Digital Transmission

July 7, 2009

This tutorial is in support of module "EE401B: Digital Transmission".

On completing this module and tutorials students should be able to:

- Describe a range of baseband and passband digital modulation techniques including baseband PAM, passband PAM, PSK, QPSK, QAM and Spread Spectrum modulation.
- Calculate the bandwidth and probability of error for a subset of these techniques.
- Explain the advantages and disadvantages of using different modulation techniques with regard to signal bandwidth and noise performance.
- Explain the operation and advantages of combining error coding with modulation using soft decision decoding. Use the Viterbi algorithm to decode a sequence from a simple convolution code.
- Describe a range of modern advanced modulation techniques, and typically where they are used and what advantages they give. This should include GMSK, DMT and COFDM.

Students are expected to complete all assessments associated with this tutorial, and the marks will be counted formally towards their final mark for the module.

Modulation Basics

This set of tutorial pages is intended to introduce and justify some of the basics behind modulation for transmission of signals. It is the first section for my Digital Transmission Course and should take 1-2 weeks to complete

Telephone	4 kHz (300--3400Hz)
Radio - monaural, AM	8 kHz (160Hz--8kHz)
Radio - stereo, FM	15 kHz (30Hz--15kHz)
Telegraphy (CCITT-2)	120 Hz
Facsimile (FAX)	1.1 kHz
Television	6.5 MHz
High Definition TV	30 MHz

Assessment

Some very simple preliminary questions on material that you are expected to know before starting the course.

Deadline: 2009-10-18 00:59:00
No Questions: 3
Time Allowed: 10 min

Bandwidth Requirements Of Common Signals

Given below is a table of the bandwidth of some common signals.

Audio

Two factors determine the frequency range required for the transmission of speech by telephony (i) intelligibility as a function of frequency and (ii) energy as a function of frequency. Tests to determine the recognisability of syllables as the frequency range is reduced have revealed that the frequency range 1.5-2.5 kHz is most important. However most of the energy (80%) in typical speech lies in frequencies <1kHz. By international agreement the range adopted for telephony is 300 Hz - 3.4 kHz which embraces >60% of the energy and a high articulation efficiency. To make allowance for multiplexing

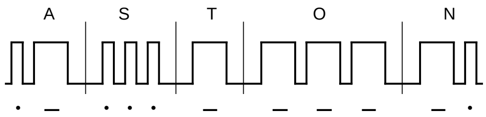

Figure 1: Morse Code

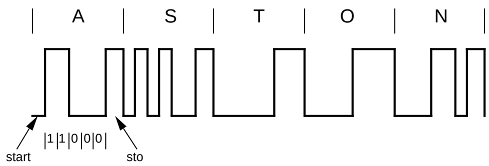

Figure 2: CCITT-2

and filtering a bandwidth of 4 kHz is allocated. For entertainment purposes, as on monaural radio, the bandwidth must be increased to 8 kHz to be acceptably natural.

Music

The range of the most acute human hearing extends from 10 Hz to 18 kHz. The range 160 Hz to 8 kHz has been taken as standard for monaural radio reproduction while stereophonic transmission uses 50 Hz to 15 kHz. A good home speaker system can reproduce sounds in the range 40 Hz - 20 kHz. (Purists argue that subtlety of tone or colour requires the inclusion of harmonics beyond the human audible range).

Telegraphy

The earliest form of digital communication was telegraphy dating from 1837. The sending operator tapped out *Morse code* with a key and the receiving operator listened to (or looked at) the incoming pattern of pulses and translated them into the message.

Although Morse code is still used for amateur short-wave transmissions it is not suitable for machine telegraphy where conversion from message (letters) to signal (pulses) and back again is carried out automatically by teleprinters. For machine telegraphy and telex codes such as CCITT-2 are used. This uses 5 equal length bits to represent a character plus a start sequence of 1 bit and an end mark of 1.5 bit lengths.

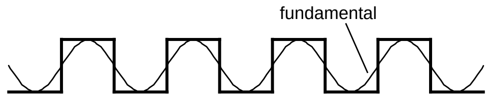

Figure 3: The relationship between the fundamental frequency of a signal and its bit rate

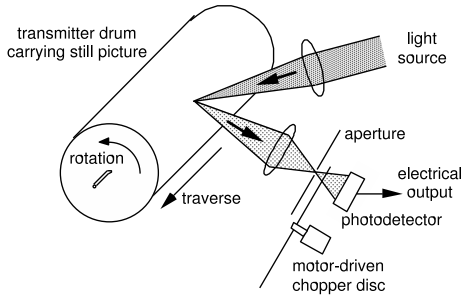

Figure 4: The analogue Fax scanner

The bit length in this code is 22 ms. The transmission speed is measured as the reciprocal of the time interval of the shortest pulse (one bit length). For CCITT-2 this is 1/22ms = 45 bauds. To determine the bandwidth that must be allocated we determine the highest number of transitions per second which can be made i.e. when the signal is 0101010 (see figure 3). The sinusoidal signal with the same period as this pulsed pattern is the fundamental frequency. Show for yourself that this is 22.5Hz for CCITT-2.

We shall find that modulation requires that we double this to 45 Hz. If the signal is multiplexed with others guard bands are necessary between the channels to allow for filtering and the bandwidth allocated per channel is 120Hz.

Fax

This system transmits pictorial information over standard telephone lines.

In the analogue fax scanner the graphic is wrapped around a cylindrical drum which rotates at a constant speed and simultaneously traverses axially on a lead screw. A light spot focused on the surface traces a spiral path around the drum. The reflected light intensity is modulated using a chopper disk to provide modulation and measured by a

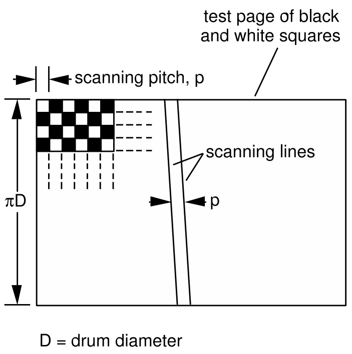

Figure 5: The Fax Raster Scan

photodetector. The fax receiver is similar in that the received signal is varies the intensity of a light source focused onto photographic paper wrapped round an identical drum to that in the receiver.

By international agreement the speed of rotation of drum is 1 rev/sec and a parameter called the index of co-operation is defined by

index of co-operation = drum diameter (D)/scanning pitch (p) =352.

By adherence to a common index drums of different diameters may be used in transmitter and receiver without distorting the transmitted image. In order to obtain equal resolution in the horizontal and vertical dimensions the system must be able to reproduce a test pattern of black and white squares where the side length of each square is equal to the scanning pitch p (The horizontal scan steps by p at the end of each vertical scan).

Hence no of squares scanned per sec $= \pi D/p = 352\pi$ and time to scan 1 black and one white square $= 2/352\pi$ sec giving a fundamental frequency of $353\pi/2 = 553$ Hz.

As with telegraphy we double this figure to arrive at the bandwidth required to take account of modulation to be 1106 Hz

Television

We can estimate the bandwidth required for television transmission in exactly the same way as for fax. In the case of television the graphic is scanned by electron beam. In order to give an impression of continuous movement without flicker the graphics must be scanned at a rate of 25 graphics/second.

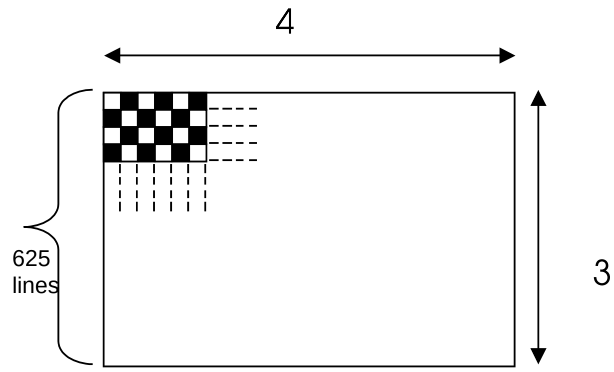

Figure 6: The Television Raster Scan

The British PAL television system (show in figure 6)) uses 625 lines and the graphic aspect ratio = width/height = 4/3; hence one graphic has (625)2 x 4/3 squares and the number of pairs of squares scanned per sec (fundamental frequency) $= 652^2 \times 4/3 \times 25/2 = 6.51 \times 10^6$Hz

Video signals are modulated using a principle different from that for telegraphy or fax and 6.51 MHz is the bandwidth i.e. we do not double the fundamental frequency. Despite many subtleties this estimate for the bandwidth is more or less correct in practice.

What Is Modulation?

Modulation is the variation of the characteristics of one signal (the carrier) in proportion to the amplitude of a second, information-carrying signal, as illustrated in figure 7). Modulation results in the transfer of the signal information to higher or lower frequency. If a sinusoidal carrier is used (as is usually the case) then either the amplitude, frequency or phase of the carrier signal may be modulated giving rise to amplitude modulation (AM), frequency modulation (FM) or phase modulation (PM) respectively.

Digital Modulation

In the case of digital modulation there are a number of steps which must be taking in encoding and decoding the information for transmission.

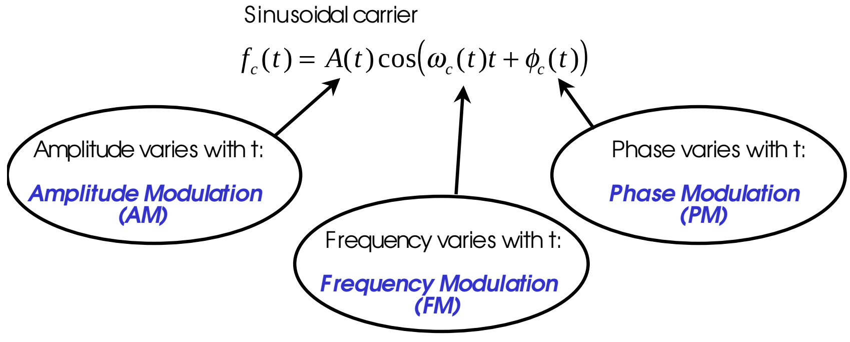

Figure 7: Modulation

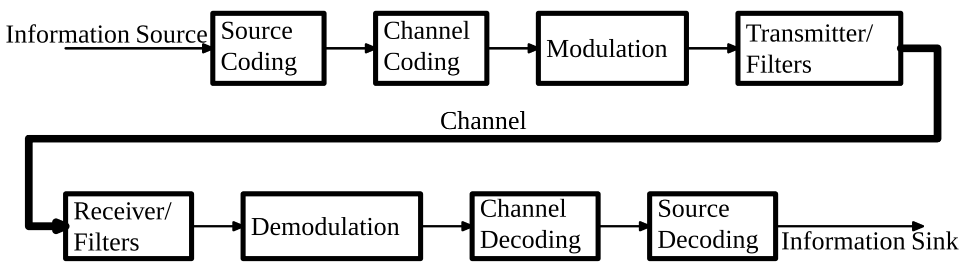

Figure 8: A Digital Transmission System

Starting from an information source the first step is called *source coding* . This process maps the information source efficiently into a discrete digital format. If the original source were analogue then source coding corresponds to efficiently representing the signal in a digital format. If the original source is digital then signal compression may be used to increase efficiency. The result is discrete digital data

The next process, before modulation is *channel coding* which maps the discrete data onto format suitable for the characteristics of the transmission system. For example if the transmission introduces errors in the data error control coding, optimised for the specific transmission system, will be required.

Finally the digital modulation maps the discrete digital signal onto an analogue signal (*quantisation*)for the transmission medium, as all transmission media (excepting quantum based communications which are beyond the scope of this tutorial) are analogue in nature.

Why Modulate?

The principle reasons why we need to modulate signals are as follows:

To make efficient use of the lines (multiplexing)
As already noted a telephone channel is quite intelligible within the 300--3400 Hz audio range. Typically coaxial cable has about 3MHz bandwidth, and so can accommodates many channels. We therefore need to shift the frequencies of some of the channels so we can pack them together (multiplex) and make full use of the available bandwidth

To make radio communications feasible
Radio antennas are only efficient when approximately the same size as the period of the radio frequency. A 1kHz radio signal has ,a period of 300km which is clearly impractical. We therefore need to shift the transmission to higher frequencies with shorter period more practical for transmission. The second advantage of doing this is that

we are converting a “broadband” signal into a “narrowband” signal e.g. baseband audio has a bandwidth to frequency ratio of 3400/300 = 11, but modulating that to 10 MHz gives 1003400/1000300 = 1.01. Again we can exploit this extra bandwidth for multiplexing many radio channels.

To simplify signal processing It is much simpler to design electronic systems for narrow frequency bands. Using the principle described above we can use modulation to convert a broadband signal to a narrowband signal. Some component types are optimal in a fixed frequency range. Modulation allows us to shift our signal to that range, and example of this is that high gain, high selectivity IF (intermediate frequency) filters are only feasible at certain frequencies.

Types Of Signals And Systems

We need mathematical tools to develop insight into the transmission of signals in a communications system. Depending on the types of signals and which feature we are interested in we need to distinguish different classes of signals

Periodic and aperiodic signals

A signal, $g(t)$ is *periodic* if it satisfies the condition

$$g(t) = g(t + T_0)$$

for all t where t is time and T_0 a constant. T_0 is the period of the signal, the duration of one complete cycle of $g(t)$.

Any signal for which this condition doesn't hold, for any value of T_0 is *aperiodic* .

Deterministic and Random Signals

A *deterministic* signal is a signal for which there is no uncertainty with respect to its value at any time. It may modelled as a completely specified function of time. With a *random* signal there is an uncertainty in a signal before it occurs, and we can model it only as having a probability of a particular value as a function of time

Energy and Power Signals

In electrical systems a signal is represented by a voltage or current. If we consider a voltage $v(t)$ across a resistor R producing current $i(t)$ then we may write the instantaneous power dissipated in the resistor as

$$p(t) = |v(t)|^2/R = |i(t)|^2 R$$

i.e. the power dissipated is proportional to the signal amplitude (current or voltage) squared. In analysing signals we normalise the calculations by assuming a 1-ohm resistor to we may express the instantaneous power for a signal as

$$p(t) = |g(t)|^2$$

Based on this we then have the total energy of a signal g(t) defined by

$$E = \int_{-\infty}^{\infty} |g(t)|^2 \, dt$$

Correspondingly, the average power of of a signal $g(t)$ defined by

$$P = \lim_{t \to \infty} \frac{1}{2T} \int_{-T}^{T} |g(t)|^2 \, dt$$

For an *energy signal* the total energy of the signal satisfies the condition

$$0 < E < \infty$$

For a *power signal* the total energy of the signal satisfies

$$0 < P < \infty$$

These are mutually exclusive conditions. An energy signal has zero average power and a power signal has infinite energy. Usually periodic signals and random signals are power signals, and signals that are deterministic and aperiodic are energy signals.

