



Intuitive Guide to Principles of Communications
www.complextoreal.com

Inter Symbol Interference (ISI) and raised cosine filtering

Inter-symbol interference (ISI) is an unavoidable consequence of both wired and wireless communication systems. Morse first noticed it on the transatlantic telegraph cables transmitting messages using dots and dashes and it has not gone away since. He handled it by just slowing down the transmission.

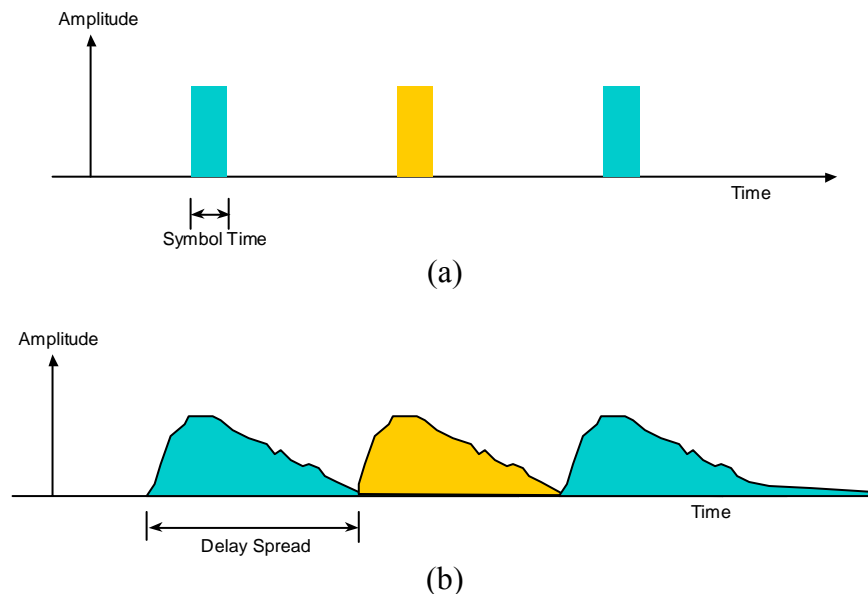


Fig 1 - (a) What was sent, (b) What was received.

During these early attempts at transmission, it was noticed that the received signals tended to get elongated and smeared into each other. A short pulse to represent a dot was received as a much-smearred version of the same thing. The problem appeared to be related to the properties of the medium used and the distance of signal travel. To counter this undesired effect, intermediate repeating stations were established and ways had to be devised to reduce this smearing.

How does this actually affect us now? Fig 2 shows a data sequence, 1,0,1,1,0, which we wish to send. This sequence is in form of square pulses. Square pulses are nice as an abstraction but in practice they are hard to create and also require far too much bandwidth. So we shape them as shown in the dotted line. The shaped version looks essentially like a square pulse and we can quickly tell what was sent even visually. Advantage of (an arbitrary) shaping at this point is that it reduces bandwidth requirements and can actually be created by the hardware.

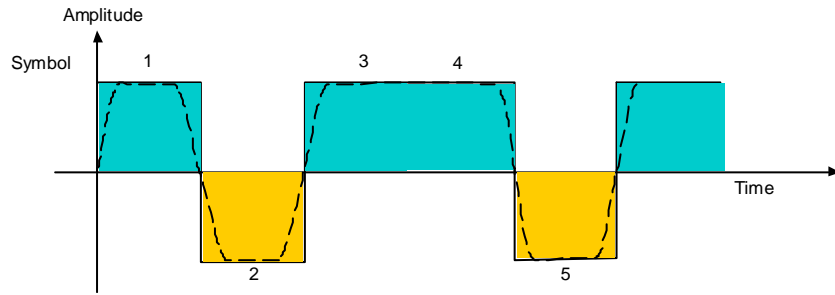


Fig 2 - Sequence 101101 to be sent, the dashed line is the shape that is actually sent.

Fig 3 shows each symbol as it is received. We can see what the transmission medium creates a tail of energy that lasts much longer than intended. The energy from symbols 1 and 2 goes all the way into symbol 3. Each symbol interferes with one or more of the subsequent symbols. The circled areas show areas of large interference.

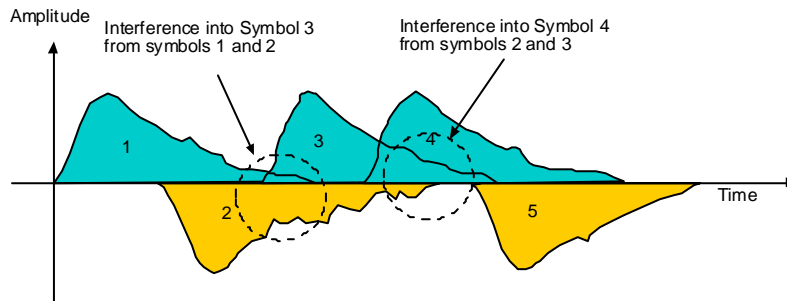


Fig 3 - Each symbol is spread by the medium

Fig 4 shows the actual signal seen by the receiver. It is the sum of all these distorted symbols. Compared to the dashed line that was the transmitted signal, the received signal looks quite indistinct. The receiver does not actually see this signal, it sees only the little dots, the value of the amplitude at the timing instant. Notice that for symbol 3, this value is approximately half of the transmitted value, which makes this particular symbol more susceptible to noise and incorrect interpretation and this phenomena is the result of this symbol delay and smearing.

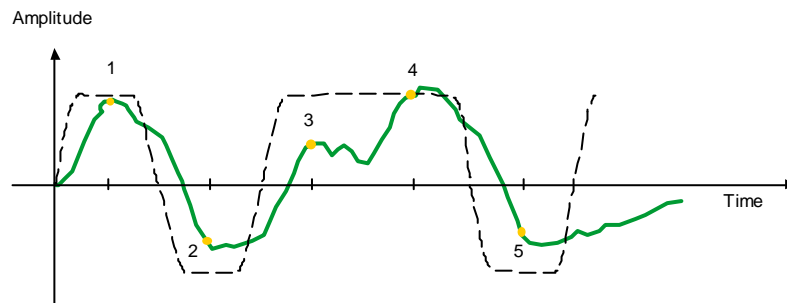


Fig 4 - Received signal vs. the transmitted signal

This spreading and smearing of symbols such that the energy from one symbol effects the next ones in such a way that the received signal has a higher probability of being interpreted incorrectly is called Inter Symbol Interference or ISI.

ISI can be caused by many different reasons. It can be caused by filtering effects from hardware or frequency selective fading, from non-linearities and from charging effects. Very few systems are immune from it and it is nearly always present in wireless communications.

Communication system designs for both wired and wireless nearly always need to incorporate some way of controlling it.

What can you do about ISI?

The main problem is that energy, which we wish to confine to one symbol, leaks into others. So one of the simplest things we can do to reduce ISI is to just slow down the signal. Transmit the next pulse of information only after allowing the received signal has damped down, as shown below. The time it takes for the signal to die down is called delay spread, whereas the original time of the pulse is called the symbol time. If delay spread (see Fig. 1) is less than or equal to the symbol time then no ISI will result, otherwise yes.

Slowing down the bit rate was the main way ISI was controlled on those initial transmission lines. Then faster chips came and allowed us to do signal processing controlling ISI and transmission speeds increased accordingly.

