Tutorial 16
By Charan Langton

Signal Processing and Simulation Newsletter

Partial Response signaling and Quadrature Partial Response (QPR) modulation

Nyquist limit states that the number of symbols you can safely transmit in a channel of bandwidth 1 Hz that has inter-symbol interference is 2 symbols per second. But to achieve this bandwidth efficiency you need a brick wall filter, which is not build able. Raised cosine signaling helps counter ISI but at the cost of increased bandwidth.

> **Fact 1** : *A channel with no ISI using a raised cosine signaling with alpha = 0, allows us to successfully transmit a symbol rate equal to twice the bandwidth.*
>
> **Fact 2** : *Since practical alpha is around .2 to .3, the realistic symbol rate possible is only* $(1 + \alpha)W$*, with* αW *as the excess bandwidth over the theoretical limit.*

One of the main reasons we can not transmit 2 symbols per Hz is inter-symbol interference. Raised cosine signaling is one way to counter ISI. But it limits the symbol rate that can be safely transmitted. Is there some other way that also counters ISI but allows us to achieve 2 symbols per Hz of bandwidth?

Partial Response signaling, also called Quadrature Partial Response (QPR) a concept that is used both for pulse shaping and as a way to modulate information was proposed by Adam Lender in 1964. The technique also goes by the general name of ***correlative coding***.

There is an alphabet soup here that is often confusing. *Partial response, correlative coding, duobinary, modified duobinary, Class I coding*, are all names for the same thing.

Inter-symbol interference can be defined as spreading of symbol energy into adjacent symbol. The result of which is that the adjacent symbol shape is corrupted. However, with raised cosine signaling, the interference can be controlled such that if it is zero at the slicer timing pulse, then its effect is negated. It does not matter how much ISI distorts the signal as long as we know at the time of the sampling instant what the amount of that interference is. This amount is zero for raised cosine signaling because the signal shape (a sinc function) is forced to pass through zero as shown in Fig. 1. Although the interference is present, it is zero just at the instant we sample the signal to make decision about if it's a 0 or a 1. This means that at the sampling instant, the sampled voltage belongs only to the pertinent symbol, so the decision based on this value is likely to be correct.

And that's the concept of *controlled introduction of ISI*.

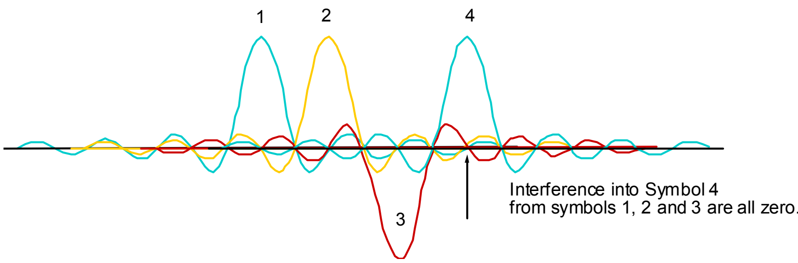


Fig 1 – Even though the pulse spreads out over a large time interval, at the moment of the sampling, it is zero for the raised cosine signal. So its effect has been nulled.

Partial response signaling (and its various names) offers another way to introduce a form of controlled interference. But unlike raised cosine signaling, it allows us to transmit 2 symbols per Hz achieving full theoretical capability.

Let's see how it works.

Take the following random sequence of bits.

10100010000111, shown below in polar format that is 1 volt for a 1 and –1 volt for a 0.

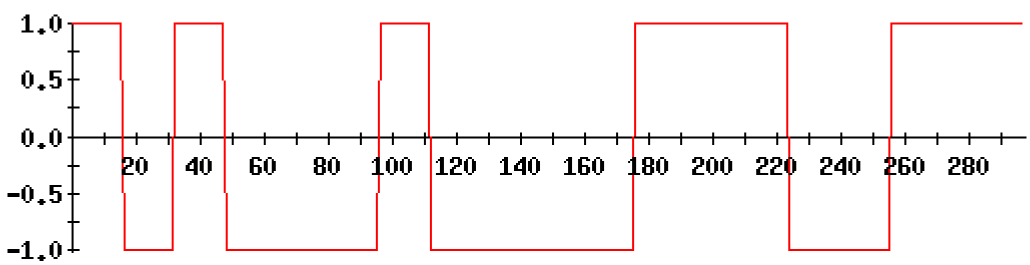


Figure 2 – Random binary sequence, 10100010000111

The raised cosine shaping of this sequence is simple and intuitive as shown below. Each bit is shaped individually. Each pulse is independent and does not depend on any other. It is a two level signal as shown in the eye diagram below. At slicing point, the voltage level can be mapped to only one of two possible values, a 0 or 1. Decoding consists of applying this rule; if the voltage level detected is greater than 0, then the bit is a 1 and if the voltage level is less than 0, then the bit is a 0.