Representation of Signals

There are many possible ways to represent signals however the most useful, in the context of communications is to represent them as a sum of sinusoidal components of different frequencies and phases. The reasons for this are due to the fact that the response of a system to a sinusoidal input is another sinusoid of the same frequency but with a different amplitude and phase provided:

1. The system is *linear* and obeys the *principle of superposition* . That is if a linear combination of input signals is applied to the system then the output of the system will be a linear sum of the individual outputs that would be obtained if the signals were applied individually;

2. The system is *time invariant* . That is the response of a system to an input that is time delayed is simply time delayed.

This leads to the use of Fourier methods for the analysis of signals.

Despite seeing a forest of integration signs when dealing with Fourier methods there is in fact very little integration done, and that which is is at A-level. A familiarity with the theorems associated with Fourier Analysis (See Fourier Analysis (Section)) together with some basic integration results will solve most common problems. In practice Fourier Analysis is often done digitally with a minimum of mathematics to, for example, extract information from noisy signals, design electrical filters, and clean TV graphics.

If a signal is periodic then *Fourier series* are used to represent the signal as a set of harmonically related sine waves. If the signal however is an energy signal the *Fourier transform* is used to represent the signal as a continuous range of sinusoids. Applying these techniques we obtain the *frequency-domain* representation or *spectrum* of the signal which we can then use to simplify our analysis of linear systems such as communication systems.

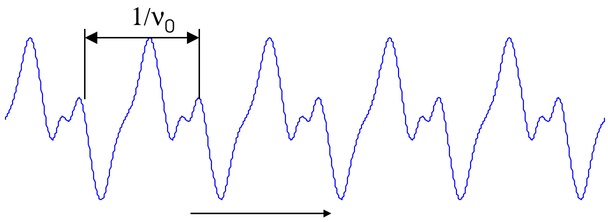

Figure 9: A periodic Waveform

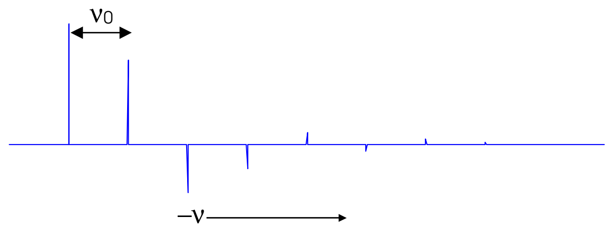

Figure 10: The spectrum of a periodic Waveform

Fourier Synthesis

If a signal waveform is periodic, as shown in figure 9), then it contains harmonics (multiples) of the fundamental frequency with various amplitudes and frequencies.

The waveform can be analysed to find the amplitudes of these harmonics, and a list can be made of their various amplitudes and phases. Alternatively we can plot a graph (the spectrum) of the amplitudes versus frequency, as shown in figure 10).

Mathematically we can express the signal as a sum of these individual frequencies. sines and cosines are needed to allow for the harmonics all having different phases.

$$
\begin{aligned}
F(t) &= a_0 + a_1 \cos 2\pi\nu_0 t + b_1 \sin 2\pi\nu_0 t \\
&\quad + a_2 \cos 4\pi\nu_0 t + b_2 \sin 4\pi\nu_0 t + \cdots \\
&= \sum_{n=-\infty}^{\infty} a_n \cos(2\pi n\nu_0 t) + b_n \sin(2\pi n\nu_0 t) \\
&= \frac{A_0}{2} + \sum_{n=1}^{\infty} (A_n \cos(2\pi n\nu_0 t) + B_n \sin(2\pi n\nu_0 t))
\end{aligned}
$$

In this last form we have use the property $\cos(x) = \cos(-x)$ and $\sin(x) = -\sin(-x)$ to change the sum to be between positive integers: $A_n = a_{-n} + a_n$ and $B_n = b_{-n} - b_n$ and A_0 is divided by 2 so we don't count it twice.

The process of constructing a waveform by adding together a fundamental frequency and its harmonics of various amplitudes is called *Fourier Synthesis* .

Fourier Analysis

Given a signal, we often want to extract from it the various frequencies and amplitudes that are present. This is called *Fourier Analysis* . To find these amplitudes we exploit the *orthogonality* property of sines and cosines - that is if you take a sine and a cosine, or two sines, or two cosines that are multiples of a fundamental frequency, and integrate them over one cycle of that fundamental period then the result is always zero unless they are the same. Mathematically, if $P = 1/\nu_0$ is one period

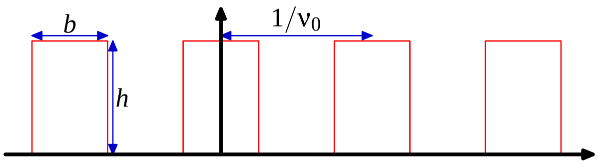

Figure 11: A rectangular waveform

$$\begin{aligned} \begin{matrix} \int_{t=0}^{P} \sin(2\pi n\nu_0 t)\sin(2\pi m\nu_0 t)\,dt \\ \int_{t=0}^{P} \cos(2\pi n\nu_0 t)\cos(2\pi m\nu_0 t)\,dt \end{matrix} &= \begin{cases} 0, \; n \neq m \\ \frac{1}{2\nu_0}, \; n = m \end{cases} \\ \int_{t=0}^{P} \sin(2\pi n\nu_0 t)\cos(2\pi m\nu_0 t)\,dt &= 0, \text{ always} \end{aligned}$$

Now if we take our signal function $F(t)$, multiply it by $\sin(2\pi m\nu_0 t)$ and integrate over a single fundamental period we get

$$\begin{aligned} \int_0^P F(t)\sin(2\pi m\nu_0 t)\,dt &= \int_0^P \left(\frac{A_0}{2} + \sum_{n=1}^{\infty} \left(\begin{matrix} A_n \cos\cos(2\pi n\nu_0 t) \\ + B_n \sin(2\pi n\nu_0 t) \end{matrix} \right) \right) \times \sin(2\pi m\nu_0 t)\,dt \\ &= \int_0^P B_n \sin^2(2\pi n\nu_0 t)\,dt \\ &= \frac{B_m P}{2} \end{aligned}$$

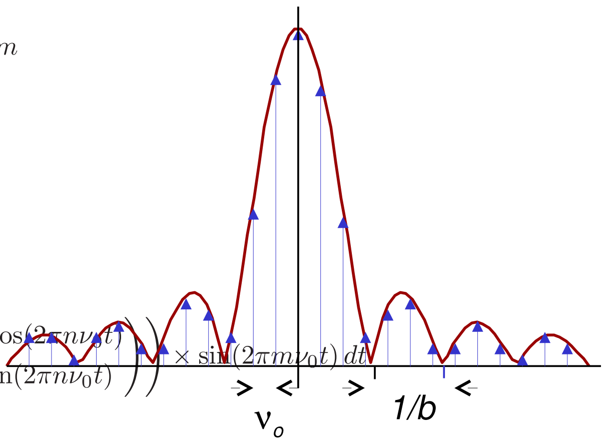

Figure 12: Spectrum of a rectangular waveform

Similarly for $\cos(2\pi m\nu_0 t)$. That is all the terms in the Fourier expansion vanish except that at the frequency we multiplied it by. Therefore we can determine the coefficients of the Fourier expansion as follows

$$\begin{aligned} B_m &= \frac{2}{P}\int_0^P F(t)\sin(2\pi m\nu_0 t)\,dt \\ A_m &= \frac{2}{P}\int_0^P F(t)\cos(2\pi m\nu_0 t)\,dt \end{aligned}$$

Fourier Analysis of a periodic rectangular waveform We shall now apply the principles of Fourier Analysis to one of the simplest waveforms - a rectangular Waveform of period

$$1/\nu_0$$

, pulse width b and pulse height h.

Since the function is zero outside the pulse width, and has an amplitude h we can write the fourier coefficient integral as

$$\begin{aligned} A_m &= 2\nu_0 \int_{-1/2\nu_0}^{1/2\nu_0} F(t)\cos(2\pi m\nu_0 t)dt \\ &= 2h\nu_0 \int_{-b/2}^{b/2} \cos(2\pi m\nu_0 t)dt \end{aligned}$$

Solving this integral gives the coefficients for the cosine terms as

$$\begin{aligned} A_m &= \frac{2h\nu_0}{2\pi m\nu_0}\left[\sin(\pi m\nu_0 b) - \sin(-\pi m\nu_0 b)\right] \\ &= 2h\nu_0 b \frac{\sin(\pi m\nu_0 b)}{\pi m\nu_0 b} \end{aligned}$$

Since the function is even the sinusoidal coefficients will be zero. The power spectrum is therefore a set of delta functions spaced at ν_0 with a sinc envelope with a width of $1/b$.

We can therefore rewrite this signal in terms of the Fourier series below

$$F(t) = h\nu_0 b + 2h\nu_0 b \sum_{m=1}^{\infty} \left[\frac{\sin(\pi m \nu_0 b)}{\pi m \nu_0 b} \right] \cos(2\pi m \nu_0 t)$$

This spectrum clearly demonstrates a number of important properties for Fourier transforms: the higher the frequency of the signal, the closer together the spectral peaks; and the wider the waveform (pulse) the narrower the spectrum. See Fourier Transform Properties (Section)

Complex Notation Usually when performing Fourier analysis it is more convenient to use complex exponentials to manipulate the formulae than sines and cosines. Using the expression $e^{i\theta} = \cos\theta + i\sin\theta$ to change from sines to cosines to complex exponentials we get the following for the Fourier series

$$\begin{aligned} F(t) =& \frac{A_0}{2} + \sum_{n=1}^{\infty} e^{2\pi i n \nu_0 t}(A_n - iB_n) \\ =& \frac{A_0}{2} + \sum_{n=1}^{\infty} e^{2\pi i n \nu_0 t} C_n \end{aligned}$$

Now we have a complex amplitude C_n representing both the amplitude R_n and phase ϕ_n of the signal.

$$\begin{aligned} C_n &= A_n - iB_n = R_n e^{i\phi_n} \\ R_n &= |C_n|^2 = \sqrt{A_n^2 + B_n^2} \\ \tan\phi_n &= -\frac{B_n}{A_n} \end{aligned}$$

The inversion formula becomes

$$\begin{aligned} C_n =& 2\nu_0 \int_0^{1/\nu_0} F(t) e^{i2\pi\nu_0 t} dt \\ =& \frac{\omega_0}{\pi} \int_0^{2\pi/\omega_0} F(t) e^{i\omega_0 t} dt \end{aligned}$$

Fourier Analysis of non-periodic functions So far we have looked at periodic functions which can be expressed as a sum of harmonics of the fundamental frequency. If we increase the period of the function then the fundamental frequency and thus

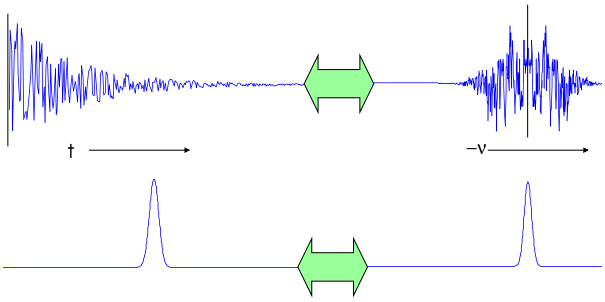

Figure 13: An illustration of the spectra corresponding to non-periodic signals - an abrupt clash like signal and a smooth pulse

the spacing between the harmonics will decrease. A non-periodic function may be treated as the limiting case where the period goes to infinity and the spacing between the Fourier terms goes to zero. Instead of having a spectrum of discrete harmonics the spectrum is a continuous function. Shown below are examples of two non-periodic functions and their spectra - an abrupt clash type signal and a smooth pulse.

Mathematically the summation sign used when expressing periodic signals as a sum of harmonics is changed to an integral and we can write the function in terms of its spectrum as

$$g(t) = \int_{-\infty}^{\infty} G(f) e^{i2\pi f t}\, df$$

As with the Fourier series we can perform the inverse process to determine the spectrum from the signal itself.

$$G(f) = \int_{-\infty}^{\infty} g(t) e^{-i2\pi f t}\, dt$$

These functions $g(t)$ and $G(f)$ are called a *Fourier Pair* .

Fourier Transform If we have a function $g(t)$ then we perform the *Fourier transform* to calculate its frequency spectrum $G(f)$.

$$G(f) = \int_{-\infty}^{\infty} g(t) e^{-i2\pi f t}\, dt \qquad (1)$$

We perform the *Inverse Fourier Transform* to calculate a function from its frequency spectrum

$$g(t) = \int_{-\infty}^{\infty} G(f) e^{i2\pi ft}\, df \tag{2}$$

The Fourier transform provides us with the link between the time domain and frequency (or spectral) domain views of a signal. A signal is uniquely defined by either representation.

If we calculate the energy of a signal the the result must be the same in either representation

$$E = \int_{-\infty}^{\infty} |g(t)|^2\, dt = \int_{-\infty}^{\infty} |G(f)|^2\, df \tag{3}$$

The amplitude spectrum of a signal is defined by the modulus of the complex spectrum $|G(f)|$

The energy spectral density is given by $E_g = |G(f)|^2$ in units of joules per hertz.

Dirichlet's Conditions Not all functions are Fourier transformable. Dirichlet's Conditions are the requirements on a function if it is to be transformable. The Fourier Transform of a function $g(t)$ exists if

1. The function $g(t)$ is single-valued, with a finite number of maxima and minima in any finite time interval.

2. The function $g(t)$ has a finite number of discontinuities in any finite time interval (piece-wise continuous).

3. The function $g(t)$ is absolutely integrable, that is $\int_{-\infty}^{\infty} |g(t)|\, dt < \infty$

In practice the Fourier transform exists for all signals which are physically realisable. In particular condition 3 above holds for all energy signals. In nature, these conditions seem to hold for all phenomena which can be described mathematically.

Fourier Transform Properties There are several theorems which are of great use in manipulating Fourier-pairs, and they should be memorised. The art of practical Fourier transforming is in the manipulation of functions using these theorems rather than in doing extensive and tedious elementary integrations. It is this, as much as anything, which makes Fourier theory such a powerful tool for the practical working scientist. Proof of most of these properties is fairly trivial.

In what follows we assume the following Fourier-pairs, $g_1(t) \Leftrightarrow G_1(f)$ and $g_2(t) \Leftrightarrow G_2(f)$.

Addition Theorem If we add two functions we add their Fourier Transforms

$$c_1 g_1(t) + c_2 g_2(t) \Leftrightarrow c_1 G_1(f) + c_2 G_2(f)$$

Time Scaling Compressing a signal in time causes a proportional increase in spectral width. The spectral magnitude must decrease proportionally as the two signal representations must have the same energy

$$g(at) \Leftrightarrow \frac{1}{|a|} G\left(\frac{f}{a}\right)$$

Duality If we know the Fourier transform of a function then we also know the Fourier transform of the Fourier transform

$$\text{If } g(t) \Leftrightarrow G(f) \text{ then } G(t) \Leftrightarrow g(-f)$$

Time shift and Frequency Shift States how the Fourier transform of a function changes when we shift it in time or frequency

$$\text{If } g(t) \Leftrightarrow G(f) \text{ then } g(t - t_0) \Leftrightarrow G(f) e^{-2\pi i f t_0}$$

$$\text{If } g(t) \Leftrightarrow G(f) \text{ then } g(t) e^{-2\pi i f_0 t} \Leftrightarrow G(f - f_0) \tag{4}$$

There follows a number of important examples of problems where these sets of properties are used to make the calculations simple.