In our march for ever higher bit rates, slowing down the data rate is an easy but an unacceptable solution. What else can we do to counter ISI that does not require reducing the bit rate?

The main tool used to counter ISI is pulse shaping. How can pulse shaping help control ISI? The secret lies in the digital demodulation process used. When the timing pulse slices the signal to determine the value of the signal at that instant, it does not care what the signal looked like before or after it. So if there was some way we could keep the symbols from interfering in such a way that they do not affect the amplitude at the slicing instant, we can counter ISI successfully.

Look at the wildly bouncing signal in Fig. 5. However, the receiver only sees the points at the timing pulses (shown below the signal) and rest of the variation has no effect. So as long as at these points, we can reduce the effect of adjacent symbols, that's all we need to do to mitigate the effect.

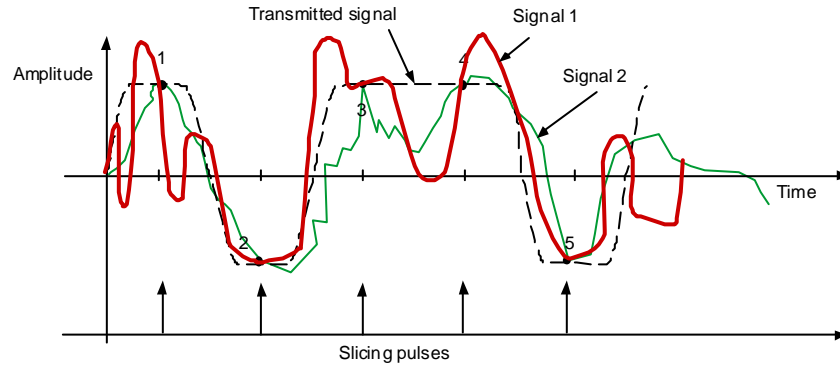


Fig 5 - We only care about what the signal does at the moment of sampling. What it does in between is not important.

Using a square pulse shape

We will start by looking at the use of a square pulse. It is an intuitive shape and we want to see what if anything is wrong with using it.

Let's define some terms

T_s = Symbol Time, 1 second in the example below. (Don't be confused by the negative time. The symbol time spans both positive and negative axis.)

R_s , the symbol rate is inverse of Symbol Time, T_s . R_s is directly related to bandwidth such that larger the symbol rate, the more bandwidth is required.

$$R_s = 1/T_s$$

Fig. 6 shows a square pulse of amplitude A, lasting 1 second. (From -0.5 secs to +0.5 secs)

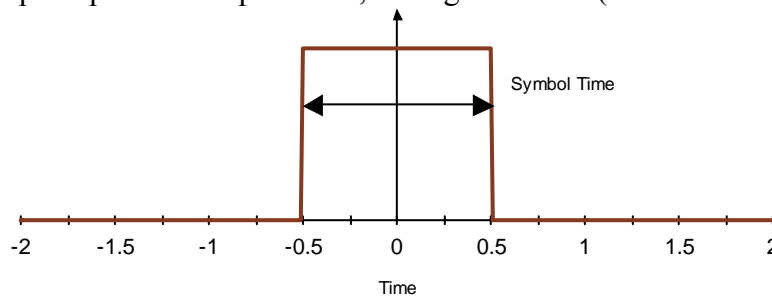
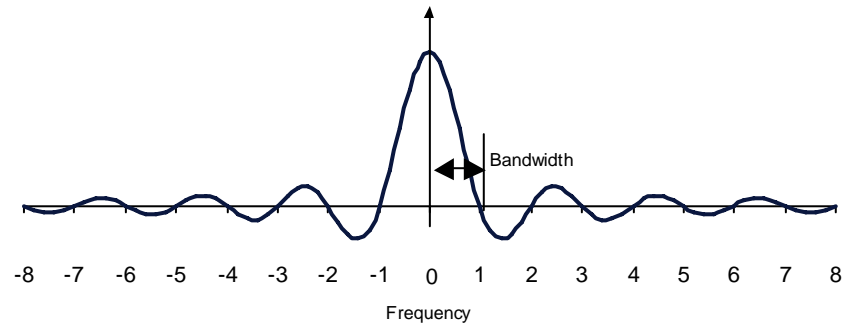


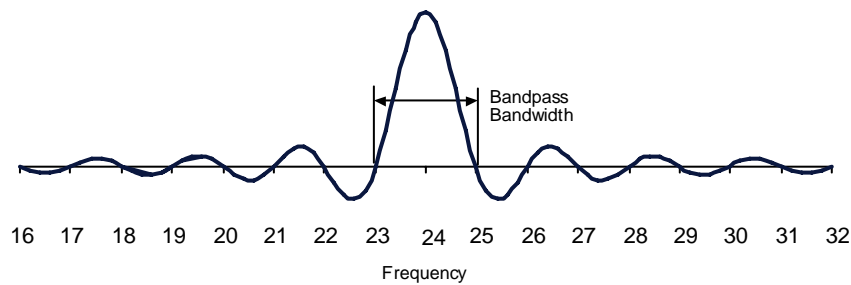
Fig 6 - The square pulse in time-domain

Since the symbol time is 1 second, the symbol rate is 1 symbol per second. The frequency response of this square pulse (its Fourier Transform) is given by the equation

$$h(t) = \frac{\sin(2\pi t / T_s)}{2\pi t} \text{ where } T_s = \text{symbol time (1 sec)}$$



(a) Lowpass



(b) Bandpass

**Fig 7 - The frequency response of the square pulse is a sinc function
lowpass bandwidth is one half of the bandpass case.**

In the above case, the symbol time is 1 second. The symbol rate hence is also equal to 1. The frequency response of the square pulse is in the shape of a sinc function ($\sin x/x$). It has a maximum amplitude of AT_s and it crosses the zero amplitude at integer multiples of R_s . The lowpass bandwidth which is defined as the distance from origin to the first zero crossing, is equal to twice the symbol rate or, 1 Hz. The bandpass case is twice that. This lowpass, bandpass business is confusing but you can understand it better if you realize that bandwidth is always measured on the positive axis.

The following figure shows square several pulses and their frequency response.

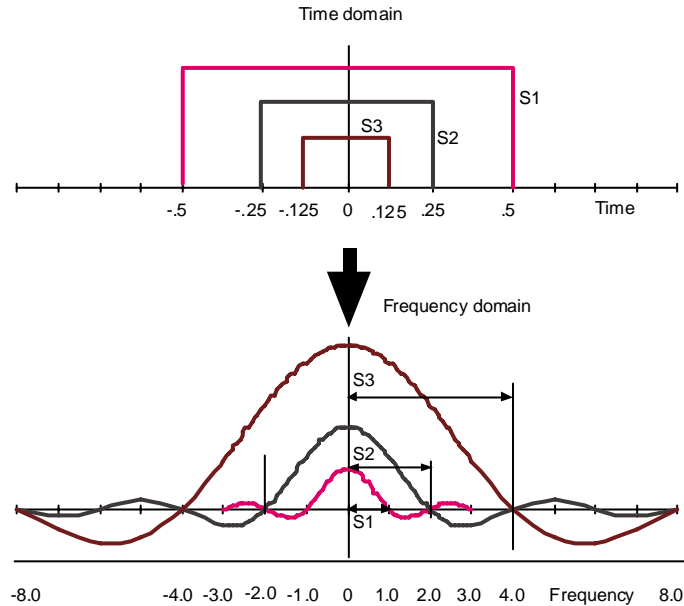


Fig 8 - The effect of square pulse symbol times and their frequency response

A narrow pulse (which is approaching a delta function) has a wide frequency response because it has a lot of frequency content. A wide one (similar to a flat line if you squint or have a good imagination) has lesser frequency content and hence its bandwidth is smaller. For each pulse, the bandwidth which we measure is only on the positive half and is equal and its symbol rate in Hz.