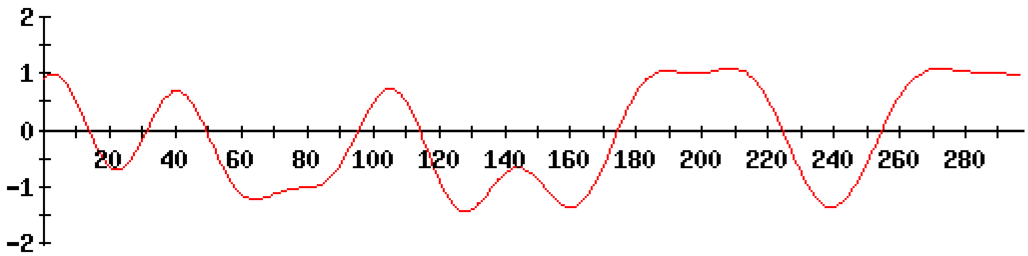


Figure 3 – Raised cosine pulses for the bit sequence 10100010000111

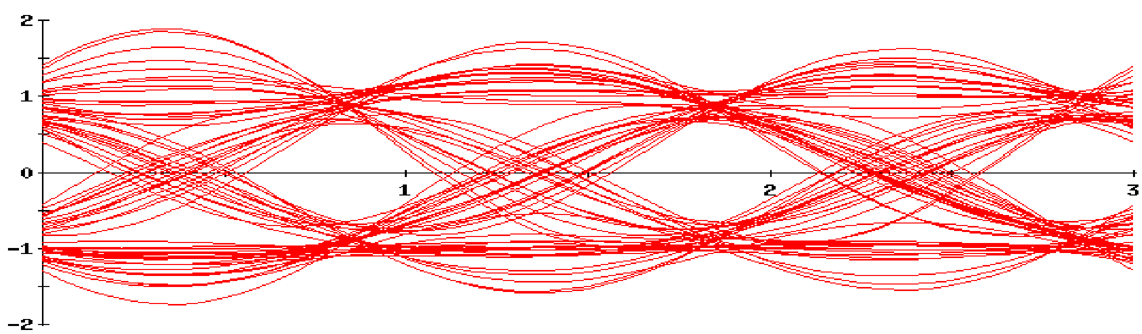


Figure 4 – Eye pattern of the raised cosine signal for the bit sequence 10100010000111

Duobinary – the first from of partial response

In Duobinary which is a specific type of partial response signaling, bit shaping is done in pairs of bits. The bits are shaped such that the transmitted pulse is based on the sum of last two bits. In this way, each symbol transmitted is dependent on what the previous symbol was.

Here is how to create the duobinary pulse. First we take the binary signal of symbol time T and delay it by one symbol as is shown in Fig. 5. Then add these two symbols. Since the signal is a square wave, at this point it has unlimited bandwidth. Now cutoff with an ideal brick wall filter of cutoff frequency f = 1/2T. This gives us 2 symbols for Hz.

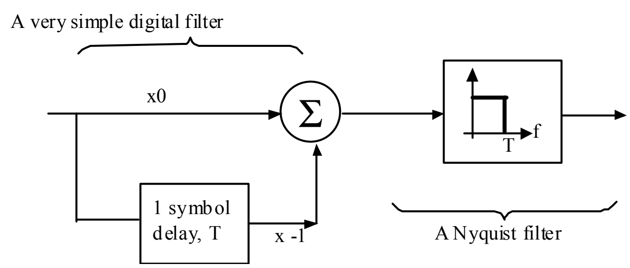


Figure 5 – Conceptual duobinary signal generation by first adding two bit together and then brick wall filtering them.

We will create a new signal by doing two things.

1. Delay and add the bits and then
2. Filter the resulting sequence by a brick wall filter such that it has no frequency components above the bandwidth, W.

Bit sequence

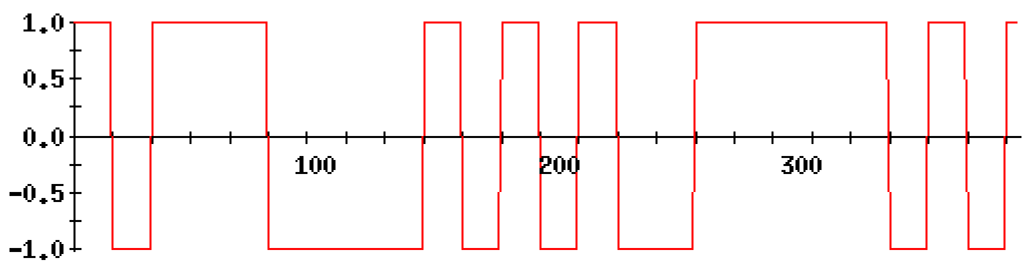


Figure 6 – Random sequence No. 2, consisting of bits 101110000101010011

Step 1. Delay and Add operation

Bit		1	0	1	1	1	0	0	0	0	1	0	1	0	1	0	0	1	1
Signal	1*	1	-1	1	1	1	-1	-1	-1	-1	1	-1	1	-1	1	-1	-1	1	1
Add $x_i + x_{i-1}$		2	0	0	2	2	0	-2	-2	-2	0	0	0	0	0	0	-2	0	2

* Intitalization

1. The first row shows bits to be transmitted,
2. The second is in its polar voltage levels. 1 for 1 and -1 for a 0.

3. In third row we add the current bit and the one before. The first tow bits added give us a 2. The second two bits give a 0.
4. In fourth row, the pulse is then created based on this three-level sum. If we get a 2, then the pulse amplitude is 2 and so on.

This sum (row 3) is shown in Fig. 7 below.

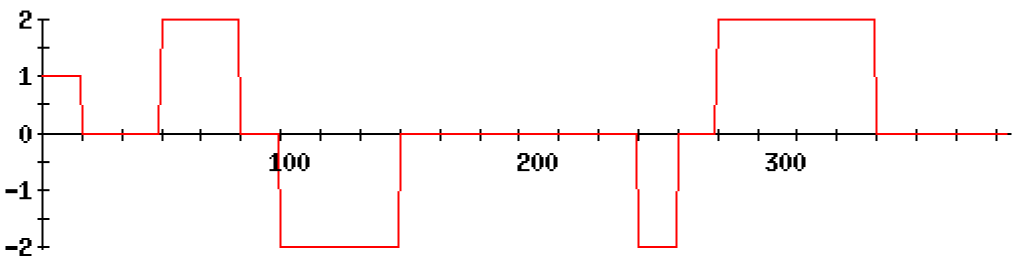


Figure 7 – signal formed by adding consecutive bits. Level can be more than 2 or less than -2.

Step 2 – filter the sequence by a brick wall filter

Now assume that we do have a brick wall filter. We pass the signal through this brick wall filter cutting off all frequencies above W Hz. What we get is a signal that looks like as shown below. This is the partial response signal for this bit sequence. At each slicing point, the voltage is mapped to three levels, 0, 2 or -2 computed by adding consecutive bits. (the numbers we obtained in row 3 above.)