Fourier Transform Of A Rectangular Pulse Shown in figure 14) is a rectangular pulse of amplitude A and width T. Mathematically such a pulse is written as $A\mathrm{rect}\frac{t}{T}$

The Fourier transform of this function is then given by

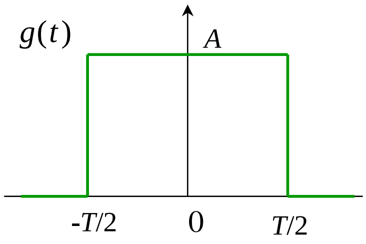

Figure 14: A Rectangular Pulse

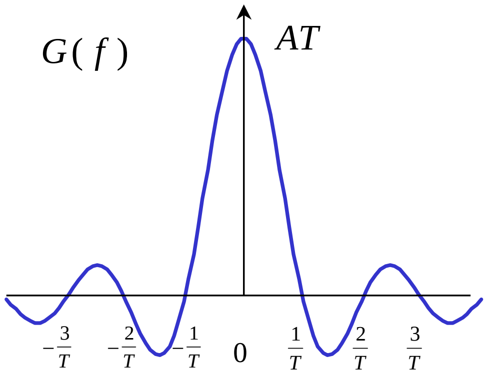

Figure 15: Fourier Transform of A Rectangular Pulse

$$
\begin{aligned}
G(f) &= \int_{-\infty}^{\infty} A\text{rect}\left(\frac{t}{T}\right) e^{-2\pi i f t}\, dt \\
&= A \int_{-\frac{T}{2}}^{\frac{T}{2}} e^{-2\pi i f t}\, dt
\end{aligned}
$$

where we have put the interval to be between $-T/2$ and $T/2$ where the function is non-zero. Solving the integral and putting in the intervals gives

$$
\begin{aligned}
G(f) &= A \left[\frac{e^{-2\pi i f t}}{-2\pi i f} \right]_{-\frac{T}{2}}^{\frac{T}{2}} \\
&= A \frac{e^{-\pi i f T} - e^{\pi i f T}}{-2\pi i f} \\
&= AT \left(\frac{\sin(\pi f T)}{\pi f T} \right) \\
&= AT \text{sinc}(fT)
\end{aligned}
$$

This is plotted in figure 15). We can therefore say that these two functions are a Fourier transform pair.

$$
A\text{rect}\left(\frac{t}{T}\right) \Leftrightarrow AT\text{sinc}(fT)
$$

If we look at this Fourier transform pair of functions we can immediately see an important general relationship between pulse width in the time domain and bandwidth in the spectral domain - notably they are reciprocals. That is the pulse width is given by T while the width of the main peak in the sinc spectral function is proportional to $1/T$. Therefore as the pulse width broadens in time the spectral width proportionally decreases. This is generally true, no matter what the pulse shape and corresponding spectrum are.

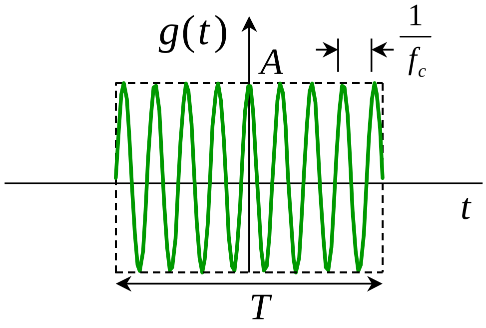

Figure 16: A Pulsed Carrier

Assessment
Deadline: 2009-10-18 00:59:00
No Questions: 3
Time Allowed: 15 min

Fourier Transform of a Radio-Pulse A modulated pulse is represented mathematically by

$$
g(t) = A\text{rect}\left(\frac{t}{T}\right) \cos(2\pi f_c t)
$$

shown in figure 16). Remember that the cosine function can be written as

$$
\cos(2\pi f_c t) = \frac{1}{2}\left(e^{2\pi i f_c t} + e^{-2\pi i f_c t}\right)
$$

So our function becomes

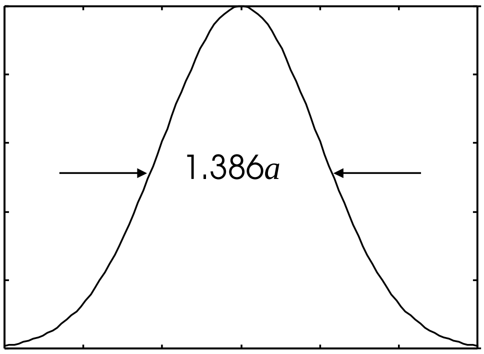

Figure 18: A Gaussian

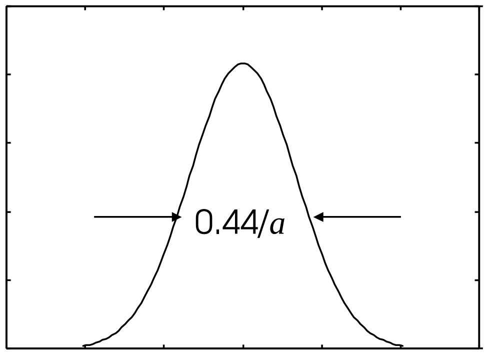

Figure 19: Fourier Transform of a Gaussian

$$g(t) = \frac{1}{2} A\text{rect}\left(\frac{t}{T}\right)\left(e^{2\pi i f_c t} + e^{-2\pi i f_c t}\right)$$

The addition theorem allows us to split this into two Fourier transforms, and the Frequency shift Theorem 4) tells that the Fourier transform for each part is simply that of the rectangular pulse shifted by $+f_c$ or $-f_c$ so we have

$$G(f) = \frac{AT}{2}\left[\text{sinc}\left(T(f-f_c)\right) + \text{sinc}\left(T(f+f_c)\right)\right]$$

as plotted in figure 17)

Fourier Transform of a Gaussian We now turn ourselves to finding the Fourier transform of a Gaussian function (figure 18))

$$G(t) = e^{-t^2/a^2}$$

This involves what is probably the most difficult integral we will need to do as part of the digital modulation module.

a is the width parameter of the function and the full width at half maximum (FWHM), commonly used when referring to pulses is $1.386a$.

The Fourier Transform integral of this function is

$$g(f) = \int_{-\infty}^{\infty} e^{-t^2/a^2} e^{2\pi f t} dt$$

We need to do a bit of rearranging of this integral to make it solvable.

Firstly we rearrange the exponent into the form $-(t/a - \pi i f a)^2 - \pi^2 f^2 a^2$

We then change variables, defining$z = (t/a - \pi i f a)$) **so**
Definition]
that $dt = adz$

The integral then becomes

$$\begin{aligned} g(f) =& a e^{-\pi^2 f^2 a^2} \int_{-\infty}^{\infty} e^{-\pi^2} dz \\ =& a\sqrt{\pi} e^{-\pi^2 f^2 a^2} \end{aligned}$$

where we have used the well known result $\int_{-\infty}^{\infty} e^{-\pi^2/a^2} dz = a\sqrt{\pi}$

Thus the Fourier transform of a Gaussian of FWHM $1.386a$ is also a Gaussian of width $0.44/a$ as shown in figure 19).

Properties of the Delta function The properties of the *Dirac delta-function* are very useful when working with Fourier analysis. This function is defined by the properties

$$\begin{aligned} \delta(x) = 0, &\quad x \neq 0 \\ \delta(x) = \infty, &\quad x = 0 \\ \int_{-\infty}^{\infty} \delta(x) dx &\quad = 1 \end{aligned}$$

Note that it disobeys Dirichlets conditions in that is is unbounded at $x = 0$, however we can still determine its Fourier transform. If we take this function as the limiting form of a Gaussian of unit area.

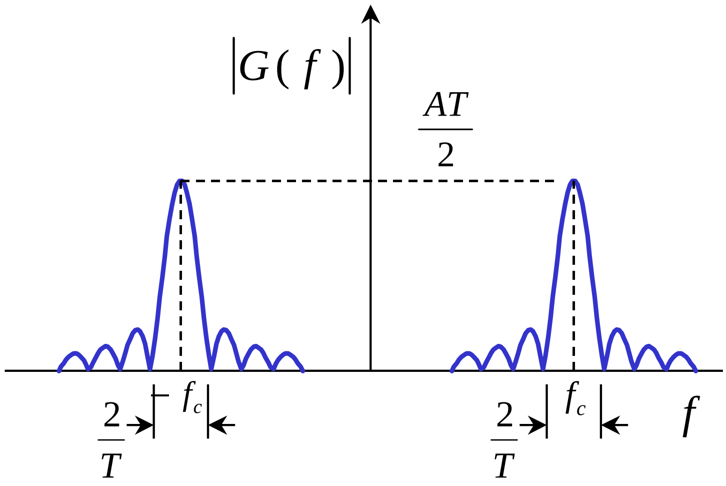

Figure 17: Spectrum of A Pulsed Carrier

$$g(t) = \lim_{\tau \to 0} \frac{1}{\tau} e^{-\frac{\pi t^2}{\tau^2}}$$

As T goes to zero we have a Gaussian of decreasing width and increasing height until it becomes infinitely narrow and infinitely large in amplitude yet still has unit area. The Fourier transform will then be given by

$$G(f) = \lim_{\tau \to 0} e^{-\pi^2 \tau^2 f^2}$$

Which tends to a Gaussian of infinite width and unit height. The Fourier transform of the delta function is therefore unity.

$$\delta(x) \Leftrightarrow 1$$

Similarly, if we have a constant signal then using the principle of duality for Fourier Transforms the spectrum must be a delta function.

The usefulness of the delta function is apparent when we consider the following property.

Remembering that $\delta(x-a) = 0$ unless $x = a$ we have

$$\int_{-\infty}^{\infty} F(x)\delta(x-a)dx = F(a)$$

i.e. the delta function picks out a single value in the integral. We thus get Fourier transform integral for the complex exponential function as

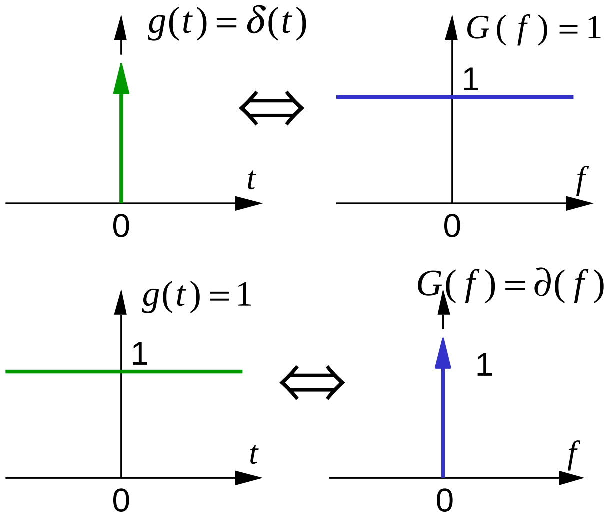

Figure 20: Fourier Transform of a Delta Function

$$\int_{-\infty}^{\infty} e^{i2\pi ft}\delta(x-f_0)dx = e^{i2\pi f_0 t}$$

Which gives us the the Fourier transform pair

$$e^{i2\pi f_0 t} \Leftrightarrow \delta(f-f_0)$$

This states that the complex exponential function $e^{i2\pi f_0 t}$ is transformed in the frequency domain into a delta function at f_0 It is just our old friend the frequency shift theorem.

Fourier Transforms Involving The Delta Function We can use the relationship derived previously, (see 4)) to calculate the Fourier transforms of sines and cosines by writing them in terms of their complex exponentials. We thus obtain the relationships shown in figure 21).

The Fourier transform of the cosine function is a pair of delta functions at pmf_0 and similarly for the sinusoidal function we have the result in figure 22).

Now we can see how periodic signals, previously described by discrete Fourier series can be represented using delta functions in Fourier Transforms.

Fourier Transforms of Periodic Signals We have already seen how a periodic signal can be represented in terms of the complex exponential Fourier series Also Fourier transforms can be defined for complex exponentials. Therefore it is not unreasonable to suppose that a periodic signal can be represented in terms of a Fourier transform containing delta functions (the Fourier transform of complex exponentials).

We can write a periodic function $g_{T_0}(t)$ in terms of a generating function $g(t)$ (the function representing one period only and zero elsewhere) as follows.

$$g_{T_0} = \sum_{m=-\infty}^{\infty} g(t-mT_0)$$

where T_0 is the period. This periodic function may be represented as a Fourier series

$$g_{T_0} = \sum_{n=-\infty}^{\infty} c_n e^{i2\pi f_n t}$$

where the complex coefficients are given by

$$c_n = \frac{1}{T_0}\int_{-T_0/2}^{T_0/2} g_{T_0}(t)e^{-i2n\pi f_0 t}dt$$

We can substitute the generating function in here and change the limits to infinity since the generating function only has a nonzero value in the limits $-T_0/2$ to $T_0/2$ The formula for the complex Fourier coefficients then becomes

$$c_n = f_0\int_{-\infty}^{\infty} g(t)e^{-i2n\pi f_0 t}dt$$

Now the generating function $g(T)$ is Fourier transformable, and so this integral corresponds to

$$c_n = f_0 G(nf_0)$$

were $G(nf_0)$ is the Fourier transform of $g(T)$ evaluated at frequency nf_0. i.e. the values of the complex coefficients in the Fourier series are the same as the values the Fourier transform of the generating function would have at the same frequency. Substituting this into our Fourier series expression we get the following expression for the periodic function in terms of its Fourier series:

$$\sum_{m=-\infty}^{\infty} g(t-mT_0) = f_0 \sum_{n=-\infty}^{\infty} G(nf_0)e^{i2\pi f_0 t}$$

and finally, using the result for the Fourier transform of a complex exponential we have

$$\sum_{m=-\infty}^{\infty} g(t-mT_0) \Leftrightarrow f_0 \sum_{n=-\infty}^{\infty} G(nf_0)\delta(f-nf_0)$$

or alternatively

$$G_{T_0}(f) = \sum_{n=-\infty}^{\infty} c_n\delta(f-nf_0)$$

This is a very important result, stated in words it means

> **Periodicity in the time domain has the effect of changing the frequency domain or spectrum of a signal into a discrete form defined at integer multiples of the fundamental frequency.**