The important thing to note at this point is that a square pulse of symbol rate R_s has a bandwidth of R_s Hz (for bandpass signal it is twice that.)

A very important relationship:

Bandwidth of a square pulse

$$= R_s \text{ for lowpass signals,}$$

$$= 2 \text{ times } R_s \text{ for bandpass.}$$

The frequency response of the square pulse goes on forever. This is not a good thing. This would theoretical interfere with others and is not allowed by the FCC.

The square pulse has some disadvantages and they are

1. The square pulse is difficult to create in time domain because of rise time and a decay time.
2. Its frequency response goes on forever and decays slowly.
The second lobe is only 13 dB lower than the first one.
3. It is very sensitive to ISI.

If a square pulse gives us a sinc function in the frequency domain, then couldn't we use a sinc function as a pulse shape in time domain and get a brick-wall (square wave) frequency response? We can indeed.

We can use a pulse that is shaped like a sinc function instead of a square pulse and get that very nice brick-wall spectrum, with nothing spilling outside the bandwidth. This is great.

A sequences of bits shaped with sinc pulses may look like this.

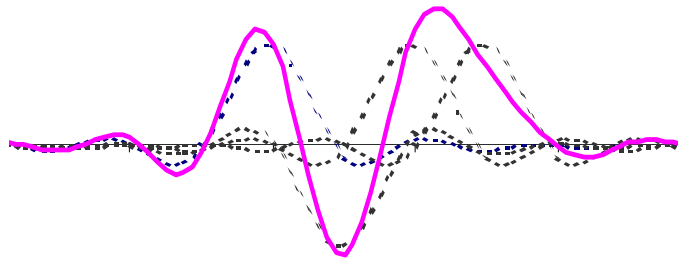
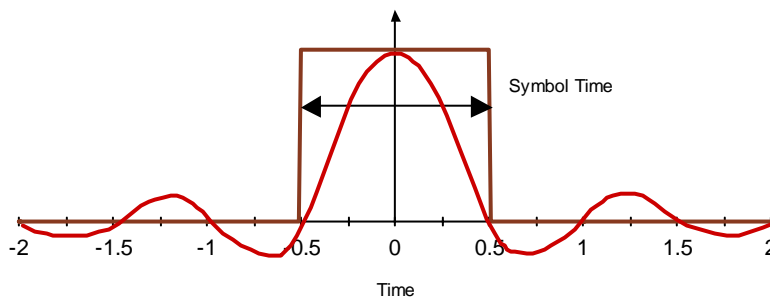


Fig 9 – Transmitting a sequence (1011) by shaping the bits as sinc pulses

Let's see what is better about using the sinc pulse as a shape instead of the square pulse. Now we change the x-axis and go the other way. The sinc pulse becomes the time domain shape. Its frequency response is the square shape.



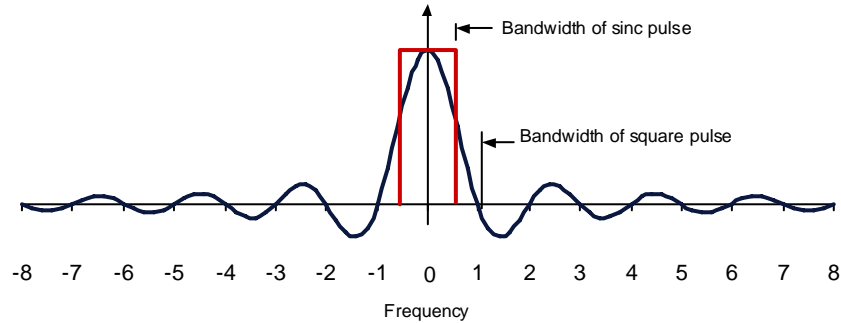


Fig 10 - The effect of sinc pulse and its frequency response

Here is the wonderful part about using the sinc pulse. As opposed to the square pulse, we see that using the sinc pulse cuts the bandwidth requirement to one-half! In the figure below, several pulses are shown of symbol rates 1, 2 and 4. In each case, we see that the required lowpass bandwidth is one-half of the symbol rate.

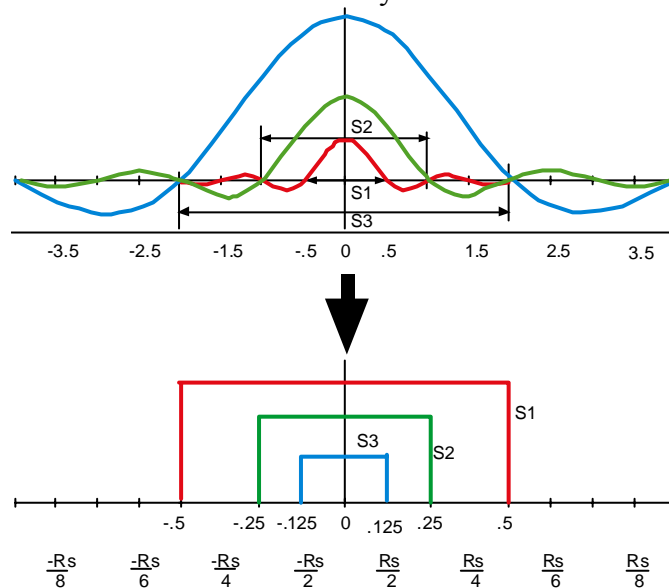


Fig 11 – Pulse width vs. frequency response

The bandwidth achieved by the sinc pulse is called the Nyquist bandwidth in honor of the man who developed it. It requires only $1/2$ Hz per symbol.

Can we find something even better? It turns out that we have not been able to find any other shape that can improve on this. It is an ultimate limit for perfect reconstruction of the signal.

Wow, two problems gone, this is great! Band-limited spectrum in frequency domain with no energy going to waste and small total bandwidth requirement. Not so great however, because a sinc pulse is actually no more possible to build than is a square pulse.

Here is what we are facing

1. In time domain a true sinc pulse is of infinite length with tails extending to infinity so the energy can theoretically continue to add up even after the signal has ended. We can only design an approximation to the real sinc pulse of a finite length. But truncation leads to an imperfect pulse that does not have a true sinc pattern and allows ISI to leak in.
2. The pulse tails that fall in the adjacent symbols decay at the rate of $1/x$ so if there is some error in timing, this pulse is not very forgiving. It requires near-perfect timing to achieve decent performance.

Nyquist offered ways to build (realizable) shapes that had the same good qualities as the sinc pulse and less of the disadvantages.

One class of pulses he proposed are called the raised cosine pulses. They are really a modification of the sinc pulse. Where the sinc pulse has a bandwidth of W , where W is specified as

$$W = 1/2T_s$$

The raised cosine pulses have an adjustable bandwidth which can be varied from W to $2W$. We want to get as close to W , which is called the Nyquist bandwidth, as possible with a reasonable amount of power. The factor α related the achieved bandwidth to the ideal bandwidth W as

$$\alpha = 1 - \frac{W}{W_0}$$

where W is Nyquist bandwidth, and W_0 is the utilized bandwidth.