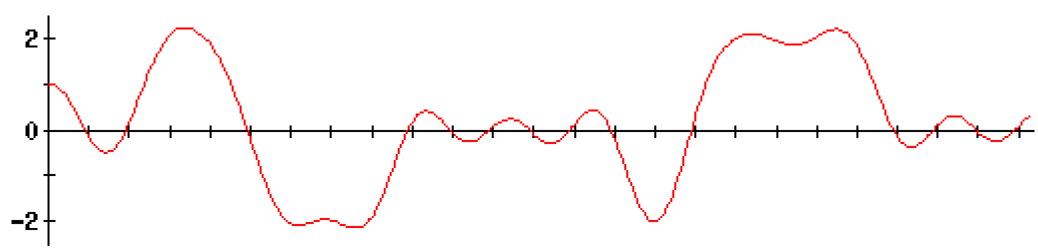


Figure 8 – Partial response pulse for the above bit sequence. Note that the signal value at the timing instant is equal to the sum in Row 3.

At the receiver, the following rules tell the receiver how to decode.

Rule 1: If received level is 2, then the transmitted bit was a 1.
Rule 2: If received level is -2, then the transmitted bit was a 0.
Rule 3: If received level is 0, then the bit is opposite of the previously decoded bit.

Decode it now.

Received level	2	0	0	2	2	0	-2	-2	-2	0	0	0	0	0	0	-2	0	2
Decoded bit	1	0	1	1	1	0	0	0	0	1	0	1	0	1	0	0	1	1

This is just what was transmitted.

This type of pulse shaping is called duobinary or class I partial response. The flaw in the above algorithm is that a Nyquist filter which we used in step 2 to limit the bandwidth is not realizable. So unless we can figure out some alternate to the Nyquist filter block in figure 5, the method will not work.

This is where some math comes in handy. We will find an alternate transfer function that is equivalent to the above process but can actually be built. But we are going to have to resort to some heavy looking equations.

The delay and add block is actually an elementary digital filter with just one tap. Its impulse response for a delay of T sec is equal to

$$H_1(\omega) = 1 + e^{-j\omega T} \qquad 1$$

If you have a signal h(t) and its FT is H(ω), then if you delay the signal by time T sec, the FT of the delayed function is H(ω) times an exponential e^{-jwT} . This is called the shifting property of Fourier transform and shown below in usual mathematical terms.

$$h(t) \Leftrightarrow H(\omega)$$
$$h(t-T) \Leftrightarrow H(\omega)\, e^{-j\omega T}$$

Now the second part, the one about the Nyquist filter. We need to cut the frequency at W. This is the Nyquist filter and is mathematically given as

$$H_2(w) = \begin{Bmatrix} 1 & for\ f \le \pm\dfrac{1}{2T} \\ 0 & for\ f \succ \pm\dfrac{1}{2T} \end{Bmatrix} \qquad 2$$

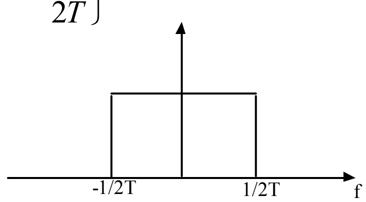


Figure 9 – Frequency response of Nyquist filter.
This filter has a bandwidth of half or symbol rate Rs which is equal to inverse of symbol time T.

The total impulse response (which is what we want) is the product of the response of the two parts. Multiplying the response of 1-tap delay filter and the Nyquist filter we get,

$$H(\omega) = H_1(\omega) H_2(\omega) \qquad 3$$

$$= 1(1 + e^{-j\omega T})$$

Now we invoke an another basic equation, curtsey of Euler,

$$\frac{e^{-j\omega T} + e^{j\omega T}}{2} = \cos\omega T \qquad 4$$

Dividing both sides by of Eq 4 by e^{jwT}, we get

$$1 + e^{-j2\omega T} = 2\cos\omega T \; e^{-j\omega T}$$

Now substitute T/2 for T on both sides

$$1 + e^{-j\omega T} = 2\cos\frac{\omega T}{2} \; e^{-j\frac{\omega T}{2}}$$

Substitute this equation into Eq, 1 , we get

$$H(\omega) = (1 + e^{-j\omega T}) = 2\cos\frac{\omega T}{2} \; \underline{e^{-j\omega T}} \qquad 5$$

The response of the combined filters can now be written as

$$H(\omega) = 2\cos\frac{\omega T}{2} \qquad 6$$

(We just threw away the complex exponential (underlined terms in Eq 5).. Why? Because multiplication by a complex exponential in frequency domain merely shifts the spectrum to the right or left but has no effect on the amplitude. That simplifies things.)

The frequency response of the combined process is a cosine filter. (Not to be confused with raised cosine or root raised cosine filters we talked about in the last paper. This is all together different. This term confusion and not just the math is part of what makes this field so complex.) So we see how handy math can be. It took a two step operation and replaced it with just one cosine filter. Hard to see if you don't do the math.

The total response of this cosine filter is limited to the band W = 1/2T since it was multiplied by the brick wall filter.

$$H_2(w) = \begin{Bmatrix} 2\cos\dfrac{2\omega}{T} & for\ f \le \pm\dfrac{1}{2T} \\ 0 & for\ f \succ \pm\dfrac{1}{2T} \end{Bmatrix}$$

The frequency response of this cosine filter is shown in figure 10. The spectrum is very nicely behaved, goes to zero gracefully and can be built with fairly good accuracy. We see that its bandwidth (remember, its only the positive side) is .5. For a symbol rate of 1, the required bandwidth is .5, giving us 2 symbol per Hz of bandwidth.

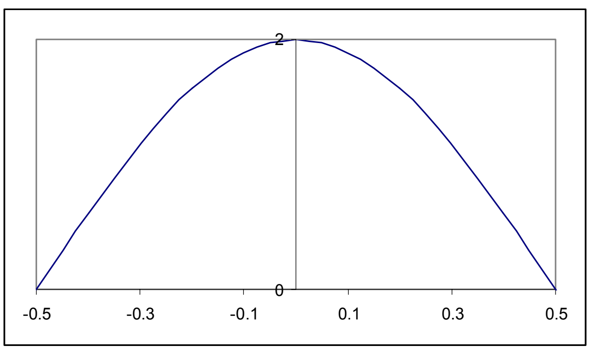


Fig 10 – Spectrum of a cosine signal. This is equivalent filter which produces the partial response signal.