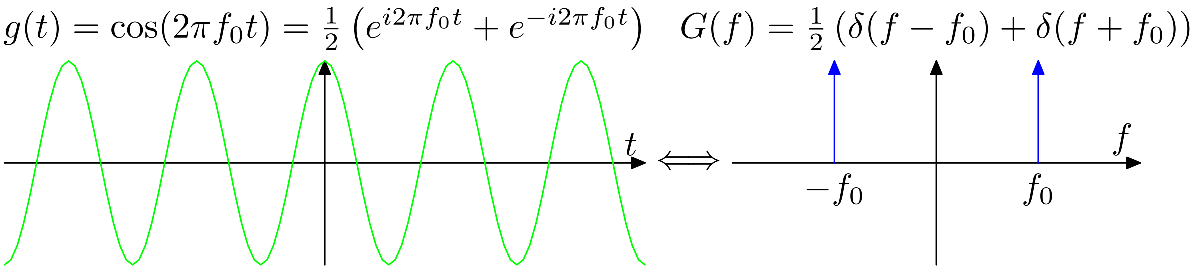

Figure 21: Fourier Transform of a Cosine

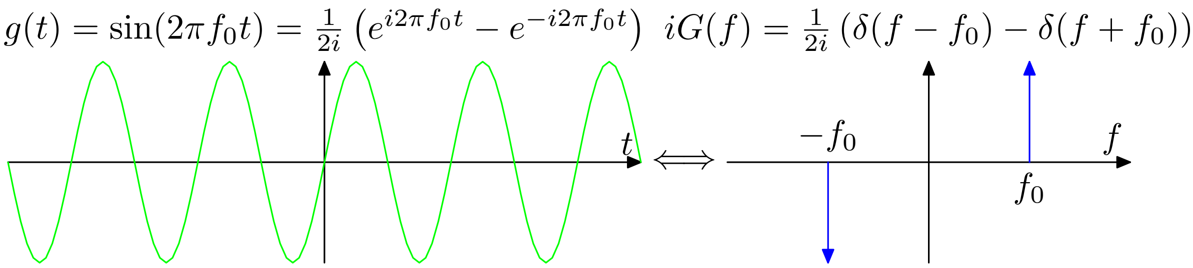

Figure 22: Fourier Transform of a Sine

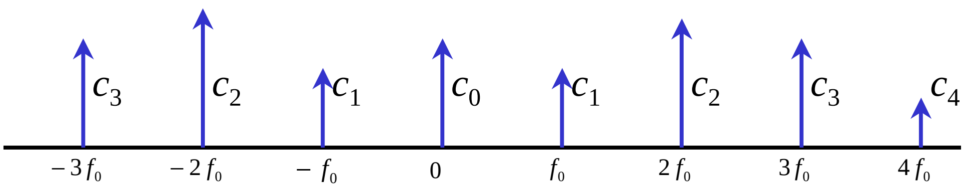

Figure 23: Fourier Transform of a Periodic Waveform

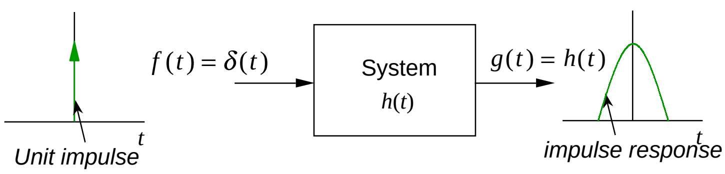

Figure 24: The Impulse Response

The impulse response of a system is defined as the response of the system (with zero initial conditions) to a unit impulse or delta function d(t) applied to the input of the system.

The Impulse Response And Convolution

This idea arises in many applications - for example the response of a lens to a point image (the point spread function - in 2 dimensions), the response of a spectrometer to a monochromatic source and of course here we have the electrical response of a system to a unit impulse. If the system is time-invariant the it is the same no matter when the impulse is applied to the system.

Now if we apply a general signal to the system, what is its output going to be?

We can approximate the input signal $f(t)$ by a sum of pulses of width Δt as shown in figure 25). As Δt tends to zero these functions will tend to impulse functions times the amplitude of the signal. The output from the k th pulse at time $k\Delta t$ will given by $f(k\Delta t)h(t-k\Delta t)\Delta t$ where $h(t)$ is the impulse response of the system i.e. the pulse will spread out by the impulse response of the system. As Δt tends to zero these functions will tend to impulse functions times the amplitude of the signal. To find the signal amplitude we need to sum over all the responses from all the pulses giving the *Convolutional integral* between the input signal and the system impulse response

$$g(t) = \int_{-\infty}^{\infty} f(t)g(t-\tau)d\tau = f(t) * h(t)$$

This is often written in the second form since convolution is an operator between functions much like addition and multiplication. It is a very important concept in many different areas of physics and electrical engineering. In words we can say

> **The present value of the response of a linear time-invariant system is the weighted integral over the past history of the input signal, weighted according to the impulse response of the system.**

Thus the impulse response acts as a *memory function* for the system

The Convolution Theorem

The *Convolution Theorem* is one of the most useful results in Fourier Theory. It states that if $G_{12}(t)$ is the *convolution* of $G_1(t)$ with $G_2(t)$ then its Fourier pair $g_{12}(f)$ is the *product* of $g_1(f)$ and $g_2(f)$, the Fourier pairs of $G_1(t)$ and $G_2(t)$. Symbolically

$$g_1(f)g_2(f) \Leftrightarrow G_1(t) * G_2(t)$$

In words:

> **The multiplication of two signals in the frequency domain is transformed into the convolution of their individual Fourier transforms in the time domain.**

The importance of this should now become obvious. If we have a linear system (such as a transmission system or filter) and we want to calculate its response we can either, in the time domain, calculate the convolution between the impulse response of the system and the incoming signal or we can, in the frequency domain, multiply the frequency response of the system with the spectrum of the signal. The latter is usually very much easier than the former which is why we often work in the frequency domain. Note that the inverse process is also true due to the invertability of Fourier transforms - convolution in the frequency domain corresponds to multiplication in the time domain.

If you recall the impulse response of a system is the response to the unit impulse (See 24)). In the frequency domain, the Fourier transform of the unit impulse is a spectrum of unit amplitude, and the frequency response of the system is therefore the response of the system to that spectrum

If we apply a general signal then the response of the system is the *convolution* of the signal with the impulse response of the system. In the frequency domain the response is the *multiplication* of the frequency response of the system with the signal spectrum.

Proof of the Convolution Theorem Proof of the convolution theorem is trivial. By definition we can write

$$G_{12}(t) = \int_{-\infty}^{\infty} G_1(\lambda)G_2((t-\lambda)d\lambda$$

Taking Fourier transforms of both sides gives us

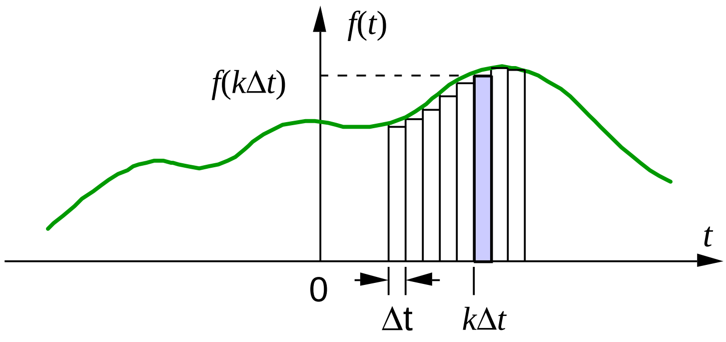

Figure 25: Splitting a signal into Impulses

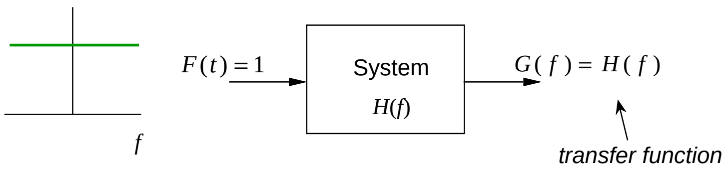

Figure 26: The Frequency Response

$$g_{12}(f) = \int_{-\infty}^{\infty} G_{12}(t)e^{2\pi ft}dt$$
$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} G_1(\lambda)G_2(t-\lambda)e^{2\pi ft}d\lambda dt$$

Now we can change variables, defining $y = t - \lambda$ which allows us to write this integral as

$$g_{12}(f) = \int_{-\infty}^{\infty} G_{12}(t)e^{2\pi ft}dt$$
$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} G_1(\lambda)G_2(y)e^{2\pi f(y+\lambda)}d\lambda dy$$

This integral is separable and so we finally get the result proving the convolution theorem

$$g_{12}(f) = \int_{-\infty}^{\infty} G_2(y)e^{2\pi fy}dy$$
$$\times \int_{-\infty}^{\infty} G_1(\lambda)e^{2\pi f\lambda}d\lambda$$
$$= g_2(f)g_1(f)$$

Convolution: Graphical Interpretation

Here I am going to try and illustrate graphically what the convolution integral is. We are going to use two functions $f_1(t)$ and $f_2(t)$ both of which are a ramp function.

The convolution integral is $g(t) = \int_{-\infty}^{\infty} f_1(t)f_2(t-\tau)d\tau$ Firstly you will notice that the variable of integration is τ and that we are performing the integral of $f_2(t-\tau)$ so we need to reverse this function in time. Now this integral is a function of t and as we change τ we are sliding function $f_2(t-\tau)$ past $f_1(\tau)$ as τ increases.

We are multiplying the two functions together and then integrating. The integration corresponds to finding the area under the product of the two function - it is a measure of the amount of overlap between them which will change as the delay between the functions changes. Obviously it is zero when the two functions don't overlap and is a maximum when they have maximum overlap.

This integral it therefore a function of t as illustrated below.

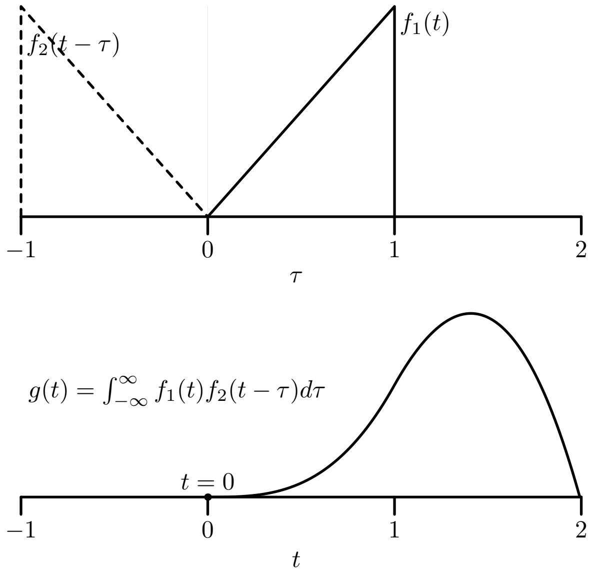

Figure 27: Convolution $\tau = 0$

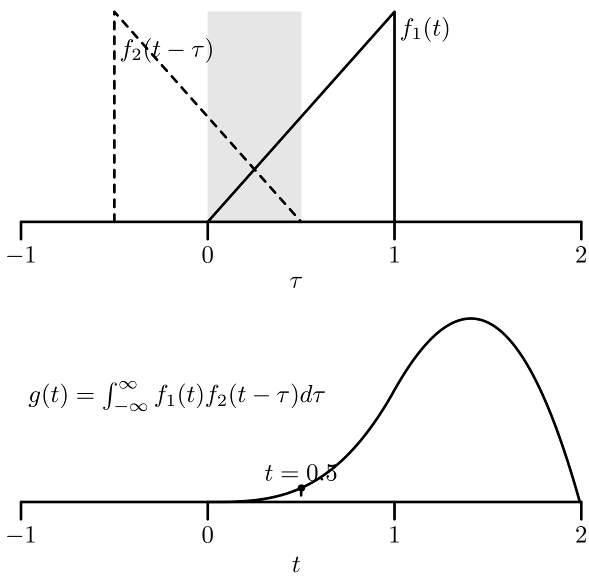

Figure 28: Convolution $\tau = 0.5$

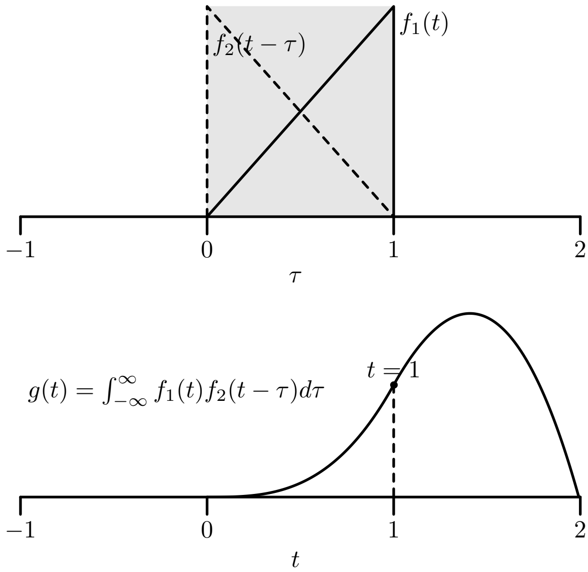

Figure 29: Convolution $\tau = 1$

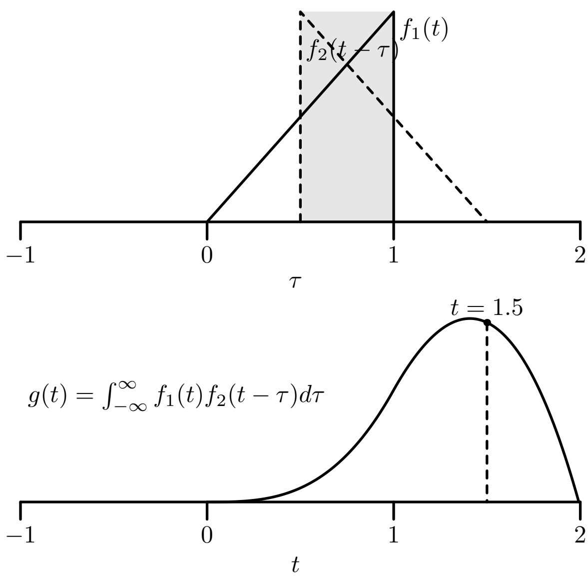

Figure 30: Convolution $\tau = 1.5$

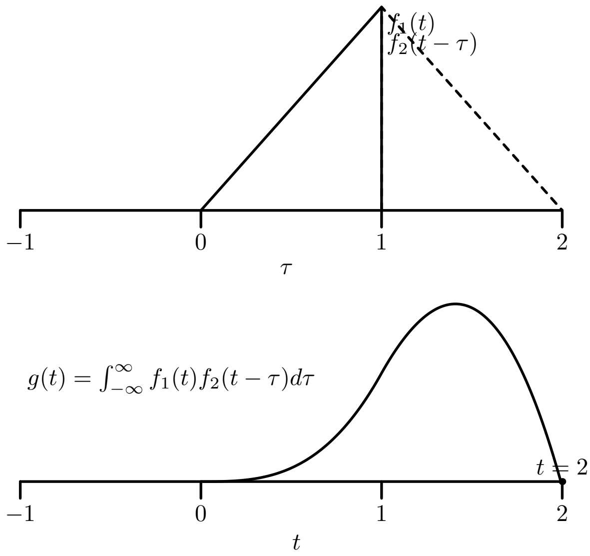

Figure 31: Convolution $\tau = 2$

Ideal Filters

seen how the response of a system can be represented either by its frequency response $H(f)$ or by its impulse response in the time domain $h(t)$ In communication systems we often wish to use filters with a designed response, for example to eliminate noise or compensate for imperfections in the transmission process. Generally the frequency response is used in describing such systems since we can calculate the overall response of a system using multiplications of the frequency response functions. In the time domain we would have to perform convolutional integrals which are much harder. In the system shown above we have an input signal of spectrum $F(f)$ being operated on by a filter $H(f)$ to produce an output $G(f)$ We can write the output in terms of the input and the system response simply as