The factor α is called the roll-off factor. It indicates how much bandwidth is being used over the ideal bandwidth. The smaller this factor, the more efficient the scheme. The percentage over the minimum required W is called the excess bandwidth. It is 100% for roll-off of 1.0 and 50% for roll-off of 50%. The alternate way to express the utilized bandwidth is

$$W_0 = (1 + \alpha)R_s$$

The typical roll-off values used for wireless communications range from .2 to .4. Obviously we want to use as small a roll-off as possible, since this gives the smallest bandwidth.

Here is how the class of raised cosine pulse is defined in time domain.

$$h(t) = \frac{\sin(2\pi t / T_s)}{2\pi t} \frac{\cos(2\pi \alpha t)}{1 - (2\alpha t / \pi)^2}$$

The first part is the sinc pulse. The second part is a cosine correction applied to the sinc pulse to make it behave better. The sinc pulse insures that the function transitions at integer multiples of symbol rate which makes it easy to extract timing information of the signal. The cosine part works to reduce the excursion in between the sampling instants. The bandwidth is now adjustable. It can be any where from $1/2 R_s$ to R_s . It is greater than the Nyquist bandwidth by a factor $(1 + \alpha)$. For $\alpha = 0$, the above equation reduces to the sinc pulse, and for $\alpha = 1$, the equation becomes that of a pure square pulse.

Raised Cosine Impulse Response

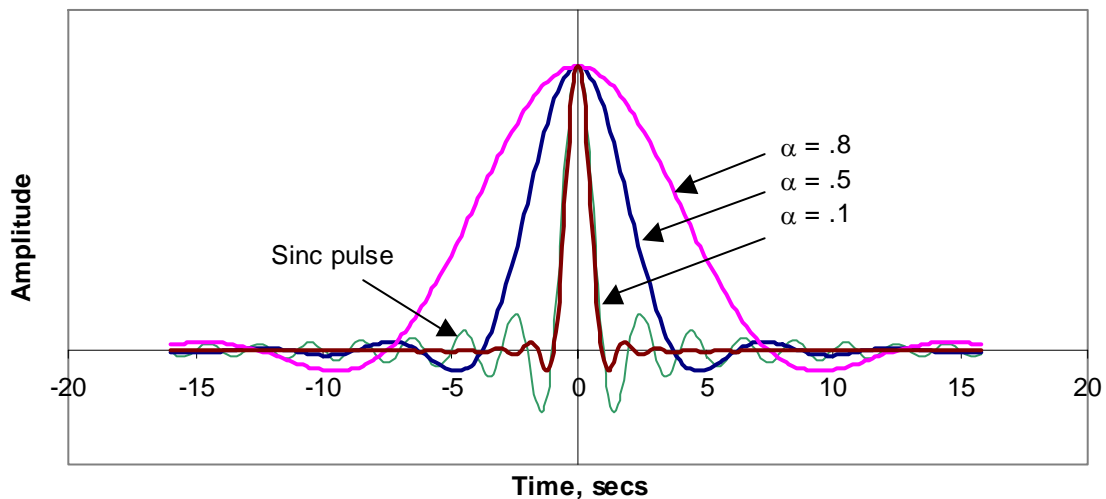


Fig 12 – Raised cosine impulse response

In frequency domain, the relationship is given by

$$H(f) = \begin{cases} 1 & \text{for } |f| \leq \frac{(1-\alpha)}{2T_s} \\ \left\{ \cos^2 \frac{\pi T_s}{2\alpha} \left(|f| - \frac{(1-\alpha)}{2T_s} \right) \right\} & \text{for } \frac{(1-\alpha)}{2T_s} \leq |f| \leq \frac{(1+\alpha)}{2T_s} \\ 0 & \text{for } |f| > \frac{(1+\alpha)}{2T_s} \end{cases}$$

Why do they call it raised cosine? Because the above response has a cosine function in the frequency domain, although other many other trigonometric representations of this equation that do not have the cosine-squared term, so it is not always clear why these are called raised cosine.

The frequency response looks somewhat like a square pulse as we would expect. A range of bandwidths are possible depending on the chosen α . The bandwidth can be anywhere from $1/2 R_s$ (this term same as W , the Nyquist bandwidth) for the sinc pulse to R_s for the square pulse. The bandwidth utilized is greater than the Nyquist bandwidth by a factor $(1 + \alpha)$. For $\alpha = 1$ the above equation reduces to the sinc pulse, and for $\alpha = 0$ the equation becomes that of a pure square pulse.

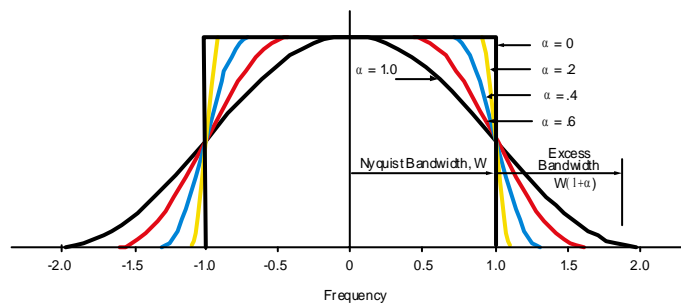


Fig 13 - The frequency response of the raised cosine pulses of $R_s = 1$

Example:

A satellite transponder has a bandwidth of 36 MHz. We use QPSK signaling. What data rate is possible using a raised cosine signaling with $\alpha = .3$?

For unshaped QPSK, which is a square pulse, the bandwidth requirement for lowpass is equal to the symbol rate. For passband, the bandwidth requirement is twice that.

$$R_s = 1/2 \ 36 \text{ MHz} = 18 \text{ Msps.}$$

If we use the ideal sinc pulse, we can send

$$R_s = B = 36 \text{ Msps}$$

If we shape the pulse with $\alpha = .3$, we can send

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

$$R_s = B/(1 + \alpha) = 27.7 \text{ Msps}$$

For QPSK, with 2 bits per symbol, we get a data rate capability of $27.7 * 2 = 55.4$ Mbps. This is about standard for most satellite systems.

Root-raised Cosine filter

To implement the raised cosine response, we split the filtering in two parts to create a matched set. When we split the raised cosine filtering in two parts, each part is called the root-raised cosine. In frequency domain, we take the square root of the frequency response hence the name root-raised cosine.

Yes, the whole raised cosine can be applied at once at the transmitter but in practice it has been found that concatenating two filters each with a root raised cosine response (called split-filtering) works better.