So that's how it looks in frequency domain. How do we create it time domain? The time domain shape for this response is found by taking the inverse FT of the cosine function. This is a very lovely satisfying combination of two sinc pulses time shifted by Tsec.. (Remember that the Fourier transform of a cosine function are two impulses.)

$$h_e(t) = \sin c\left(\frac{t}{T}\right) + \sin c\left(\frac{t-T}{T}\right)$$

In time domain, the pulse has the shape shown below in Fig. 11. It is sum of two sin functions.

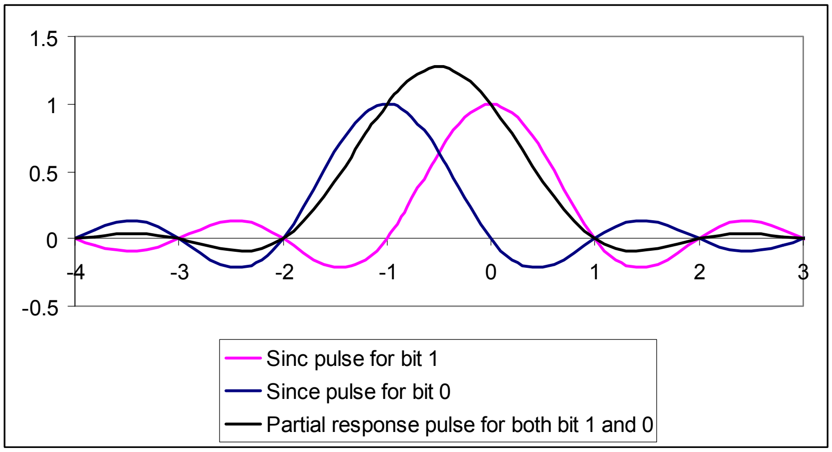


Figure 11 – Sinc pulses of two 1 bits and the resulting partial response pulse

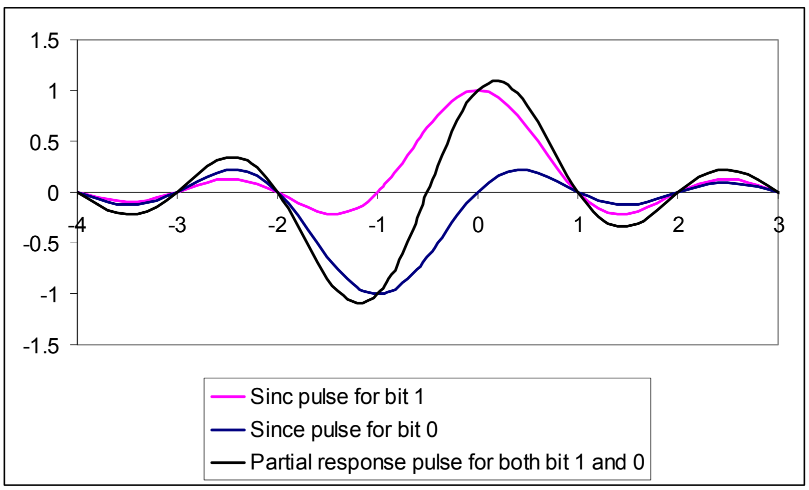


Figure 12 – Sinc pulses of 0 and 1 bits and the resulting partial response pulse

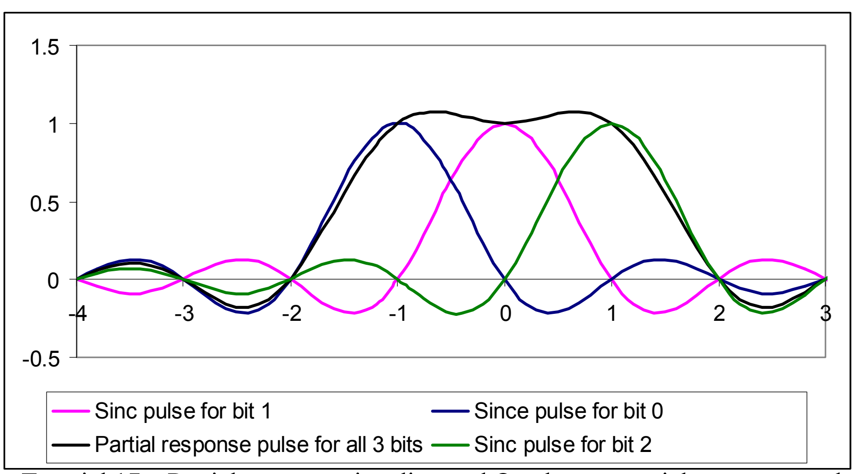


Figure 13 – Sinc pulses of three 1 bits and the resulting partial response pulse

The purpose of doing all that math was to show that we can replace the add and delay and the Nyquist filter by a much simpler and completely equivalent cosine function for the transmitter as shown below.

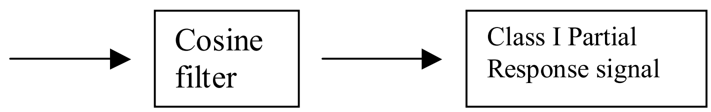


Figure 14 – A binary signal filtered by a cosine filter produces the duobinary partial response pulse.