$$G(f) = F(f)H(f)$$

Note that these are complex response functions. We can separate them into amplitude and phase components giving the expression

$$|G(f)|e^{i\phi_G(f)} = |F(f)|e^{i\phi_F(f)}|H(f)|e^{i\phi_H(f)}$$

where $|H(f)|$ etc are the real amplitude responses and $\phi_H(f)$ are the phase components of the responses. From this we obtain the separate expressions for the amplitude and phase responses

$$|G(f)| = |F(f)||H(f)|$$

$$\phi_G(f) = \phi_F(f) + \phi_H(f)$$

i.e. we *multiply the amplitude responses* and *add the phase responses.*

If we want distortionless transmission through the filter then the output must have the same shape as the input, however it may be delayed in time and may have a different amplitude i.e. for distortionless transmission

$$g(t) = Kf(t - t_0)$$

Taking Fourier transforms of this expression we obtain

$$G(f) = KF(f)e^{-i2\pi f t_0}$$

Therefore we have for the response of an ideal filter, i.e. one which doesn't distort the signal

$$H(f) = Ke^{-i2\pi f t_0}$$

The filter has a constant amplitude and a phase which varies linearly with frequency. Such a filter is not very interesting, filters which are interesting will limit the frequency range. They can be classified into 4 types as shown in the graphs here where I have plotted both the amplitude and phase responses of the ideal filters. A low pass filter only transmits frequencies below a certain value, the opposite is a high pass filter which only transmits a frequency above a certain value. We may also have a bandpass or bandstop filter which transmits or block frequencies inside a certain range. In all cases the bandwidth of the filter is given as W. Firstly you will see that all these ideal filters have a constant amplitude inside their pass regions. They also have a linear variation of phase with frequency.

Decibels And Bandwidth

As already pointed out, in general the transfer function of a linear time-invariant system is a complex quantity and can be written in the form

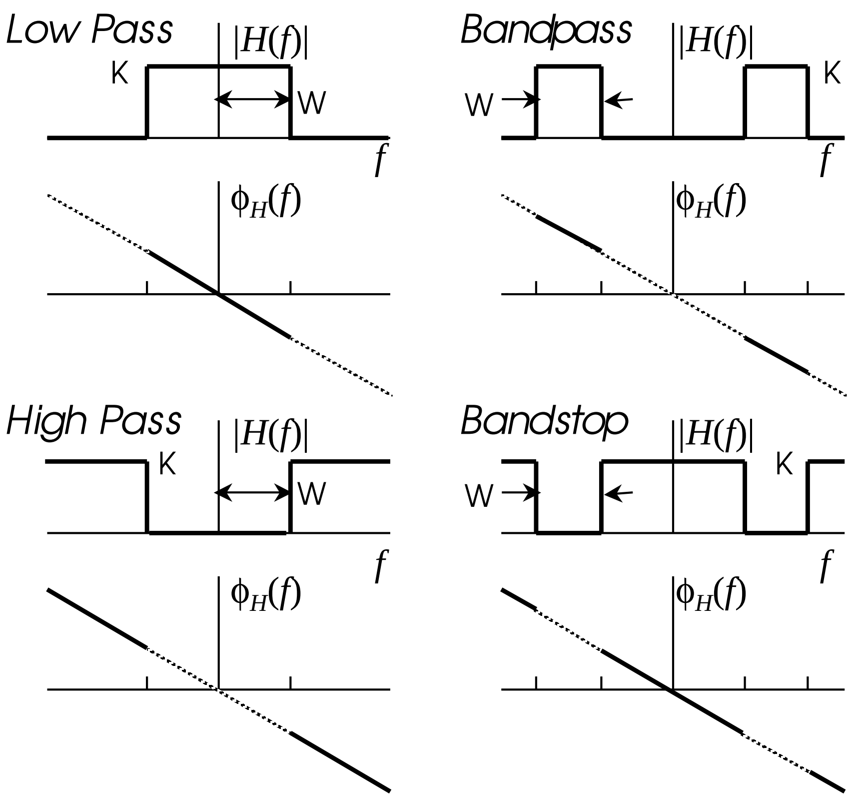

Figure 32: Ideal Filters

$$H(f) = |H(f)|e^{i\phi(f)}$$

In many applications it is preferable to work with the logarithm of $H(f)$ expressed in polar form. We define

$$\ln H(f) = \alpha(f) + i\beta(f)$$

where the system gain in *nepers* is given by

$$\alpha(f) = \ln |H(f)|$$

and the phase $\beta(f)$ is in radians.

More commonly units of *decibels (dB)* are used, where the gain of the system is then defined by

$$\alpha'(f) = 20 \log_{10} |H(f)|$$

The relationship between the gain in decibels and the gain in nepers is then

$$\alpha'(f) = 8.69\alpha(f)$$

Often when we talk about the bandwidth of the system response we mean the distance between the points where the amplitude response has fallen off by 1/2 (the *full-width at half max (FWHM) bandwidth* or -3dB)

Assessment
Deadline: 2009-10-18 00:59:00
No Questions: 1
Time Allowed: 3 min

Causality And Stability

A system is said to be *causal* if it does not respond before the signal is applied. For a linear time-invariant system it is clear that this means that the impulse response $h(t)$ must vanish for negative time.

$$h(t) = 0, \, t < 0$$

Obviously a system operating in real-time must be causal. We have already seen how the ideal low-pass filter is non-causal and therefore not physically realisable since its impulse response is a sinc function which extends to negative time. However in applications where the signal is stored before processing the response may be noncausal and

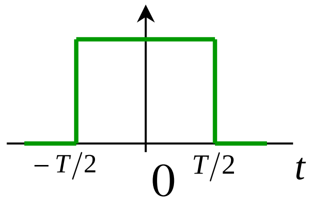

Figure 33: A Rectangular Pulse

yet physically realisable. Storing the signal can be thought of as delaying it - the more of the signal we store the longer it is delayed and the more of the leading part of impulse response we can represent. In terms of our low pass filter, the more we delay the signal the more of the leading edge ripples of the impulse response we can represent, the sharper the filter cut off and and the closer we can approach the ideal response. This is generally true - the more we delay a signal in the filter the sharper the features in the spectral response we can represent. Infinitely sharp features would however require that we delay the signal for infinitely long.

A second requirement we place on a system is that of stability by which we mean that the output is bounded for all input signals. This is called the *bounded input-bounded output (BIBO) stability* criterion. An input signal is bounded if

$$|x(t)| \leq M$$

where M is a positive real finite number. The output, for a linear time-invariant system, is given by the convolutional integral which, if we substitute in the input above becomes

$$|y(t) \leq M \int_{-\infty}^{\infty} |h(\tau)| d\tau$$

It follows that we require this to be bounded and therefore we find that the impulse response must be absolutely integrable.

$$\int_{-\infty}^{\infty} |h(\tau)| d\tau < \infty$$

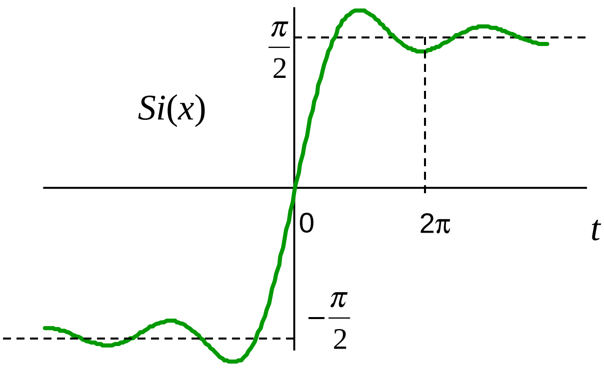

Figure 34: The Sine Integral Function

Pulse Response of an Ideal Low-pass Filter

We have already seen that the impulse response of the ideal low pass filter is given by

$$h(t) = 2W\mathrm{sinc}\left(2W(t - t_0)\right)$$

Now we want to calculate the response of such a filter to a rectangular pulse as shown right. To do this we perform the convolutional integral giving for the output of the low pass filter (replacing limits with those of the pulse). To simplify the calculation we have set $t_0 = 0$ without loss of generality.

$$y(t) = 2W \int_{-T/2}^{T/2} \frac{\sin\left(2\pi W(t-\tau)\right)}{\left(2\pi W(t-\tau)\right)} d\tau$$

To solve this integral we shall change variables putting $\lambda = 2\pi W(t - \tau)$

Giving us

$$\begin{aligned} y(t) =& \frac{1}{\pi} \int_{2\pi Wt - T/2}^{2\pi Wt+T/2} \frac{\sin\lambda}{\lambda} d\lambda \\ =& \frac{1}{\pi} \left[\int_0^{2\pi Wt+T/2} \frac{\sin\lambda}{\lambda} d\lambda - \int_0^{2\pi Wt+T/2} \frac{\sin\lambda}{\lambda} d\lambda \right] \\ =& \frac{1}{\pi} \left[\mathrm{Si}(2\pi Wt + T/2) - \mathrm{Si}(2\pi Wt - T/2)\right] \end{aligned}$$

The solution involves the *sine integral function* defined by

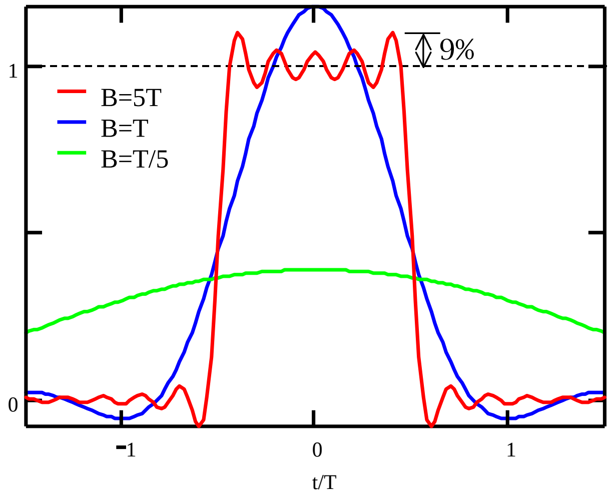

Figure 35: The Rectangular Filter Response

$$\mathrm{Si}(x) = \int_0^x \frac{\sin\lambda}{\lambda} d\lambda$$

We cannot write this integral in terms of other functions, it has to be calculated numerically (or looked up in tables). It is shown graphically in figure 34).

Our response is therefore made up of the difference between two sine integrals - one corresponding to the leading edge and one to the trailing edge. The exact form depends on the filter bandwidth , and in figure 35) it is plotted for 3 different ratios of bandwidth to pulse width.

If the bandwidth is very narrow (the green line) we see that the output is spread out over a long time. If the output equals the bandwidth (the blue line) the output shape is slightly broadened. In most cases the bandwidth of the filter will be much broader than this (the red line).

Those of you who have ever looked at the response of a pulse on an oscilloscope will be familiar with this shape. Important points to notice are that there is overshoot (of about 9% usually) and undershoot, and there are oscillations in the response. This is called *Gibbs Phenomena* . The period of these oscillations is proportional to the reciprocal of the filter bandwidth.

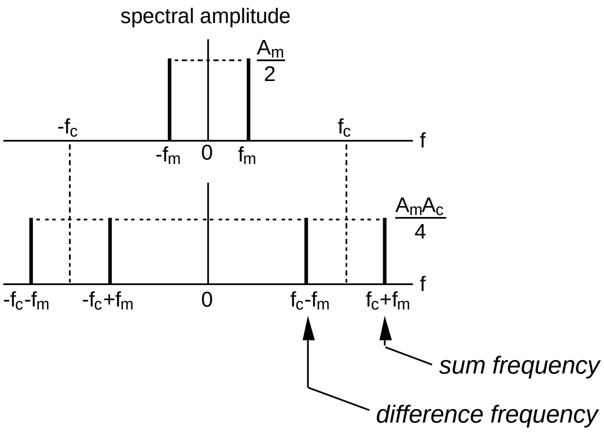

Figure 36: Heterodyning (sine wave)

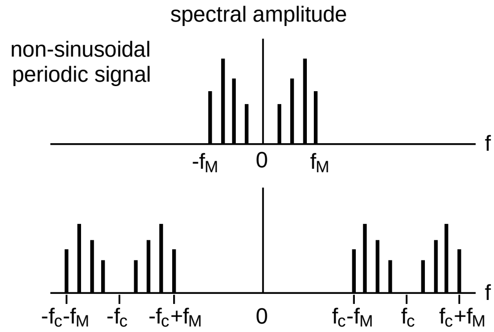

Figure 37: Heterodyning (non-sinusoidal periodic)

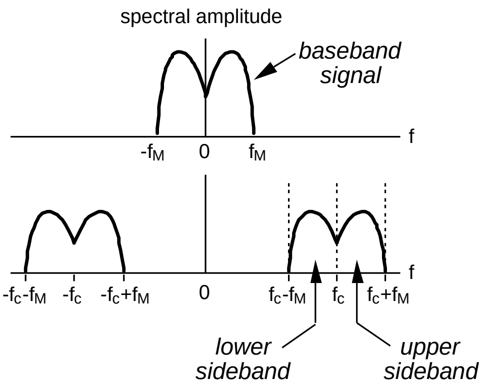

Figure 38: Heterodyning (aperiodic)

Frequency Translation

As discussed earlier one of the important reasons for modulation is to carry out frequency translation. We take a signal and a carrier and multiply them together to produce a mixing or heterodyne signal. For a sinusoidal signal we have

$$\underbrace{f_m(t)}_{\text{signal}} = A_m \cos\omega_m t$$

$$\underbrace{f_c(t)}_{\text{carrier}} = A_c \cos\omega_c t$$

$$\underbrace{f_m(t)f_c(t)}_{\text{heterodyning or mixing}} = \frac{A_m A_c}{2}\left[\cos(\omega_+\omega_m)t + \cos(\omega_0\omega_m)t\right]$$

Looking at the spectrum (figure 36)), we the sinusoidal signal has two delta functions at $\pm f_m$. Multiplying this by the sinusoidal carrier produces the sum and difference frequencies - a total of 4 peaks in the double sided spectrum. One way of looking at this is that we have taken the original spectrum and shifted it up to the carrier frequency.