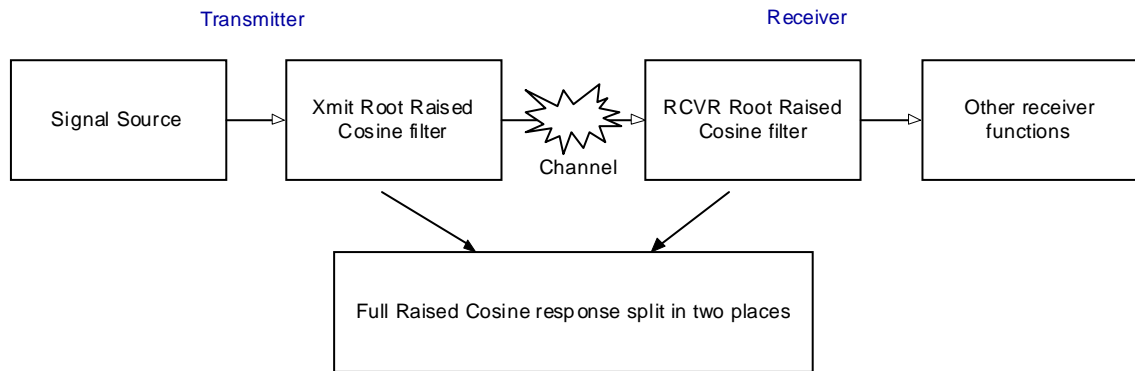


Fig 14 – Split filtering of raised cosine response, a root-raised cosine filter at the transmitter and one at the receiver, giving a total response of a raised cosine.

The root raised cosine shaping of pulses is also called baseband filtering. The frequency response of the root raised cosine is given by

$$H(f) = \begin{cases} 1 & \text{for } |f| \leq \frac{(1-\alpha)}{2T_s} \\ \left\{ \cos \frac{\pi T_s}{2\alpha} \left(|f| - \frac{(1-\alpha)}{2T_s} \right) \right\} & \text{for } \frac{(1-\alpha)}{2T_s} \leq |f| \leq \frac{(1+\alpha)}{2T_s} \\ 0 & \text{for } |f| > \frac{(1+\alpha)}{2T_s} \end{cases}$$

Compare this to the following response for the raised cosine and you see it is just a square root relationship. Each of these square root responses are applied in pairs and the total response is that of the raised cosine.

$$H(f) = \begin{cases} 1 & \text{for } |f| \leq \frac{(1-\alpha)}{2T_s} \\ \cos^2 \frac{\pi T_s}{2\alpha} \left(|f| - \frac{(1-\alpha)}{2T_s} \right) & \text{for } \frac{(1-\alpha)}{2T_s} \leq |f| \leq \frac{(1+\alpha)}{2T_s} \\ 0 & \text{for } |f| > \frac{(1+\alpha)}{2T_s} \end{cases}$$

Compare the impulse response of the root raised filter to that of the raised cosine. We do not see much of a difference except that there is a little bit more excursion in the root-raised cosine response. The time domain function is of course NOT the square root. The root part applies to frequency domain.

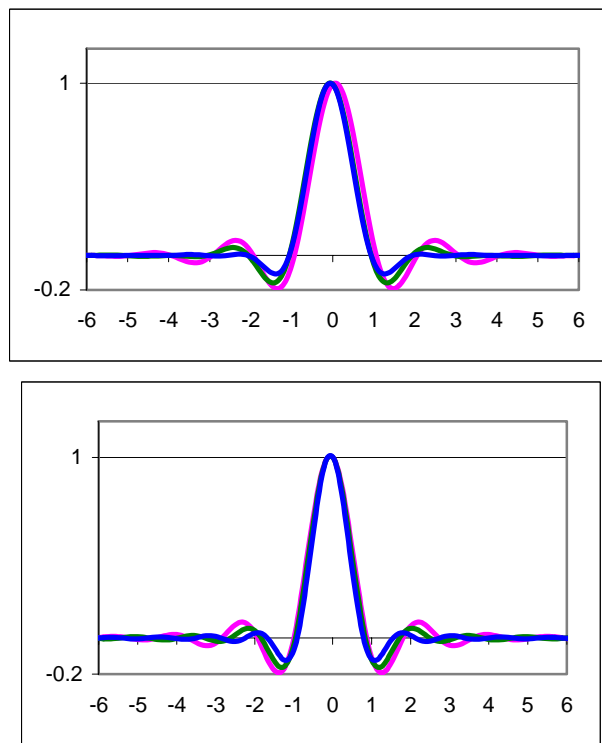


Fig 15 – Impulse response of a. Raised cosine b. Root-raised cosine

Fig 16 shows the spectrum of the root raised cosine and the raised cosine. The bandwidth of the raised cosine filter is specified at the 6 dB point due to the fact that the Nyquist frequency response is reached at 3 dB which is then doubled to give 6 dB for power. The root-raised cosine however is specified at 3 dB point because it is the square root.

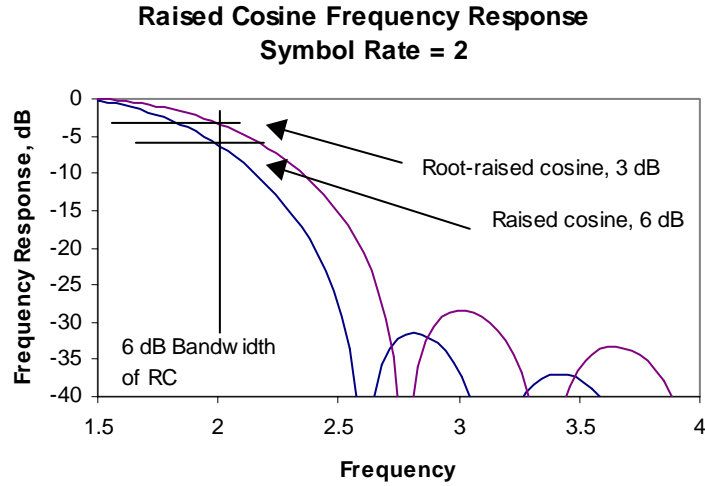


Fig 16 – Frequency response of Raised cosine and a Root-raised cosine

Looking at the eye diagram at the transmitter and the receiver gives us some further insight into how the root-raised cosine filtering splits the total filtering.

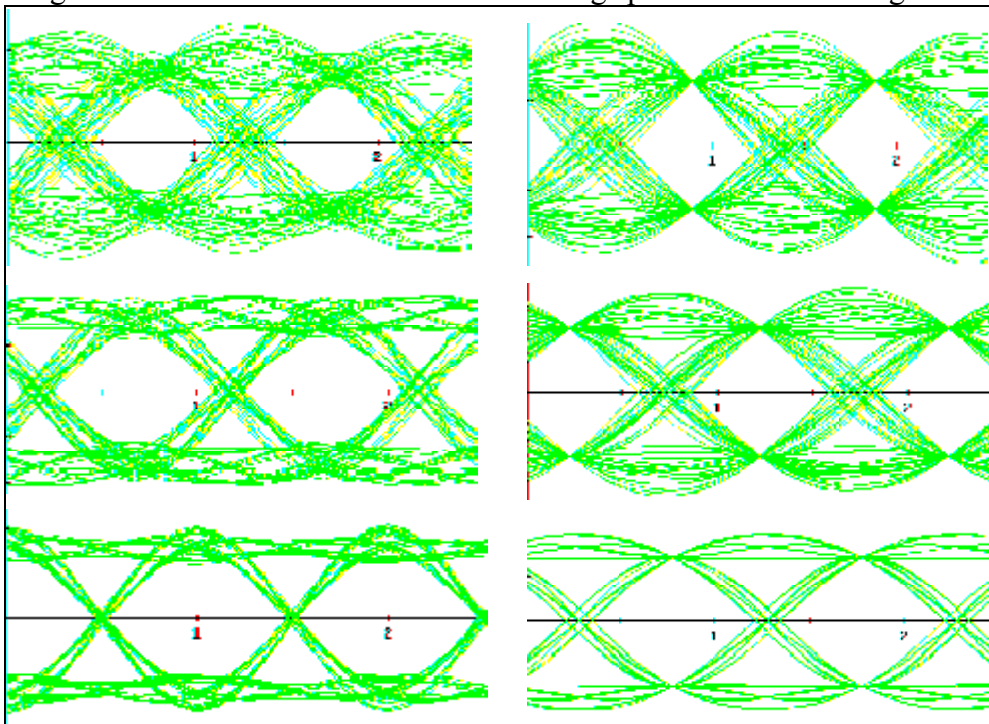


Fig 17 – Eye diagram of shaped signals, after transmit and receive root raised cosine filters