There are numerous variations possible on this theme of adding together sinc functions. The most common is the modified duobinary pulse. The following block diagram is for a pulse which is created by adding not the last bit but the one before the last bit. This is called class 4 partial response signaling.

$$y_i = x_i + x_{i-2}$$

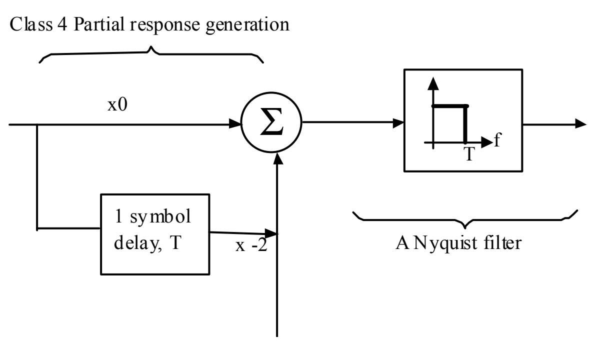


Figure 15 - Class 4 partial response signal

For Class 2 signal, we add not one but two previous bits and get,

$$y_i = x_i + 2x_{i-1} + x_{i-1}$$

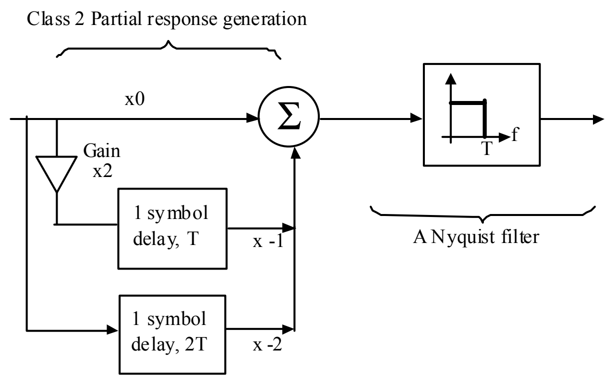


Figure 16 - Class 2 partial response signal

Adding three bits together produces a five level signal rather than a 3 level signal as for Class 1 and Class 4.

A generalized response of partial response pulses can be written as

$$H_1(\omega) = \sum_{n=0}^{N-1} h_n e^{-j\omega nT}$$

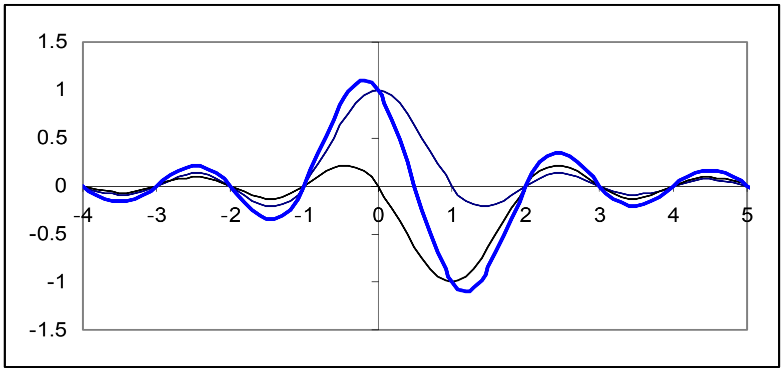
So for the class 4 shown in Fig 15, the frequency response for the delay add filter is

$$= 1 - e^{-j2\omega T}$$

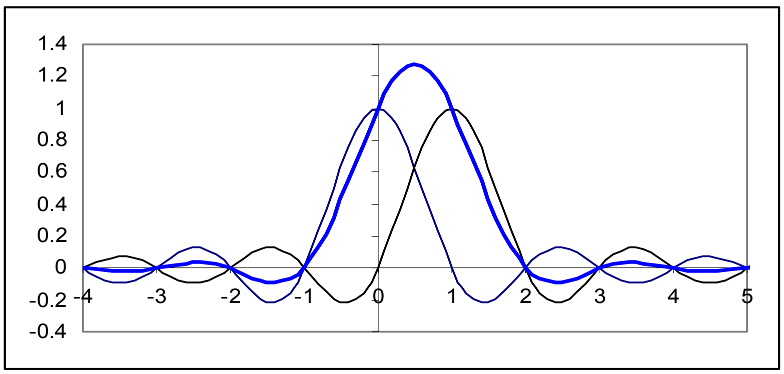
And its characteristic is

$$\left|H_1(\omega)\right| = 2\sin\omega T$$

Perhaps you recall from the chapter on Fourier transform that a sine function in frequency domain gives us two impulses in time domain but of opposite signs. So the time domain signal of a Class 4 signal is composed two opposite sign sinc functions added together.



(a) Modified duobinary pulse, Class 4



(b) Duobinary pulse, Class 1

Figure 17 – Pulses for Class1 and Class 4 partial response. Class I is the sum of two same sign sinc pulses, Class 4 is the sum of two opposite sign pulses.

This scheme is called **modified duobinary**. How does it differ from duobinary?

1. Duobinary is composed of the sum of two sinc functions of the same sign.
2. Modified duobinary consists of sum of two opposite sign sinc functions.

The main advantage modified duobinary has over the Class I (duobinary) is that it has a bimodal spectrum, so it is good for those circuits where it is difficult to pass low frequency energy. Figure 20 shows the spectrum for a class 1 signal and Fig. 21 shows the spectrum for a class 4 signal which is a bimodal function with no energy at dc.

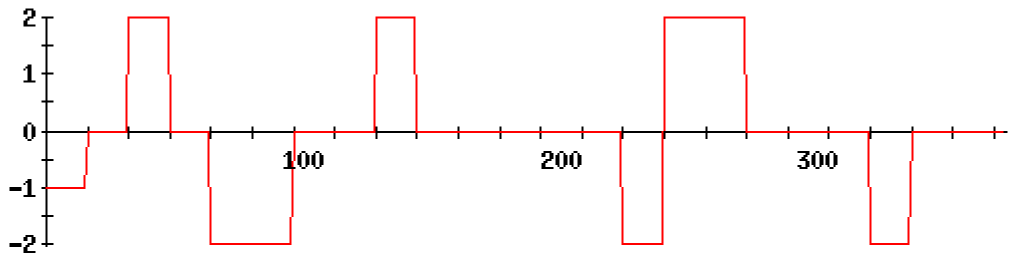


Figure 18 – Signal levels for Modified duobinary signal
For bit sequence No. 2

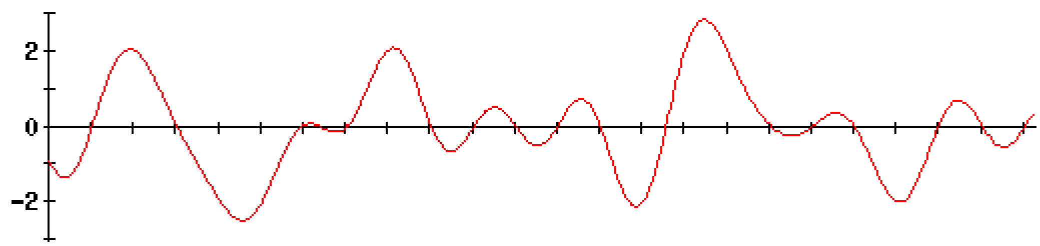


Figure 19 – Partial response signal for Modified duobinary signal
For bit sequence No. 2