A common mistake that is made is to forget that the spectrum is double sided - has both positive and negative frequencies. This is because quite often instruments (such as the spectrum analyser) only plot the positive half of the spectrum. For real signals, the amplitude spectrum is symmetric, and the phase spectrum is anti-symmetric, and so there is no need extra information negative half of the spectrum. On such a spectrum analyser we would only see a single peak for the signal sinusoidal frequency. However when we multiply this signal by the carrier we shift the *entire* spectrum up by the carrier frequency and so we see two peaks in the spectrum.

This principle of course applies for general signals so for a periodic non-sinusoidal signal we have a spectrum made up of a set of discrete frequencies separated by the period of the signal. Heterodyning this with a sinusoidal carrier shifts the entire double sided spectrum as shown here.

And similarly for a general aperiodic signal which has a continuous spectrum. Another important point to note about this process is that we effectively see a doubling of the bandwidth of the signal as when heterodyning we see both halves of the original spectrum at positive frequencies.

Another important aspect of frequency translation is that our analysis for the ideal low pass filter can also be applied for bandpass filters since they are equivalent except for the frequency translation of both signal and filter. Haykin deals with proving this is some detail. The most important thing to remember is that there is a factor of 2 difference in the bandwidth definition between the two, and this must be taken into account in the calculations. In particular we can represent a bandpass filter response $H(f)$ in terms of an equivalent low pass transfer function as follows

$$\tilde{H}(f - f_c) = 2H(f),\ f > 0$$

i.e. we take the part of the passband transfer function corresponding to positive frequencies, shift it to the origin and multiply by 2.

Phase And Group Delay

When a signal is passed through a dispersive (frequency-selective) device some delay is introduced. In an ideal low-pass filter the phase response varies linearly with frequency and a constant delay is introduced.

To start with we will consider a frequency which suffers a phase shift total of $\beta(f_c)$ radians. The equivalent delay suffered will simply be given by $\beta(f_c)/\omega_c$ seconds. The is called the *phase delay* of the channel. This is not necessarily the true delay of the channel - after all a sinusoidal signal carries no information. We must modulate the sinusoid to carry information, and it is the delay of the modulating envelope that we are often really interested in. This delay is called the *envelope* or *group delay* of the channel.

In general the phase response of our *dispersive channel* is a function of frequency, i.e. we have $\beta(f)$. An example is plotted in figure 39).

Here I am going to show what I hope is a simple way of viewing the group delay. Haykin does a more complete calculation. We are going to a consider a signal consisting of two sinusoidal components separated by a small frequency step $\delta\omega$ Thus we have at the input

$$\begin{aligned}&\cos\omega t + \cos(\omega + \delta\omega)t \\ &= 2\cos\left(\frac{\delta\omega}{2}t\right)\cos\left(\omega + \frac{\delta\omega}{2}t\right)\end{aligned}$$

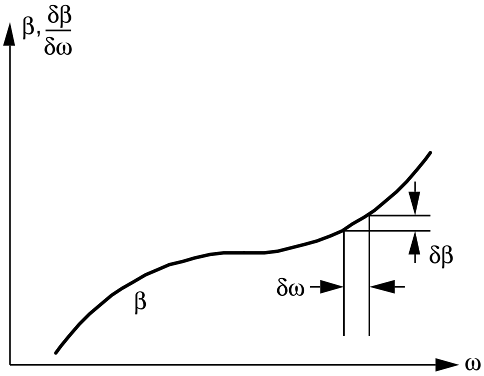

Figure 39: Group Delay

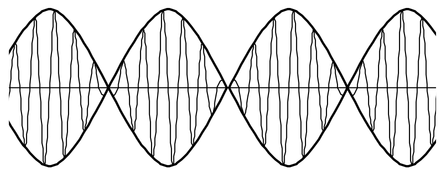
Figure 40: Group Delay Envelope

where we have rewritten the input in terms of the envelope component at frequency $\delta\omega$ and a carrier frequency component.

At the output of the channel or system we have a delay of β for frequency ω and a delay of $\beta + \delta\beta$ for frequency $\omega + \delta\omega$ giving the output signal as

$$\begin{aligned}&\cos(\omega t - \beta) + \cos\left((\omega + \delta\omega)t - (\beta + \delta\beta)\right) \\ &= 2\cos\left(\frac{\delta\omega}{2}t - \frac{\delta\beta}{2}\right)\cos\left\{\left(\omega + \frac{\delta\omega}{2}\right)t - \left(\beta + \frac{\delta\beta}{2}\right)\right\}\end{aligned}$$

Again we have separated the output into the envelope and carrier. Comparing the envelope component at the output with that at the input we see that it has a phase delay of $\delta\beta$ and is at frequency $\delta\omega$ so the group delay is given by

$$t_g = \frac{\delta\beta}{\delta\omega}$$

More generally if we have a more complex signal we need to consider the relative relationship between all the frequency components. Clearly if there is to be no distortion the relative delay between all the components must be the same i.e.

the derivative of the phase with frequency must be a constant.

Assessment
Deadline: 2009-10-18 00:59:00
No Questions: 3
Time Allowed: 10 min

Pulse Modulation

If we use a pulse carrier to effect communications then one of several parameters of the pulse train may be varied in proportion to the message signal. We can distinguish between two types of pulse modulation scheme:

1. analogue pulse modulation
2. digital pulse modulation

Analogue Pulse Modulation Schemes

In analogue pulse modulation, the carrier is a a periodic pulse train and we may continuously vary the pulse amplitude, pulse width, pulse frequency or the position of each pulse in its time slot in proportion to the sampled messages of the message waveform. In each of these cases the informationn is carried in analogue form although the transmission takes place at discrete intervals of time. Examples of these modulation formats are illustrated in figure 41).

Pulse Amplitude Modulation (PAM) This may be considered to be the pulse carrier equivalent of DSB-SC amplitude modulation although it is not exactly equivalent as may be seent when we look at the difference between Natural Sampling (Section) and Pulse Amplitude Modulation (Section) sampling.

Pulse Width Modulation (PWM) The width (duration) of each carrier pulse is varied according to the amplitude of the message signal. Signal recovery is easily acheived by low pass filtering.

Pulse Position Modulation (PPM) The position of each carrier pulse within its time slot of width T is varied. Signal recovery is acheived by measurement of the time interval between this pulse and a regular clock pulse train operating at rate $1/T$.

Pulse Frequency Modulation (PFM) This is directly equivalent to FM. The pulse repetition frequency is varried in porportion to the message signal samples; signal recovery is as for FM. PFM is used in low-cost, high performance analogue communications using fibre optics where nonlinearities in the transfer characteristics of optical sources impair the performance of PAM.

Nyquist Sampling Theorem

From what we have learn't in the tutorial on modulation basics we know that if we sample a signal at regular instants in time we are going to produce a periodic spectrum. Nyquists theorem is concerned with knowing how often we sample the signal to that we can reproduce it witout distortion. If we have a signal strictly bandlimited to bandwidth W and at regular intervals seperated by T seconds (rate $1/T$) then the resulting spectrum is going to be copys of the spectrum seperate by $1/T$ Hz as shown in figure 42).

Note that because we are modulating the original signal we see its whole double sided spectrum of width $2W$ copied at frequency intervals of $1/T$. Therefore we see that if $T \leq 1/(2W)$ these multiple copies of the signal spectrum are completed seperated, and we could filter out one to regenerate the original signal without distortion. However if we sample at too low a rate, $T > 1/(2W)$ then we see that there is overlap between the copies in the sampled spectrum and there would be distortion in the recovered signal. Therefore we can state the requirements for sampling, the *Nyquist Sampling Theorem* as follows:

Important

A band-limited signal of finite energy, which has no frequency components higher than W Hz is completely described by sampling values of the signal at instants of time separated by $1/2W$ seconds.

If we do not satisfy this criterion we cannot recover the original signal. In particular we may see fre-

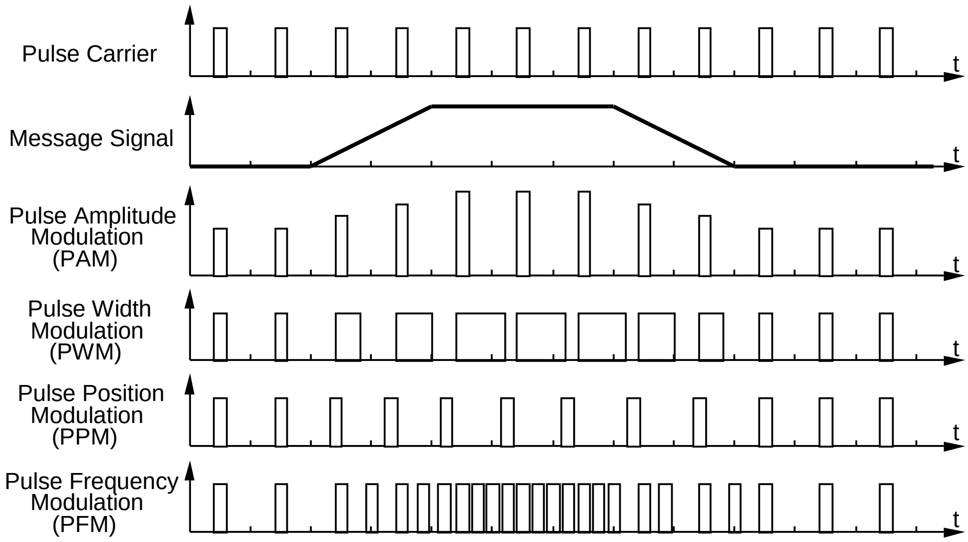

Figure 41: Pulse Modulation Schemes

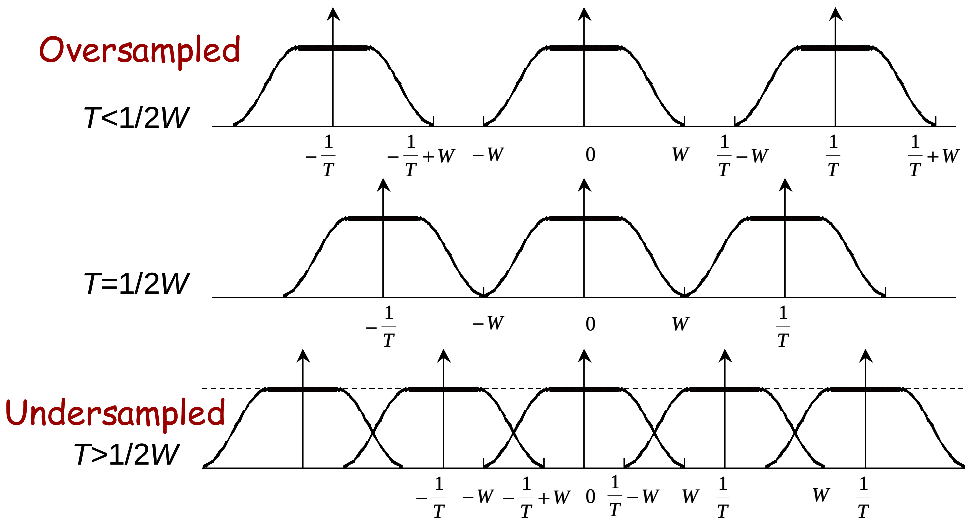

Figure 42: Nyquist Sampling Theorem

quency components which do not exist in the original signal. This is called *aliasing*. For example if we have a 4 kHz input signal and sample at a rate of only 5kHz we will see frequency components at (5+4)=9kHz which is removed by a low pass filter and at (5-4)=1kHz which isn't i.e. we will see a 1kHz signal after low pass filtering a 4kHz signal sampled at only 5kHz.

An example, which you may see in the laboratory, is that when using a digital sampling oscilloscope which has too low a sample rate (i.e. with a low time base set) and we look at high frequencies. We may actually see low frequency aliased signals which are not present at the input to the oscilloscope. Care must therefore be taken when using such instruments - in particuar if the frequency of the signal you are measuring changes as you change the timebase (sample rate) then it is probably oscilloscope aliasing that you are seeing.

Assessment
Deadline: 2009-10-31 23:59:00
No Questions: 1
Time Allowed: 5 min

Natural Sampling

In natural sampling a bandlimited signal $f(t)$ is multiplied by a rectangular pulse train $p(t)$ (the periodic gate function) with period T (the sampling period) to give the sampled signal $f_s(t) = f(t)p(t)$.

We start with the information signal which is smooth in time and has a limited bandwidth.

Our gate function is a rectangular pulse train which has a spectrum which is the sinc function evaluated at multiples of the repetition frequency.

Now we multiply these together (in the time domain). The resultant is the convolution of the two spectra in the frequency domain, which is particularly easy to evaluate due to discrete nature of the sampling spectra. It is the signal spectrum, repeted at multiples of the sampling frequency and multiplied by the amplitude of the sampling spectrum at those points. Natural sampling is characterised by sampling pulse tops which precisely follow the variations in $f(t)$.

The original signal can be recovered simply by applying a low pass filter to select only the baseband component of the spectrum which is the same as the signal spectrum. It is clear that we do not need to use rectangular pulses at all as selection of the sampling pulse shape serves only to specify the shape of the envelope of $F_s(\omega)$ which won't affect the result after filtering.

Pulse Amplitude Modulation

In pulse amplitude modulation the sample pulses are flat topped, and so, in contrast with natural sampling, do not follow the variations of the signal being sampled. The digital circuit used to achieve PAM is referred to as *sample and hold* and it comprises two operations: the instantaneous sampling of the message signal $f(t)$ followed by lengthening the duration of each sample to a constant value τ . The value of τ is chosen to reduce the bandwidth requirements: if it is too short the transmission would require excessive bandwidth. It is important to note the distinction between this process and natural sampling.

In the case of flat-topped sampling our input signal is the sampled at discrete points in time so its spectrum is the baseband spectrum repeated at multiples of the sampling frequency as shown in figure 46).

The next step in the process is the hold operation which amounts to passing these samples through a filter to acieive the required rectangular pulse shape - a filter with a rectangular impulse response $q(t)$. This hold function is illustrated in figure 47). Its spectrum is $Q(\omega)$

The output of this filter is the convolution of the samples with the rectangular hold function which corresponds to the multiplication of the two spectra as shown in figure 48).

This spectrum differs from the naturally sampled one, the PAM output spectrum is obtained by multiplying together two frequency functions and the original form of $F(\omega)$ is not maintained - the replicas are not true replicas of the signal spectrum.