Root raised cosine pulses out of transmitter receiver

$$\alpha = .2, .4, .6$$

Root raised cosines pulses at the

$$\alpha = .2, .4, .6$$

The left side is the signal out of the first root raised cosine filter. The signal does not look that great. But when it is filtered by the second root raised cosine filter (shown on the right side) the output looks really nice, just as we hoped. As the roll-off factor gets larger, the eye opens up. This says us that if there were no bandwidth limitations, it would be much easier on the receiver if we used a large α . But since bandwidth is almost always a limiting resource, the push is on to make α as small as possible.

Design of root-raised cosine filters

The raised cosine and the root raised cosine filters are both designed as FIR filters with a specific number of taps. In time domain the fewer taps mean that the impulse response has been truncated. The response of both the raised and the root raised cosine filters is effected by choice of tap length. In general fewer taps give worse rejection. As taps numbers are increased, the length of the time domain sequence is increased and the rejection increases.

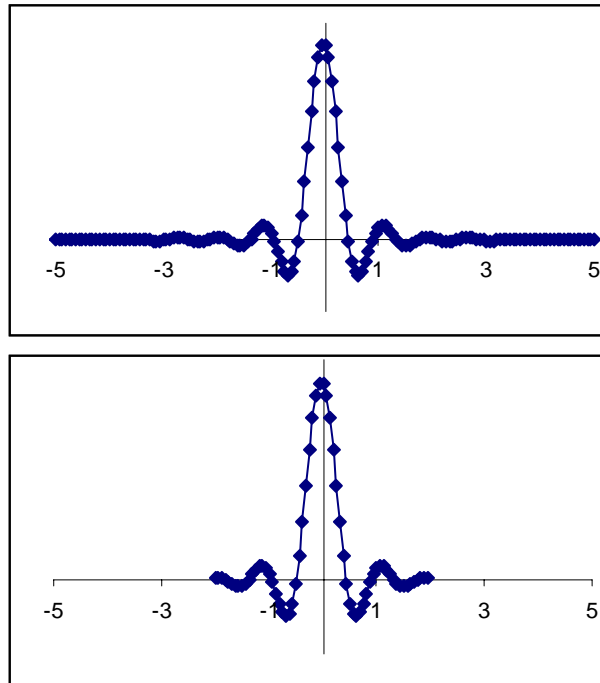


Fig 18 – RRC with taps sizes a. 128 and b. 64.

The two versions shown above are for tap length of 128 and 64. The shape for tap length 64 is exactly the same as the one for 128 taps; the only difference is that it has been truncated. The effect of this truncation is that in frequency response we get some leakage and the spectrum does not drop down as far as one for a longer tap size as shown below in Fig 19. This is true for both raised cosine and root raised cosine filters.

As we can see the rejection increases considerably as number of taps is increased. However if 3 dB bandwidths used as the discriminator, there is not a large difference so most RRC filters today use either 48 or 64 tap designs.

Figure 19 shows the spectrum of a QPSK signal through the first root-raised cosine filter of $\alpha = .3$ and various tap lengths.

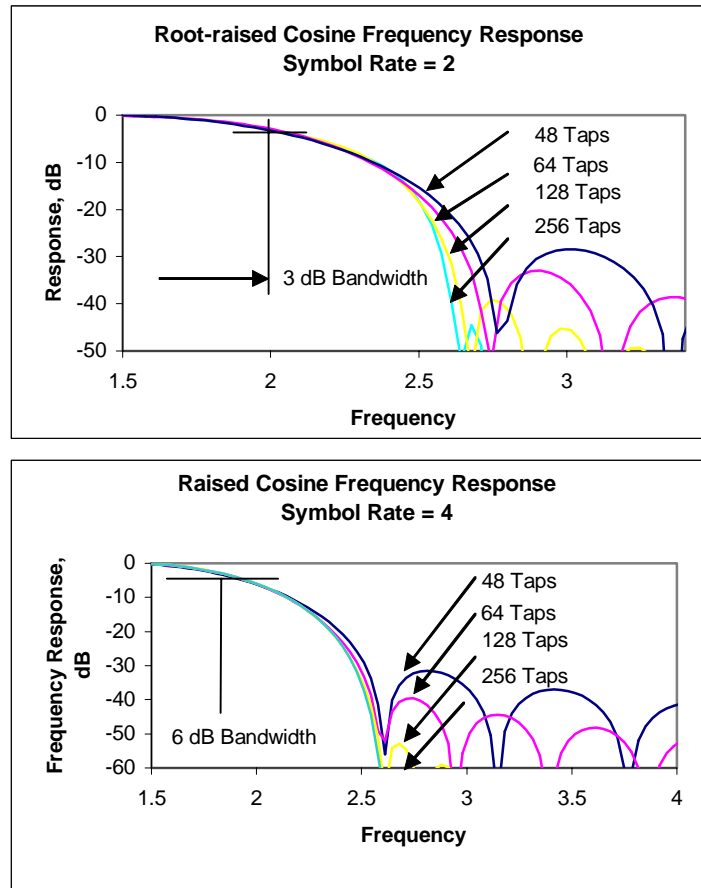
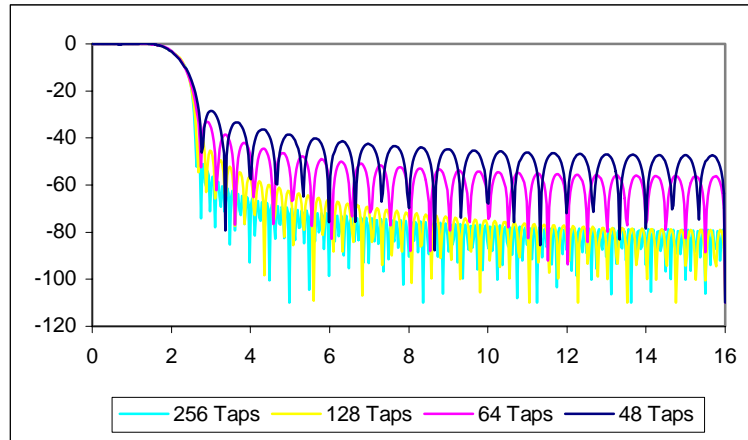
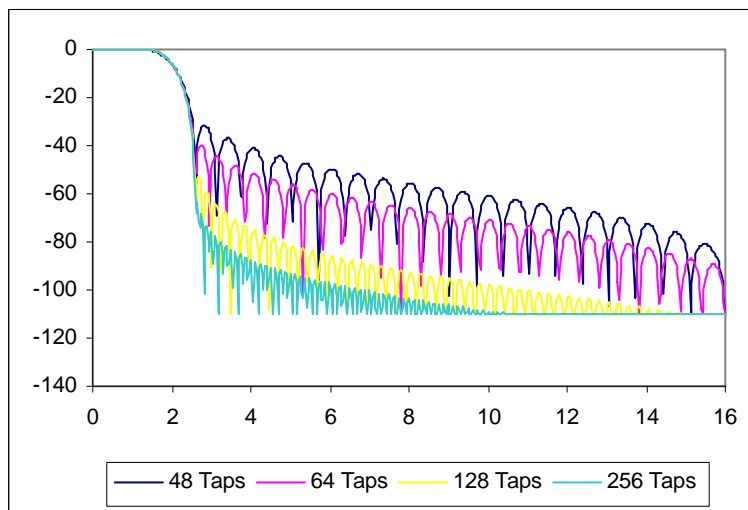