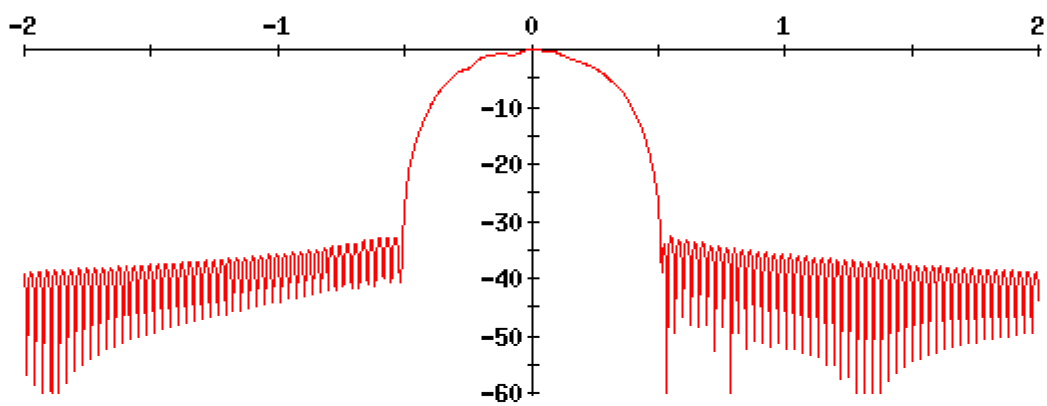


Figure 20 - Class 1 partial response signal spectrum for a signal of symbol rate 1. Note that the low pass bandwidth is .5 Hz.

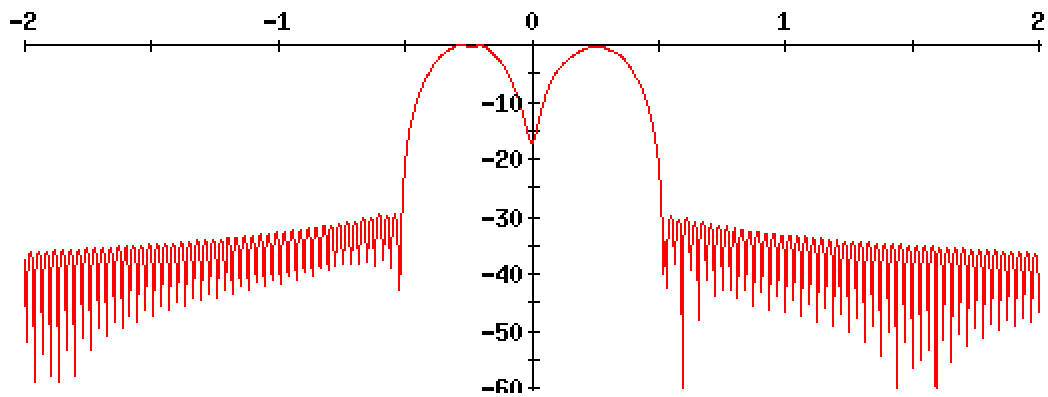


Figure 21 - Class 4 partial response signal spectrum for a signal of symbol rate 1. Note that the low pass bandwidth is still .5 Hz but there is no energy at dc

Numerous other schemes are possible. The more terms that are added, the more levels the signal will have. Following table lists the signals and the equivalent filters needed to produce the signal.

Table I – A variety of partial response signals can be formed by adding or subtracting the signal copies.

Class	Method	Transfer function, H(w)	Number of level
1	$x_i + x_{i-1}$	$2\cos\left(\frac{\omega T}{t}\right)$	3
2	$x_i + 2x_{i-1} + x_{i-2}$	$4\cos^2\left(\frac{\omega T}{t}\right)$	5
3	$2x_i + x_{i-1} - x_{i-2}$	$2 + \cos\omega T - \cos 2\omega T$	5
4	$x_i + x_{i-2}$	$2\sin\omega T + j(\sin\omega T - \sin 2\omega T$	3
5	$x_i + 2x_{i-2} + x_{i-4}$	$4\sin^2\left(\frac{\omega T}{t}\right)$	5

Ref 1 – Table 5.1

Of these classes 1 and 4 are most popular. The others are used in creating higher order QPR signals.

There is one major problem with partial response. Since the received pulses are correlated, errors have a tendency to propagate. To counter this, an extra encoding step is added. Let's see how this is done and how it helps.

Precoding the partial response pulse

This method was proposed by Lender, the creator of this signaling method. Precoding is basically a form of differential encoding. First the input bits are transformed into a sequence by the exclusive OR rule.

Without precoding

Decoding Rules

If y = -2, then bit = 1
If y = -2, then bit = 0
If y = 0, then bit is opposite of pervious bit

Input bit sequence

x		0	1	1	1	0	1	0	1	1	0
polar	-1	-1	1	1	1	-1	1	-1	1	1	-1
y		-2	0	2	2	0	0	0	0	2	0

Receive and decode per rules as

z		0	1	1	1	0	1	0	1	1	0

With precoding

Decoding Rules

If y = -2, then bit = 0
If y = -2, then bit = 0
If y = 1

Input bit sequence

X		0	1	1	1	0	1	0	1	1	0
B	0	0	1	0	1	1	0	0	1	0	0
polar	-1	-1	1	-1	1	1	-1	-1	1	-1	-1
Y		-2	0	0	0	2	0	-2	0	0	-2

Receive and decode

z		0	1	1	1	0	1	0	1	1	0

In both cases we decoded correctly. But now let's invert one of the received bits, shown in red to indicate an error condition.

Without precoding

	y	-2	0	2	0	0	0	0	0	2	0
z		0	1	1	0	1	0	1	0	1	0

One error in the fourth bit leads to five bits decoded incorrectly.