If we were to pass this signal through a low pass filter as we did for the natural sampling case we would obtain the spectral output which is $F(\omega)$. The way around this is to pass the recovered signal through another filter, with a transfer function $1/Q(\omega)$. This is referred to as *equalisation*, and the additional filter is an *equaliser*. The importance of the hold time τ should now be clear. If we use a shorter hold time τ then its spectrum will

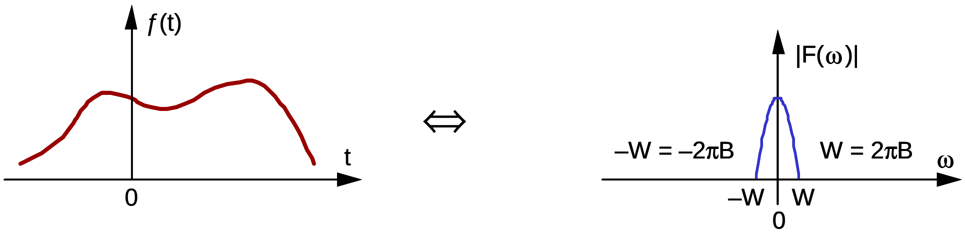

Figure 43: Information Signal for Natural Sampling

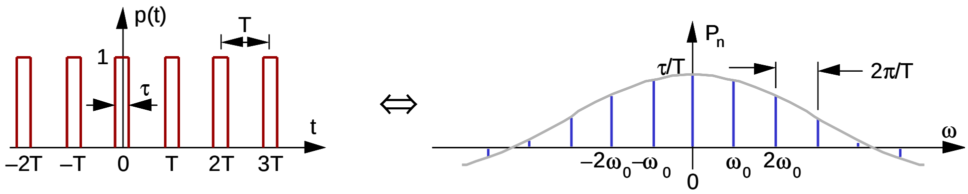

Figure 44: Gate Function for Natural Sampling

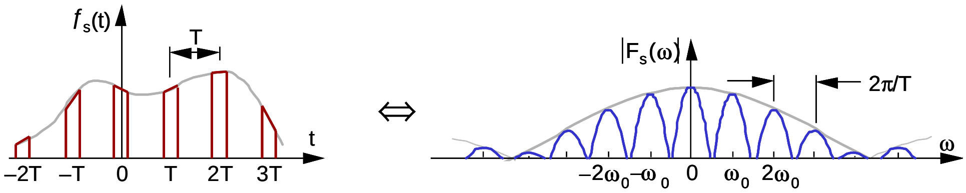

Figure 45: Natural Sampling of a waveform in the time and frequency domains

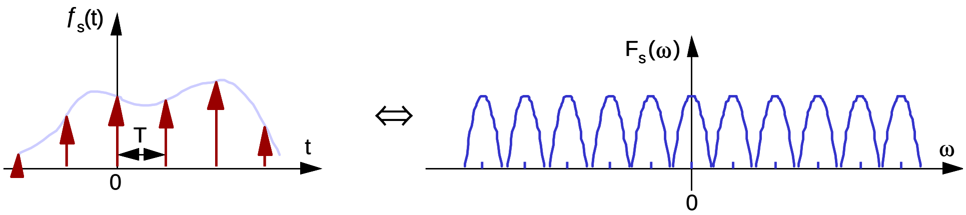

Figure 46: The Discretely Sampled information signal

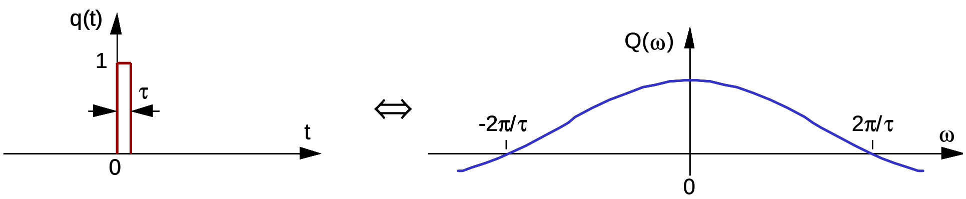

Figure 47: Rectangular Pulse for Flat topped Sampling

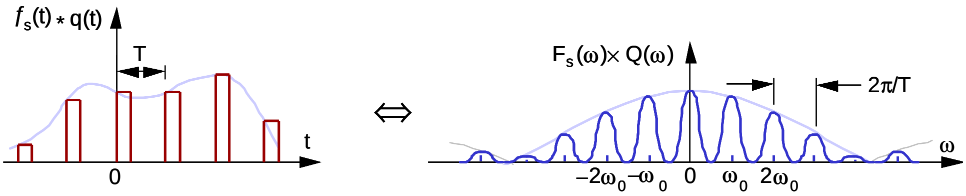

Figure 48: Flat Topped Sampling of a waveform in the time and frequency domains

broaden and become flatter and the distortion produced on the received spectrum would be reduced. It is found that if the ratio of sample time to sample period $\tau/T \leq 1$, the maximum difference between the equaliser transfer function $1/Q(\omega)$ and the ideal low pass filter is <1% in which case an additional equalisation filter is usually considered unnecessary.

Why are flat topped pulse so important when a naturally sampled signal is recoverable without distortion by low pass filtering. The reason is that the need to preserve the sampled pulse shapes in transmission conflicts with the basic advantages of digital communications. In transmission over distances which require mid-path amplification, the effects of additive noise would make the natural sampling format no better than analogue. When the pulse shape is not important, as with flat-toped sampling used in PAM, repeaters may be use to regenerate rather than amplify the signal giving giving more robustness in the presence of noise.

If the sampling pulse is very much smaller than the sampling interval then the signal power at the output low pass filter (LPF) at the receiver may be very small, requiring large amounts of amplification. An efficient way around this is to use a *sample-and-hold* circuit at the receiver, as illustrated in figure 49). In this circuit the switched is closed for the duration of the sample pulse. If the source impedance R_s is small the capacitor charges to the input voltage level within the sample time. Between samples the switch is opened. If the load resistance R is large the capacitor will retain its voltage between samples until the switch is closed again.

The output of the sample and hold circuit is therefore as shown: smoothing is obtained by passing the signal through a LPF. The sample and hold is a reliable and efficient PAM demodualtor with good noise immunity and removes the need for large amounts of equalisation. Note however that equlisation is required since the output of the sample and hold circuit is effectively long flat-topped sample pulses.

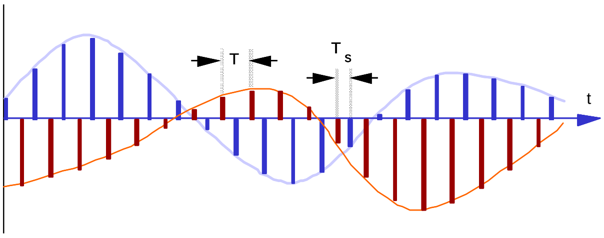

Figure 51: A Time Division Multiplexed Signal

Time Division Multiplexing

With relatively short sampling pulses PAM signals occupy only a small fraction of the sampling interval, and room is left between pulses into which samples from other signals can be insereted. This technique of combining pulses from N independant channels in a definite time sequence is referred to as *time division multiplexing (TDM)* . The principle can be applied to most pulse modulation formats. It is illustrated in figure 50) for PAM signals.

Each of the N message signals is band limited by passing it through a low-pass filter (LPF). The filtered signals are passed to a *commutator*, which, in practice, constructed from high speed digital switches but can bethought of as a rotating switch which sequentially switched to the output of each LPF. The commutator takes a narrow sample of each of the N signals at the sampling rate $f = 1/T$ at least equal to twice the LPF bandwidth as required by the sampling theorem. It also interleaves the N samples within the sampling interval T. In

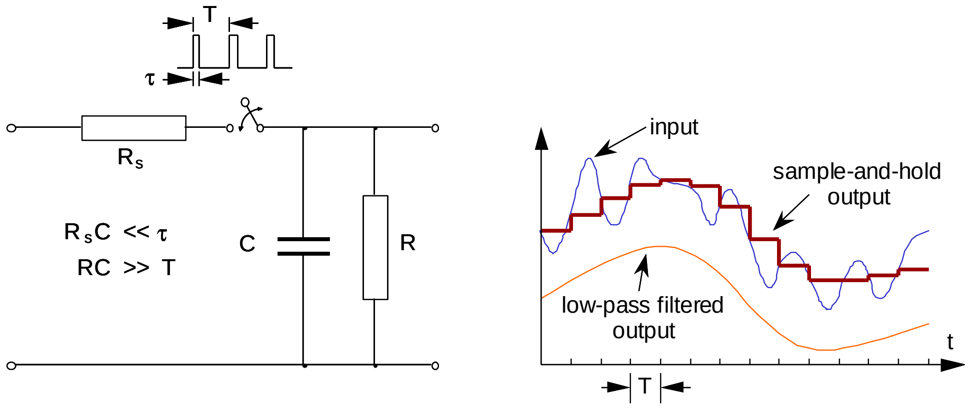

Figure 49: Sample and Hold

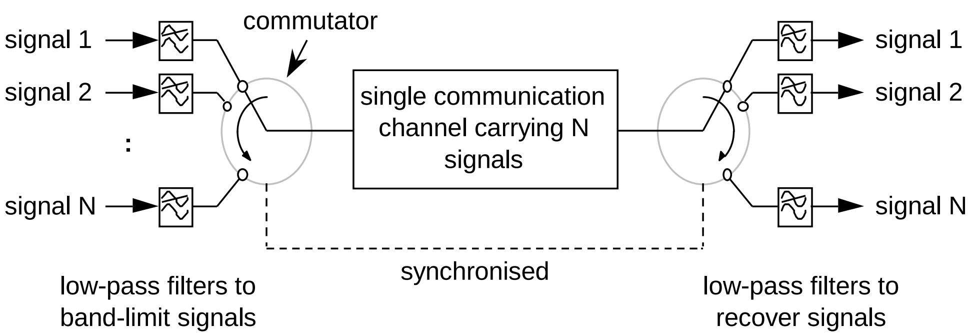

Figure 50: Time Division Multiplexing Operation

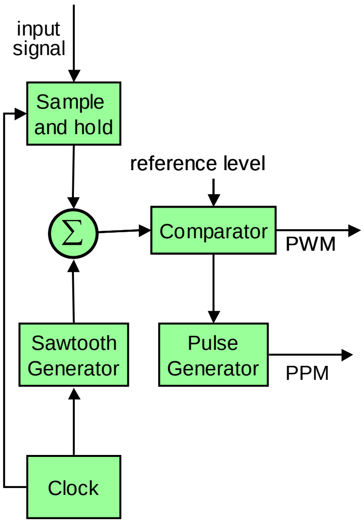

Figure 52: Pulse Position and Pulse Width Modulator

figure 51) the result of this process is shown for 2 signals. In the PAM case the TDM output pulse train so obtained is transmitted; for other pulse modulation formats a further pulse modulator may be used. At the receiving end of the channel the TDM signal passes through another commutator which sequentially distributes the pulses to each of N outputs where each message is recovered by low pass filtering. The second commutator must operate in complete synchronism with the first.

Note that the channel bandwidth B_c required to pass the TDM signal must also satisfy the Nyquist condition. Thus the N interleaved pulse trains constitute a single pulse train with separation $T_s = T/N$. The sampling theorem requires $B_c \geq 1/2T_s$ in order to prevent information loss. The TDM signal can, in effect, be filtered to B_c (low pass filtered) yet still permit separation of the constituent messages by resampling at the receiver using, for example, the sample and hold circuit (See 49)).

PPM and PWM Generation

Both pulse position modulation (PPM) and pulse width modulation (PWM) signals can be generated using the scheme shown here.

The input signal is fed through a sample and hold circuit. Each of the samples is added to a sawtooth waveform generated synchronously to produce the sum waveform shown in figure 52). This waveform is sent to a comparator with a threshold set, as indicated by the blue line on the sum waveform. The incoming amplitude , when added to the sawtooth, will cause a change in the time where the sum waveform crosses the threshold. Therefore the the amount of time spend above threshold will depend on the signal amplitude, and we have pulse width modulation. With PWM, the information is actually contained in the relative positions of the pulse edges. Consequently, the longer pulses expend significant amounts of power which is not carrying information. The PPM signal is produced by triggering a pulse generator on the falling edge of each PWM pulse to generate a pulse of constant width. Note that although PPM is generally an efficient analogue pulse modulation scheme it does require a local generation of the clock timing since, in contrast to PAM and PWM which carry clearly recognisable clock timing, the timing is lost.

Since the PPM system is band limited, it must have a finite rise time and this must place uncertainty in the determination of the input signal which must be proportional to the rise time of the system, inversely proportional to its bandwidth. The uncertainty in the system is referred to as the *resolution*. Note, the resolution only has a meaning in a system subject to noise and distortion since otherwise a given point on the rise time characteristic will always correspond to the same relative timing instant. Generally the accepted criterion for specifying the resolution of a PPM system equates to the rise time. It follows that any two pulses in a PPM train must always be separated by at least the width of the system impulse response.

Pulse Code Modulation (PCM)

The previous pulse modulation techniques are analogue in that even though the signal is sampled at discrtet intervals of time its amplitude is transmitted as an analogue parameter of the pulse waveform. The signal amplitude may itself be *quantised* as shown in figure 54). The available amplitude range is divided into a number of discrete amplitude intervals and the signal represented by the quantised level nearest to the true amplitude value at each sampling instant. This forms quantised PAM (QPAM) or M -ary PAM where M is the number of

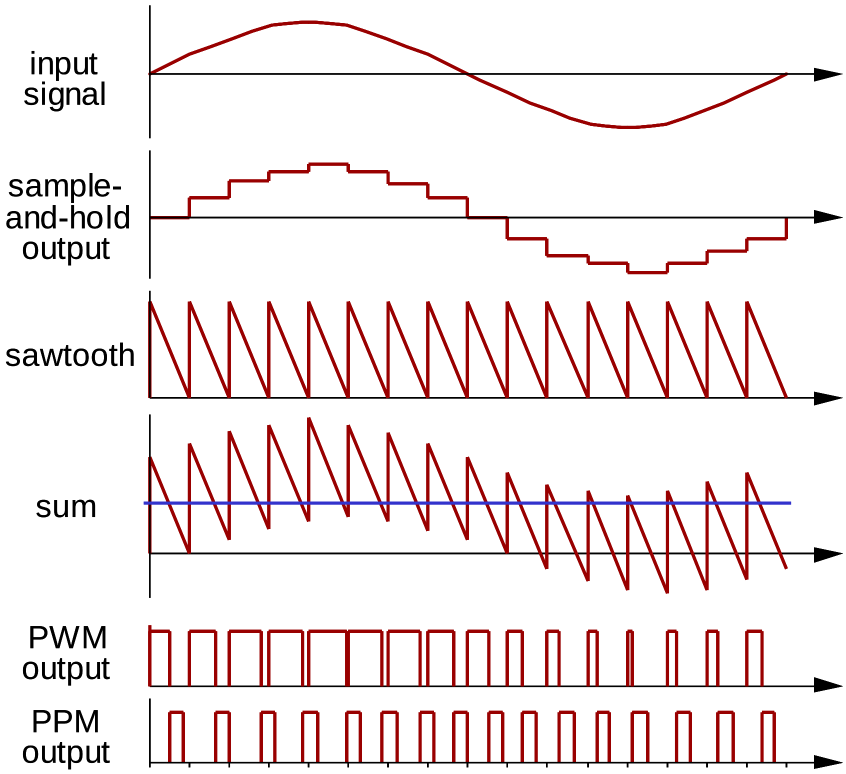

Figure 53: Pule Position and Pulse Width Modulation Generation

levels. To achieve *digital* pulse transmission a *code* is assigned to each assigned level and codes corresponding to the quantised signal at each sampling instant are transmitted as a pulse stream. This the basis of *Pulse Code Modulation* , the advantages of which are as follows:

1. Digital transmission provides robustness against noise and interference
2. PCM signals can be regenerated at intermediate repeaters.
3. Modulating and demodulating circuits are entirely digital offering compatibility with VLSI: high reliability and low cost
4. Signals can be stored in memory; digital signal processing operations such as time scaling can be easily performed
5. Encryption using special codes can allow secure communication
6. Source coding may be used to avoid unnecessary repetitions of frequent message components

Various coding formats may be used in practice; for the time being we shall assume that the signal quantised levels are transmitted as binary digits (bits); the the number n of bits required to represent any single sample is

$$N = \log_2 M$$

where $M = 2^n$ is the number of quantisation levels. The overall transmission rate R is then given by

$$R = \log_2 M \times \text{sampling rate bit/sec}$$

In early PCM systems for telephony, 128 amplitude levels where considered adequate requiring 7 bits/sample. One extra supplementary bit was added to give 8 bits/sample. With the upper frequency limit for speech of 3.4kHz, sampling 8kHz is adequate to satisfy the sampling theorem (allowing for non-ideal filtering); thus, the overall bit rate for each speech channe is established to be 64 kbit/s. Later practice used 256 levels giving n =8 but the supplementary bits where transmitted seperately so the basic octet structure was retained. For high-fidelity sound, n =16 bits are used i.e. 65536 levels.