Fig 19 – Spectrum of a root raised and a raised cosine signal. (a) After the first root raised cosine at the transmitter, (b) after the second root raised filter at receiver



(a) Root raised cosine spectrum



(b) Raised cosine spectrum

Fig 20 – Spectrum of a) root raised and b) raised cosine signal.

Sinc equalization of the root raised cosine filter

In the development of raised cosine signaling, we are tacitly assuming that an ideal sampler is sampling the signal. We assume that we receive one single pulse for a 0 or a 1, which we then shape into root, raised cosine pulse. But in reality the incoming data may be analog or digital and may have to be sampled for a variety of reasons such as for conversion of sampling rates for modulation.

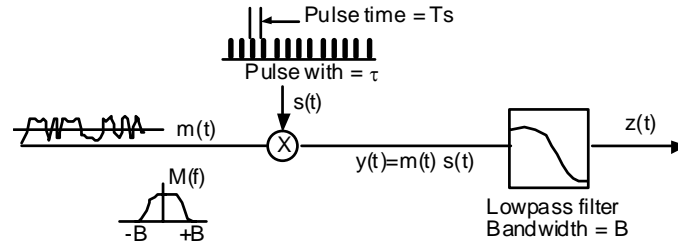


Fig 21 - Ideal vs. non-ideal sampling can lead to a sinc bias

Let's call the incoming data stream $m(t)$. It has a certain frequency response and has a bandwidth of B . For a perfect reconstruction, we need to sample it at a sampling frequency of rate $2B$ or greater. Which means that each pulse should be no more than $1/2B$ seconds apart, which defines T_s as the pulse time.

$$T_s = 1/f_s < 1/2B$$

The Fourier transform of an ideal impulse train is an impulse train in the form of frequency harmonics located at integer multiples of the sampling frequency as shown in Fig. 22. So to sample a time domain signal, with a sampling pulse train specified by $s(t)$ implies that after sampling we get a harmonically repeating response of the message stream $m(t)$. Then filter with a lowpass filter of bandwidth B to recover the original signal.

But when the sampling pulses have a finite width, they then have their own frequency response which gets multiplied by the message signal response.

$$y(t) = m(t)s(t)$$

The Fourier Transform of the two sides gives us the convolution of the Fourier Transform of the two signals.

$$Y(f) = M(f) * S(f)$$

The Fourier Transform of a finite width sampling pulses is

$$S(f) = \frac{\tau}{T_s} \sum_{n=-\infty}^{\infty} \text{sinc}\left(\frac{n\tau}{T_s}\right) \delta\left(f - \frac{n}{T_s}\right) \quad \text{where } T_s \leq \frac{1}{2W}$$

The first part is the amplitude at 0; the second part is a sinc function, which is a function of the duty cycle of the sampler. The last part just makes the whole thing made discrete by multiplying it by the delta function. The duty cycle is specified as

$$\text{duty cycle} = \frac{\tau}{T_s}$$

The duty cycle states just how wide the sampling pulse is compared to the symbol time. A large duty cycle implies a really big (or a square pulse) and a small duty cycle implies a very narrow pulse.

Using the generic response $M(f)$, for message signal $m(t)$, the convolution of $M(f)$ with the response of $S(f)$ of sampler impulse train $s(t)$ gives us the following.

$$Y(f) = \frac{\tau}{T_s} \sum_{n=-\infty}^{\infty} \text{sinc}\left(\frac{n\tau}{T_s}\right) M\left(f - \frac{n}{T_s}\right)$$

The first part again is the amplitude, the second is the sinc function and third term is the response of the message repeated at every integer multiple of the sampling frequency. The result is that we get a replicating response but its amplitude has been multiplied by the sinc function. The sinc function rolls-off according to the parameter duty cycle. A large duty cycle means the sinc function as shown below rolls off faster than when the pulse width is narrow.

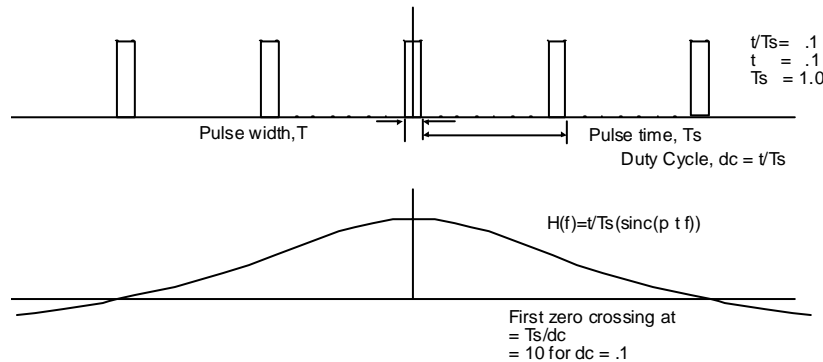


Fig. 22 - Sampling with PAM-like signal leads to a sinc bias

The smaller the duty cycle, the shallower the response since we are approaching the ideal case of zero width pulse. The first zero crossing which is important occurs at sampling time divided by the duty cycle. In this figure we see that for a duty cycle of .1, and a sampling frequency of 1, the first zero crossing is at 10 Hz.

What's the significance of all this? Well the closer this number is to the bandwidth of our lowpass filter, the worse will be the distortion introduced by this flat-top or PAM sampling.

