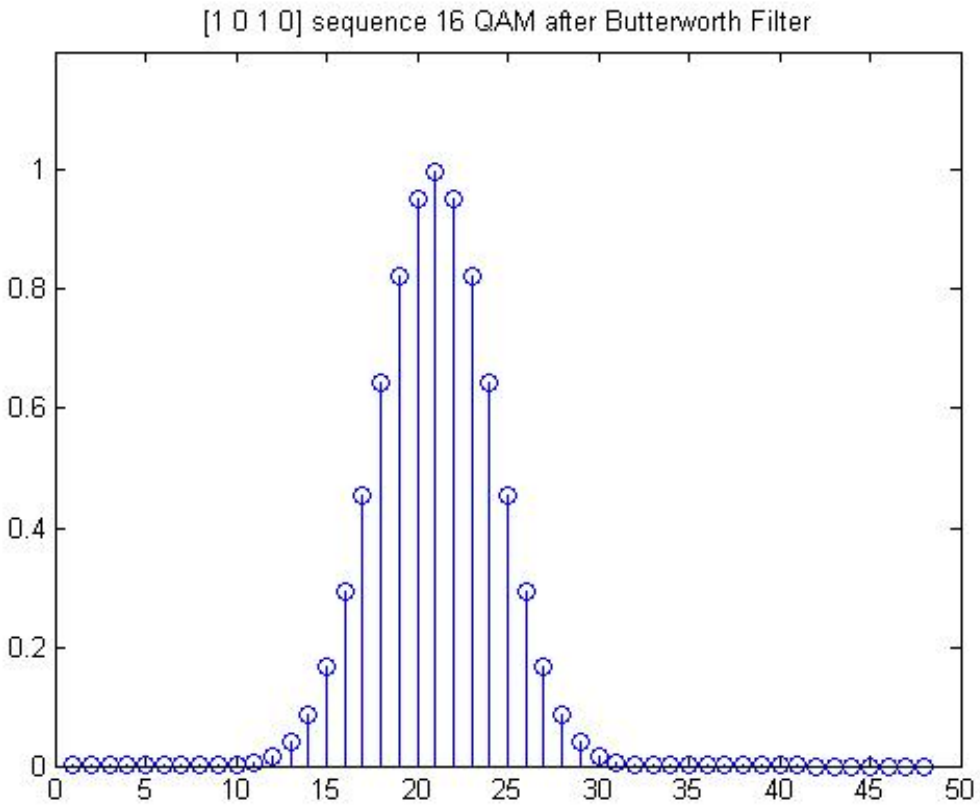
Where N is a parameter which is called the order of the filter and ω_c is the cutoff frequency. The Butterworth filter acts like a low pass filter because it has a flat frequency response that is usually unity gain in the passband and gradually rolls off to zero in the stopband. In between the passband and the stop band we have the cutoff frequency which will occur at the point where the gain is equal to 0.707 (1/sqrt(2)). Butterworth filters have a relatively slow roll off, especially when compared to the raised cosine filter. Below is the impulse response of the Butterworth filter:



Given a sequence of [1 0 1 0] the output of the low pass filtering of the real part of the signal will be:



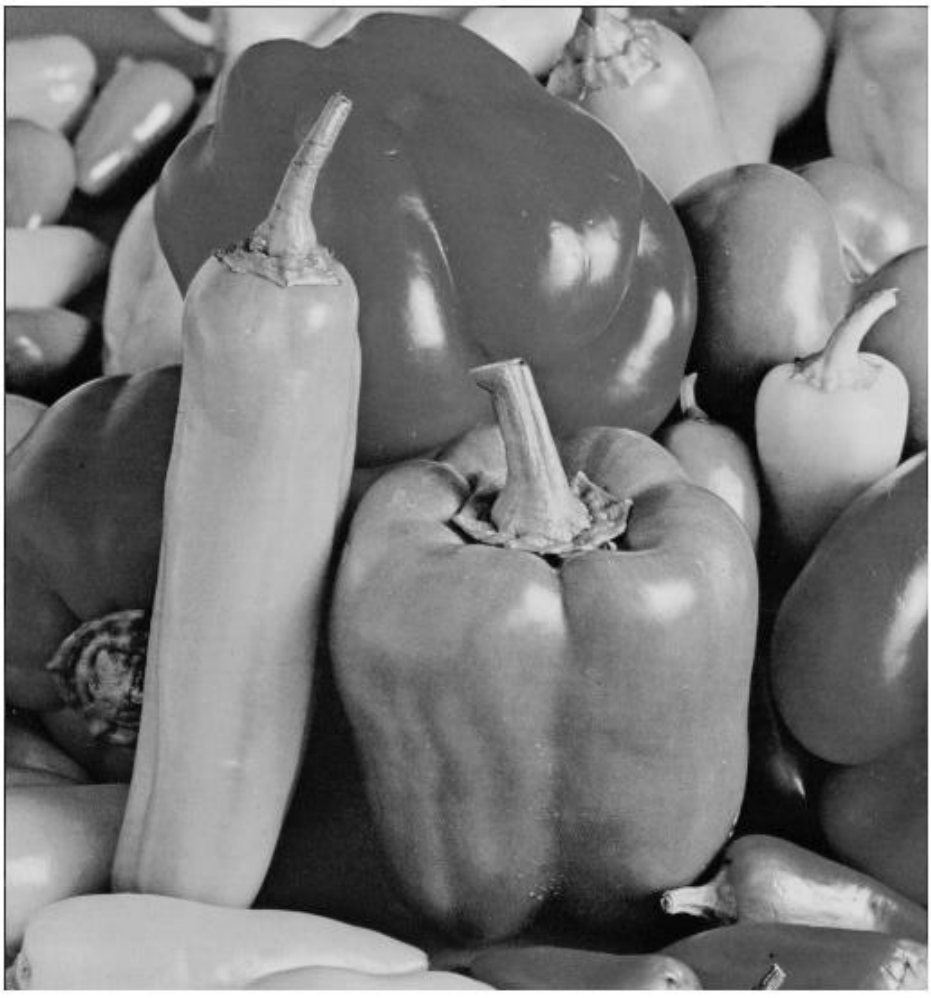
The output of low pass filtering the imaginary part of the signal with a Butterworth filter will be:



As can be seen the output of the Butterworth filter is similar to that of the raised cosine filter, where both map the real part of the sequence to -3 and the imaginary part of the sequence to 1. The main difference is that the raised cosine filter's output has ripples extending on both sides, while the Butterworth filter does not. This means that the Butterworth filter's output does not need to be truncated since it does not extend to infinity like the output of the raised cosine filter does. The output of these filters would then be modulated appropriately as described, summed together and then transmitted through the channel as described in the QAM module.

Results

Our goal in digital communications was to transmit a grayscale image shown below:

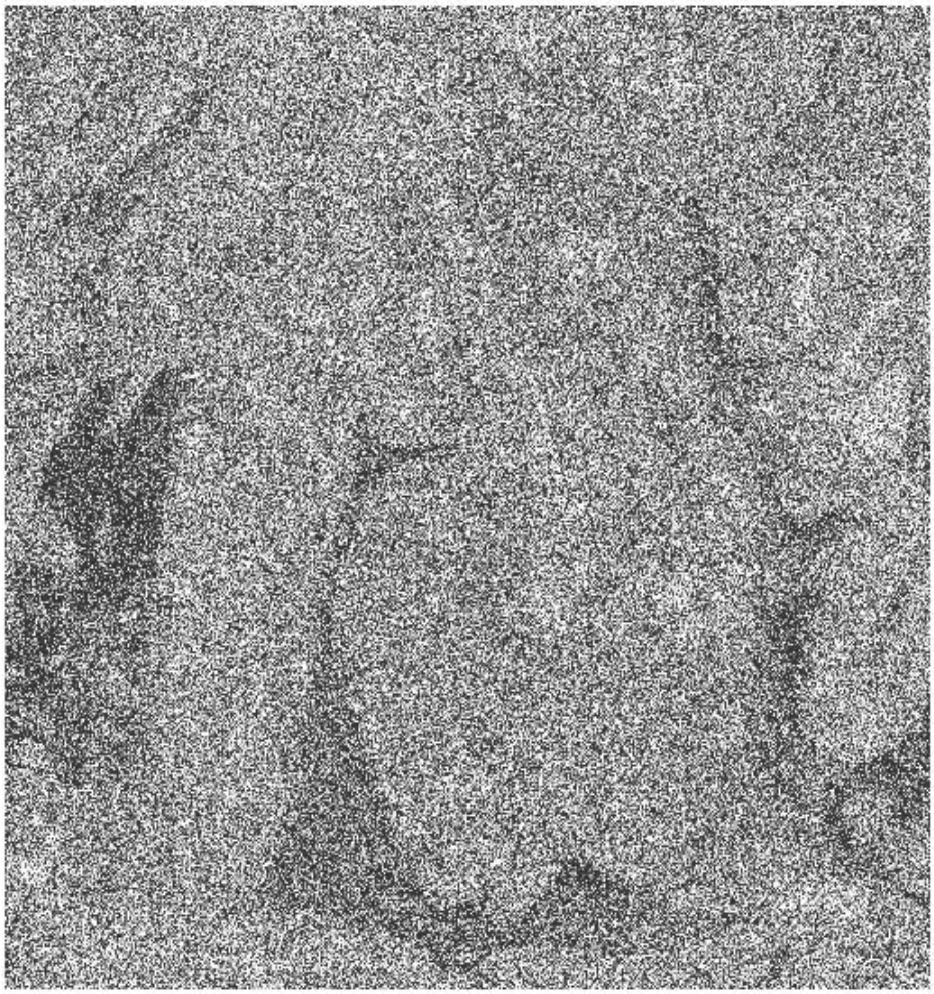


This image was represented as a matrix with dimensions 512x512 with each entry having values from 0 to 255. The values from 0 to 255 in each entry

of the matrix were taken, converted into a string of 0s and 1s and fed in four bits at a time into the 16 QAM modulator. This was then passed through a noisy channel which added an error of random numbers between -2.5 to 2.5 to the transmitted signal. The receiver collected the received signal, demodulated it and reconstructed the received image with noise. This was done with a raised cosine filter, a Butterworth filter and with no filter at all. We also compared the bit error with the original image for the raised cosine filter and the Butterworth filter. Below is the image received with noise using a raised cosine filter in the modulation phase:

Using the Butterworth filter in the modulation phase the received image with noise was:

Using no filter at all during the modulation scheme the received image was:



Error Calculations and Filter Evaluations

The bit error was calculated by analyzing the bit-representation of the original grayscale image and comparing it the bit-representation of the received image. The percentage of incorrect bits was calculated to be 34.55% when using the raised cosine filter, and 34.45% when using the Butterworth filter. From these results we can see that the performance of the Butterworth filter and the raised cosine filter are about the same, with the Butterworth filter performing slightly better. The bit error in the received image using no filter was 47%, however, percent error for the number of incorrect grayscale pixels was calculated to be 99%. The percent error for

the number of incorrect grayscale pixels using the Butterworth filter was 95.64%. The percent error for the number of incorrect grayscale pixels using the raised cosine filter was 95.66%. This is high because this includes even the slightest error which might be indistinguishable to the human eye.

We can also see the importance of using filters in the upsampling stage of the modulation phase of the digital communication scheme by looking at the received image without the use of a filter. The use of a digital filter allows the image to be recovered with far greater accuracy.

Code for Digital Communication in MATLAB

Attached are the functions which were written to simulate 16 QAM, construct the filters and transmit the image. All cases require the decimal to binary converters and binary to decimal converters and noise (these function are called in the other examples given below):d2b, b2d, channela. For the Raised Cosine Filter use:constructnew, demodcn, imagegray, imagebw. For the Butterworth filter use:constructnewb, demodcn, imagegrayb, For no filter use: constructnewn, demodcn, imagegrayn.

d2b

b2d

modulator with noise rcf

modulator with noise bwf

modulator with noise nf

demodulator

image simulator rcf

image simulator bwf

image simulator nf

image simulator rcf bw

channel noise

Future Work

Our work utilized, but did not rigorously analyze, multiple digital filters. Future students could build upon our work by computing SNRs of more types of filters, mathematically determining which provide the least errors.

Not all noise is the same; there are many models of channel noise. Future work could investigate how each of the proposed digital filters handle different types of noise, and conclude which filter behaves best for a given channel type.

Our project does not attempt to eliminate the noise once it has been introduced. Error-correcting codes could be implemented to further reduce the effects of noise on the transmitted signal.