With precoding

Y		-2	0	0	2	2	0	-2	0	0	-2
Z		0	1	1	0	0	1	0	1	1	0

With precoding, one error causes only one error during decoding. Errors do not propagate because decision is made based only on one bit.

Quadrature Partial Response (QPR) modulation

When a bi-level signal such as the binary signal with 1s and 0s is input into a duobinary or a modified duobinary filter, a three level signal is produced. So instead of a four point constellation of a QPSK signal, we would get a 9 point constellation as shown below.

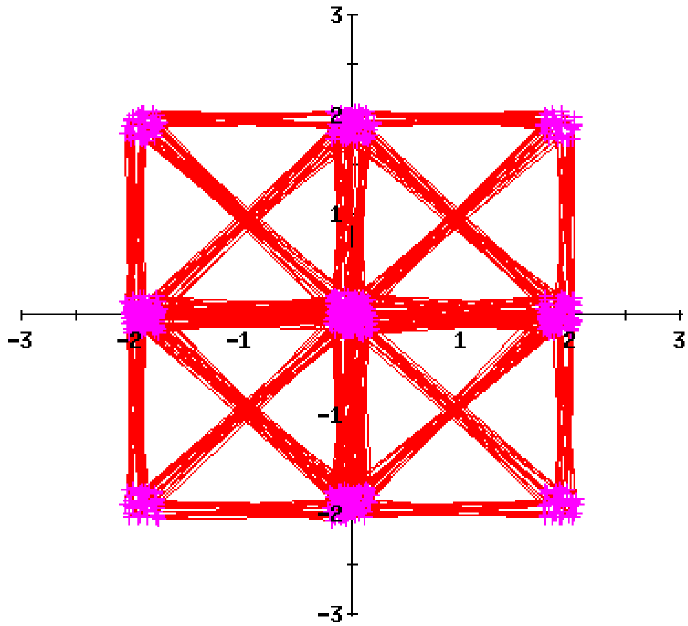


Figure 21 – a 3x3 constellation of a duobinary signal

Both class 1 and 4 signals produce a three level eye diagram and have a 9x9 constellation diagram.

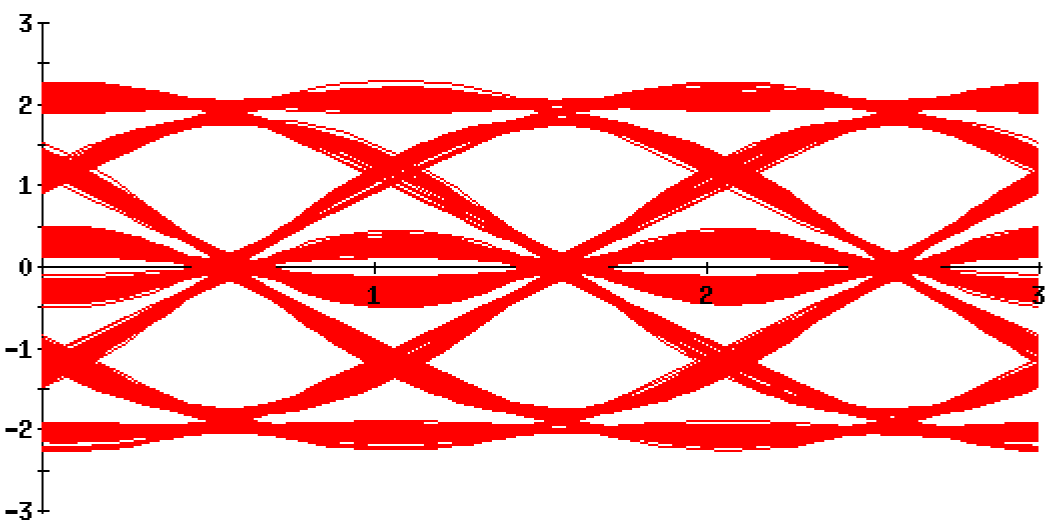


Figure 22 – The eye diagram of a duobinary signal or Class I

When class 2 partial response is input with a binary signal, it produces a five level signal. Its eye diagram and constellation diagram is shown below. It s constellation diagram is of size 5x5 as shown below.