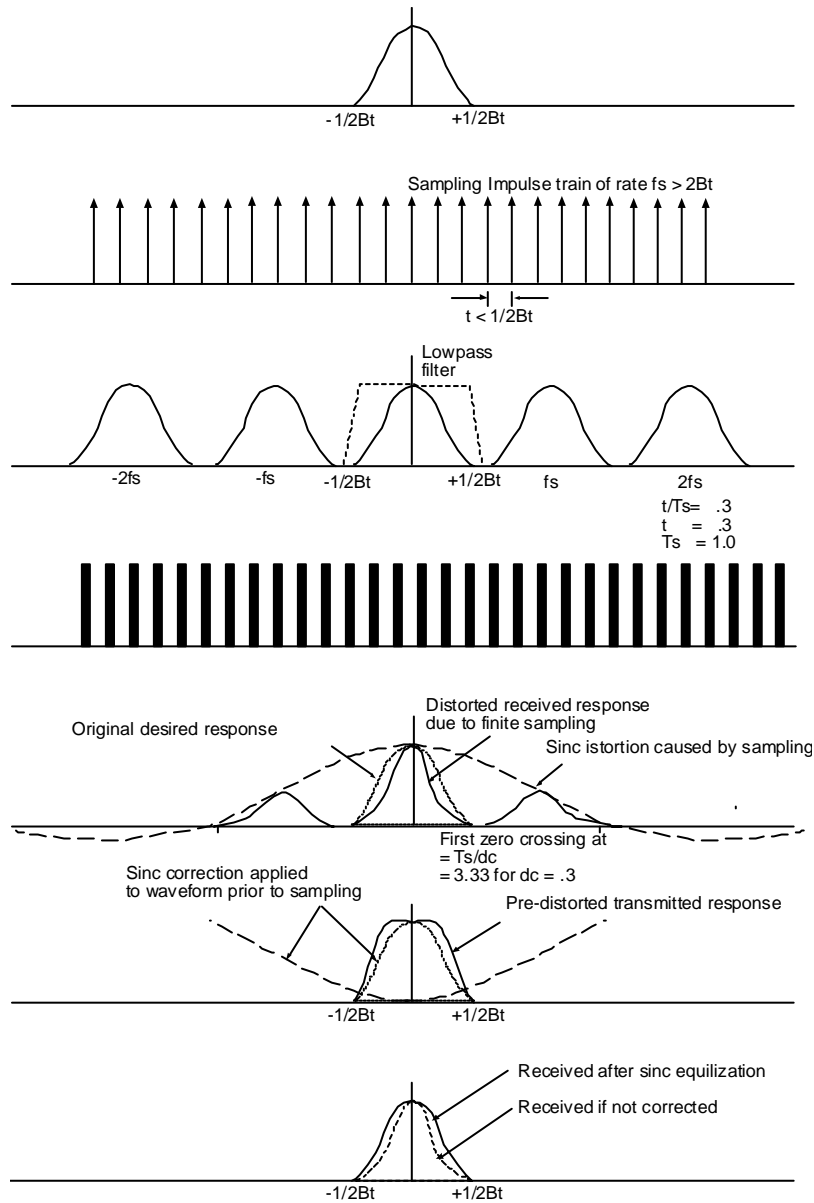


Fig 23 – Sampling with non-ideal impulses

- (a) Frequency spectrum of an arbitrary signal which has a lowpass bandwidth of B
- (b) Frequency spectrum of an ideal impulse train is also an impulse train repeated at harmonics of f_s .
- (c) Multiplying the two together gives us the replicating spectrum of message signal. This can be filtered to recover the signal.

- (d) Now we use a non-ideal sampling pulse train with duty cycle = .3.
- (e) The Fourier Transform of this sampler is a sinc function with first zero at 3.33 Hz. This distorts the spectrum of the message signal. Instead of getting the original response, we get a depressed response.
- (f) Now if we pre-multiply the signal with the inverse sinc function, then this effect is nulled out and
- (g) We can recover the message signal despite non-ideal sampling.

Once the signal is sampled by a flattop sampler, it can be recovered by using a low pass filter. But there is some energy loss, particularly in the higher frequencies due to the downward sloping bias of the sinc function. This distortion can be significant in two cases, a. when the duty cycle is large, say over .2 and b. when the roll off factor is small. The effect of large duty cycle is that there is in-band distortion. However for Duty Cycles less than .2, the distortion is quite minimal and can be ignored. But anything above that needs to be corrected if we are to make raised cosine pulse perform as it should. Since the first zero crossing occurs at T_s/dc , then a large dc gives us a close-in zero crossing, hence causes a lot of distortion.

The interaction of roll-off factor with this effect is that a smaller roll-off factor response is approaching a brick wall spectrum and has large spectrum content in higher frequencies which are affected more by the distortion. A larger roll factor is less affected but then it is not so desired either because it needs a larger frequency bandwidth.

The figure below shows the amount of correction (or equalization) required for various roll off factor. Any roll-off factor below .4 definitely requires sinc equalization.

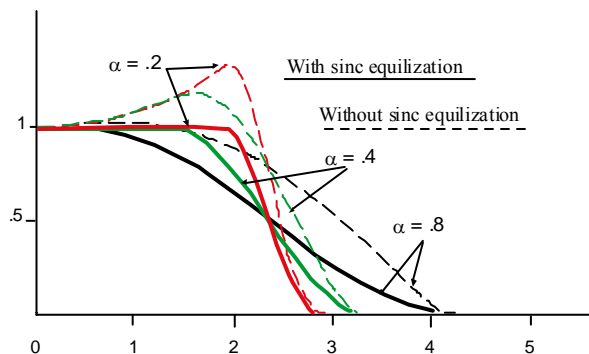


Fig 24 - Sinc equalization affects low roll-off signals a lot more

We can pre-correct for this distortion by multiplying the desired response by the inverse sinc function. See the pre-corrected response as opposed to uncorrected response in Fig 24. The signal is then sampled and returned to the desired spectrum as shown in Fig 23 (f).

The frequency response of a sinc corrected raised cosine spectrum is given by the following equation.

$$H(f) = \begin{cases} \frac{\pi f T_s}{\sin(\pi f T_s)} & \text{for } |f| \leq \frac{(1-\alpha)}{2T_s} \\ \frac{\pi f T_s}{\sin(\pi f T_s)} \left\{ \cos^2 \frac{\pi T_s}{2\alpha} \left(|f| - \frac{(1-\alpha)}{2T_s} \right) \right\} & \text{for } \frac{(1-\alpha)}{2T_s} \leq |f| \leq \frac{(1+\alpha)}{2T_s} \\ 0 & \text{for } |f| > \frac{(1+\alpha)}{2T_s} \end{cases}$$

Where the root-raised response at the transmitter and the receiver is given by the following equations. Note that only the transmitter has a sinc correction.

Transmit filter response

$$H_{XMT}(f) = \begin{cases} \frac{\pi f T_s}{\sin(\pi f T_s)} & \text{for } |f| \leq \frac{(1-\alpha)}{2T_s} \\ \frac{\pi f T_s}{\sin(\pi f T_s)} \left\{ \cos \frac{\pi T_s}{2\alpha} \left(|f| - \frac{(1-\alpha)}{2T_s} \right) \right\} & \text{for } \frac{(1-\alpha)}{2T_s} \leq |f| \leq \frac{(1+\alpha)}{2T_s} \\ 0 & \text{for } |f| > \frac{(1+\alpha)}{2T_s} \end{cases}$$

The receive filter response (note has no sinc correction)

$$H_{RCV}(f) = \begin{cases} 1 & \text{for } |f| \leq \frac{(1-\alpha)}{2T_s} \\ \left\{ \cos \frac{\pi T_s}{2\alpha} \left(|f| - \frac{(1-\alpha)}{2T_s} \right) \right\} & \text{for } \frac{(1-\alpha)}{2T_s} \leq |f| \leq \frac{(1+\alpha)}{2T_s} \\ 0 & \text{for } |f| > \frac{(1+\alpha)}{2T_s} \end{cases}$$

In frequency domain we are transmitting the distorted shape and not the ones that are shown in solid lines in Fig 24.

There are other ways to mitigate ISI such as with duo-binary or partial response signaling. These might be a topic later on.

Charan Langton

Copyright 1998, 2002 Charan Langton

I can be contacted at
mntcastle@earthlink.net

Other tutorials at
www.complextoreal.com