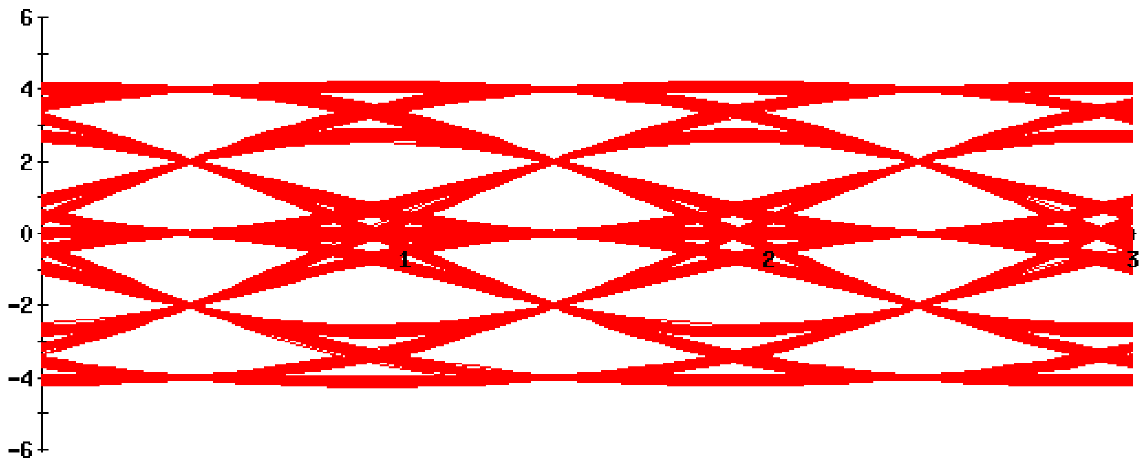


Figure 23 – The eye diagram of a Class 2 partial response signal

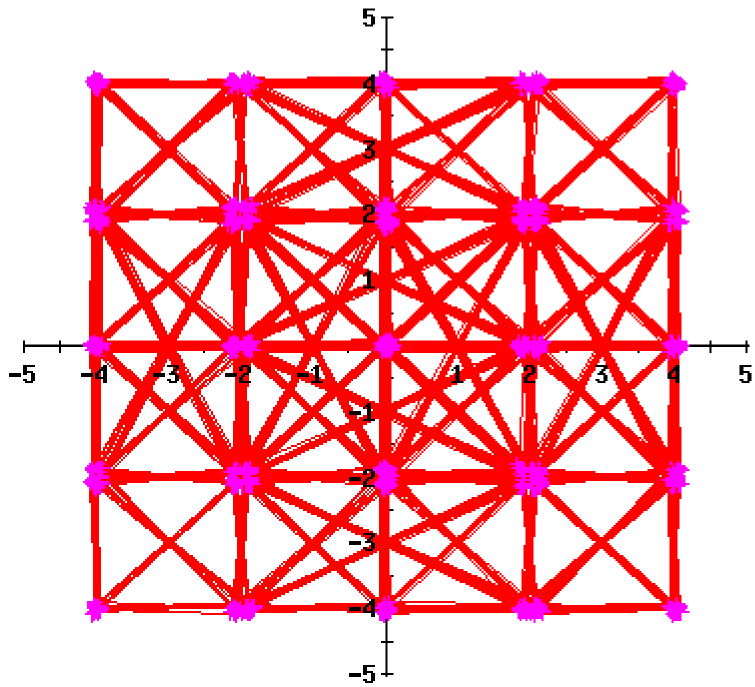


Figure 24- Five level response of the Class 2 signal

We recognize that the error potential of a signal is a function of how close the constellation points are. The closer, denser the points, the smaller the decision boundaries and easier it is for receiver to make mistakes in the presence of noise. Reducing the number of constellations and still be able to send the same symbol rate is clearly advantageous from an error reduction point of view.

QPSK error rate when used with a raised cosine pulse, is near ideal and is given in terms of its Eb/N0 as

$$P(e) = .5\,erfc\left[\left(\frac{E_b}{N_0}\right)^{\frac{1}{2}}\right]$$

Obviously we would not propose the use if duobinary to replace QPSK but if better bit efficiency is required then, where 16QAM is being considered, a QPR modulation makes a good alternate candidate.

For L-ary partial response the symbol error rate is given by

$$P(e)_{M-QPR} = 2\left(1-\frac{1}{L^2}\right)erfc\left[\frac{\pi}{4}(\log_2 L)^{1/2}\left(\frac{6}{L^2-1}\frac{E_b}{N_0}\right)^{\frac{1}{2}}\right]$$

Here L is equal to 3 for 9 point QPR signal or 5 for a 25 point signal.

Compare this to M-QAM Bit error rate probabilities, where M is equal to 2 for QPSK, 4 for 16QAM.

$$P(e)_{M-QAM} = 2\left(1-\frac{1}{M}\right)erfc\left[\left(\frac{6\,Log_2 L}{M^2-1}\frac{E_b}{N_0}\right)^{\frac{1}{2}}\right]$$

For gray-coded symbol, divide both of these by Log2 L or Log2 M, as appropriate to compute bit error rate (BER).

Plotting BER for QPSK, 9QPR and 16QAM, we see that 9QPR has app. 2.0 dB advantages over 16QAM. And since this is a constant envelope modulation, it suffers less from nonlinear amplifier characteristics.

QPR has been used as modulation on military microwave links and remains a viable alternative to 16QAM for satellite and other wireless links due to its high bit efficiency.

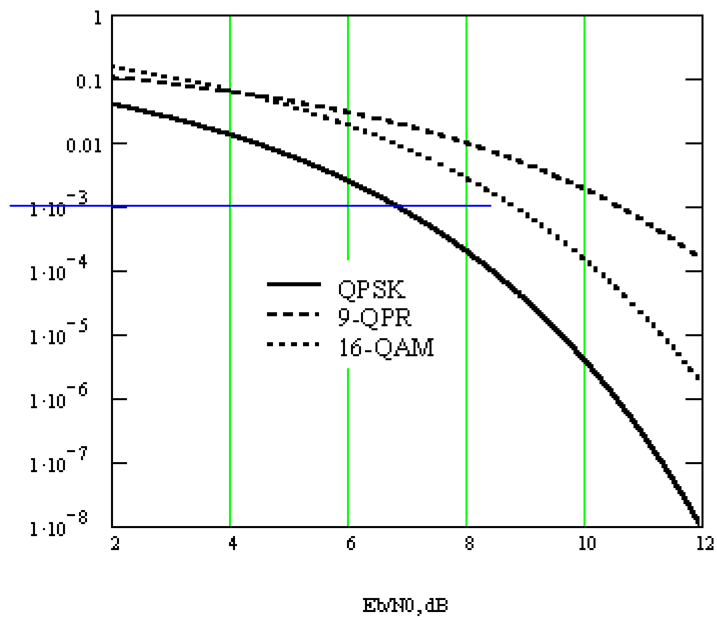


Fig. 25 – Bit Error rate of QPSK, 9QPR and 16QAM

Table II – Required Eb/N0 to obtained a BER of .001 for the uncoded channel

	BER = .001 EnN0	Bit Rate	Operating Backoff	Total
QPSK	6.80	1.6	0	6.80
9QPE	8.50	2	0	8.50
16QAM	10.50	3	2	12.50

In summary

- QPR is both a waveform shaping technique that allows transmission at the Nyquist limit and a modulation technique. In that is similar in concept to trellis coded modulation (TCM) which is also both a wave shaping mechanism as well as modulation.
- QPR pulse shaping is done by a cosine shaped filter which is easily built due to its graceful shape.
- In time domain, QPR signal is basically a sum of various combinations of sinc functions.
- QPR is a constant envelope so is not affected adversely to non-linear amplification.

Charan Langton
mntcastle@earthlink.net

References:

1. Digital Transmission Systems, David R. Smith, Ist Edition, 1985 Von Nostrand Reinhold Company