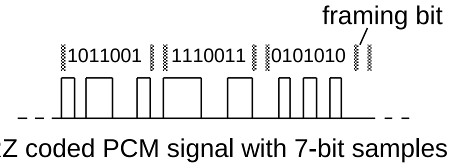

Figure 56: NRZ coded PCM signal with 7 bit samples

Assessment
Deadline: 2009-10-31 23:59:00
No Questions: 1
Time Allowed: 10 min

PCM Transmission

There are three steps in PCM generation (1) sampling, (2) quantisation and (3) encoding as illustrated in figure 55). In (1) the message is sampled using a train of narrow pulses to closely approximate the instantaneous sampling process, at a rate > twice the highest frequency in the message. In practice an anti-aliasing filter is used to exclude unwanted frequencies at greater than half the sampling rate. In (2) the sampled version of the signal is quantised: this process may use uniform or non-uniform sampling intervals. In (3) each quantised sample is converted to a coded bit sequence: a choice of binary or tertiary codes may be used. Functions (2) and (3) together can be performed by an *analogue-to-digital converter (ADC)*. The encoded bit sequence may then be transmitted serially as shown in figure 56).

The first element in the PCM receiver (figure 57)) is a regenerator which performs three functions: equalisation, retiming and thresholding. Equalisation compensated for known amplitude and phase distortion in the channel. Retiming is performed using a locally generated clock, at a frequency extracted from the incoming signal, to sample the equalised signal at instants of maximum SNR. In the thresholding operation decisions are made as to whether the incoming bit is a 1 or a 0 and a

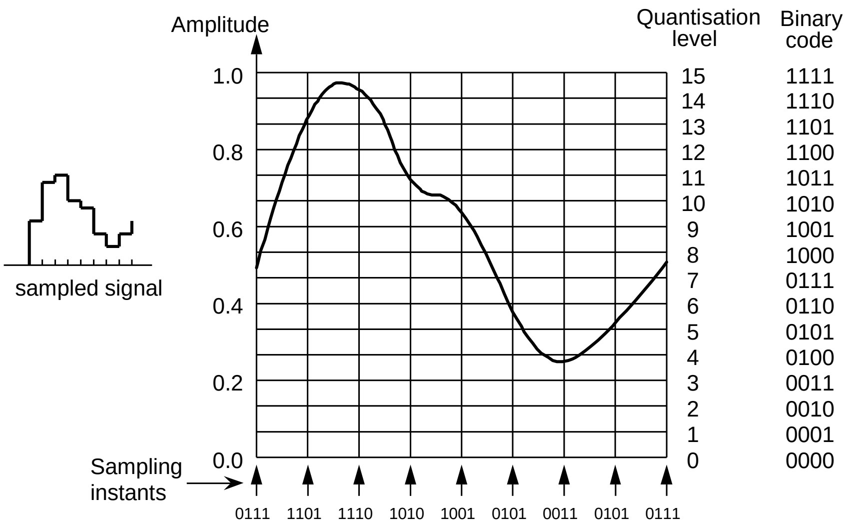

Figure 54: Pulse Code Modulation

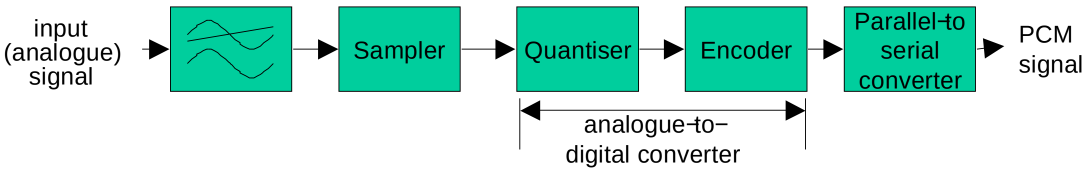

Figure 55: PCM Transmitter

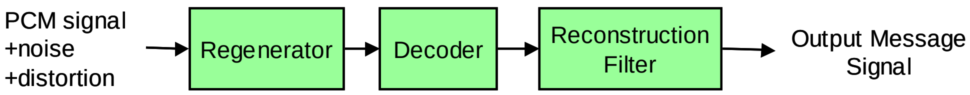

Figure 57: PCM Receiver

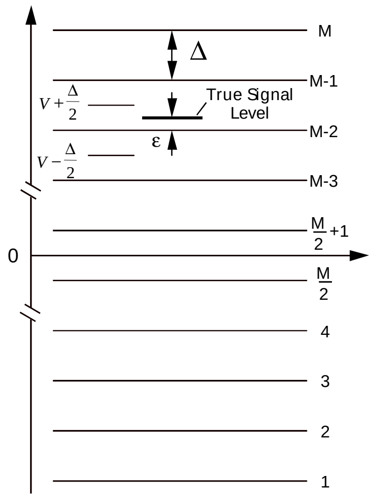

Figure 58: Quantisation Levels

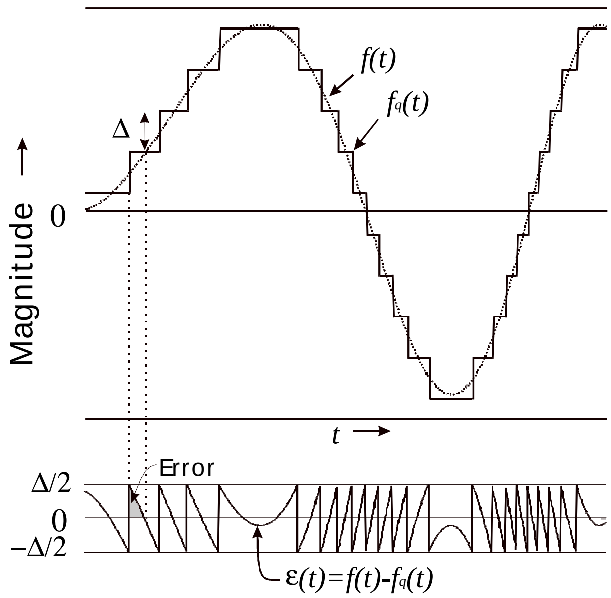

Figure 59: Quantisation

new noise-free pulse generated according to the outcome. Thus noise on the incoming signal is completely removed. This signal may then be decoded and fed to a filter to regenerate the original message signal.

Quantisation

Quantisation converts a continuous input signal $f(t)$ into a discrete-valued approximation $f_Q(t)$. The quantisation error is defined by $\epsilon(t) = f(t) - f_Q(t)$ and clearly must lie in the range $-\Delta/2 < \epsilon < \Delta/2$ where Δ is the quantisation interval (see Figure 58)). Thus the magnitude of the worst case quantisation error is $\Delta/2$. At the receiver this uncertainty is similar to that produced by an additive noise source and hence the effect is referred to as *quantisation noise* . Figure 59) shows plots of a signal, the quantised signal and the resulting quantisation noise.

Since ϵ is actually a voltage the quantisation noise power will just be its variance (mean square value) given by

$$\overline{\epsilon^2} = \int_{-\infty}^{\infty} \epsilon^2 p(\epsilon) d\epsilon$$

If we assume that all the values in the range $-\Delta/2 < \epsilon < \Delta/2$ are equally probable for any sample then the probability density function is given by $p(\epsilon) = 1/\Delta$ thus the quantisation noise power is given by

$$\begin{aligned}\overline{\epsilon^2} &= \int_{-\Delta/2}^{\Delta/2} \epsilon^2 d\epsilon \\ &= \frac{1}{\Delta}\left[\frac{\epsilon^2}{3}\right]_{-\Delta/2}^{\Delta/2} \\ &= \frac{\Delta^2}{12}\end{aligned}$$

If we have a signal that occupies the entire quantised range, the peak signal amplitude is $M\Delta/2$ where for a binary signal $M = 2^n$ then the signal to quantisation noise (power) ratio is given by

$$(\mathrm{SNR}_Q)_{\mathrm{peak}} = \frac{(M\Delta/2)^2}{\Delta^2/12} = 3M^2 = 3 \times 2^{2n}$$

or in units of dB

$$\begin{aligned}(\mathrm{SNR}_Q)_{\mathrm{peak}} =& 10[\log_{10} 3 + 2n \log_{10} 2] \\ =& 4.8 + 6n \text{ dB}\end{aligned}$$

Now if the input is sinusoidal and occupies the full quantiser range, the average signal power is

$(M\Delta/2)^2/2$ and the mean signal-to-quantisation noise (power) is then given by

$$\begin{aligned}(\mathrm{SNR}_Q)\mathrm{mean} =& 10\left[\log_{10} 1.5 + 2n \log_{10} 2\right] \\ =& 1.76 + 6n \text{ dB}\end{aligned}$$

Note that the SNR_Q increases by 6 dB for each additional bit used to quantise the signal.

The error signal, as shown in figure 59), is approximately a sawtooth of period $1/f_s$ where f_s is the sampling frequency. As the sampling frequency is increased the frequency of the error signal also increases however the power is fixed by equations above. Therefore with increasing frequency the noise power is spread over a broader bandwidth and there is less noise power occupying the same bandwidth as the signal. The noise power at the output of a bandpass filter will therefore decrease as the sampling frequency is increased. Thus oversampling a band limited waveform will increase the SNR_Q at the output of a reconstruction filter. This principle is used for advantage in compact disc players for example.

Assessment
Deadline: 2009-10-31 23:59:00
No Questions: 1
Time Allowed: 5 min

Companding

When considering quantisation signal to noise we assumed that the signal fully occupies the encoder range. The SNR_Q will be considerably lower for a weaker signal. If we vary the step size making it smaller for low levels and larger near the maximum input level then the SNR_Q can be made constant over as wide a range of input values as possible. This is referred to as *companding*, *compressing* the signal then encoding it in a linear encoder and ex *panding* the decoded signal with the inverse of the compression characteristic. Since the SNR increases with M^2 we ideally would require that the amplitude step size decreases in the form

$$\frac{dV_{out}}{dV_{in}} = k/V_{in}$$

Integrating this gives the ideal compression characteristic as

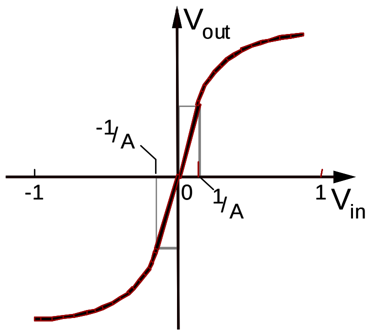

Figure 60: A-law compression characteristic

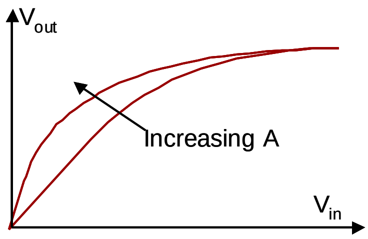

Figure 61: Variation of A-law characterstic with A

$$V_{out} = 1 + k \ln V_{in}$$

Unfortunately this characteristic is not practical as $V_{out} \to -\infty$ as $V_{in} \to 0$. Instead two practical characteristics are used. These are the μ -law characteristic used in North America

$$V_{out} = \frac{\ln(1 + \mu V_{in}}{\ln(1 + V_{in})}, \; 0 \le |V_{in}| \le 1 \qquad (5)$$

and the A law characteristic used in Europe

$$\begin{aligned} V_{out} &= \tfrac{AV_{in}}{1+\ln A} & 0 \le |V_{in}| \le \tfrac{1}{A} & \quad \text{linear region} \\ V_{out} &= \tfrac{1+\ln(AV_{in})}{1+\ln A} & \tfrac{1}{A} \le |V_{in}| \le 1 & \quad \text{logarithmic region} \end{aligned} \qquad (6)$$

A is known as the compression coefficient. The linear region ensures $V_{out} = 0$ when $V_{in} = 0$ and the logarithmic region is specified so $|V_{out}| = 1$ when $|V_{in}| = 1$. The characteristic is continuous at $|V_{in}| = 1/|$. For large values of A the characteristic is predominantly logarithmic and the SNR_Q is approximately constant. This is illustrated figure 61).

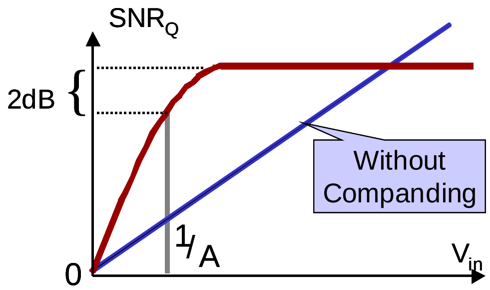

Figure 62: A-law SNR characteristic

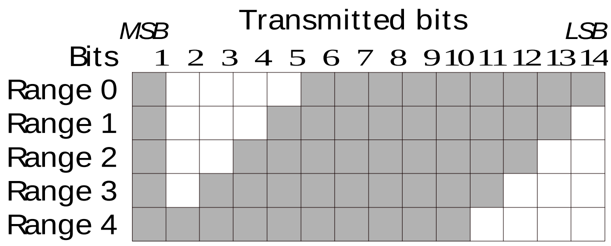

Figure 64: NICAM Encoding

In figure 62) we show a plot of the quantisation SNR_Q for the companding case (red) compared to the case without companding (blue). The improvement in SNR_Q for low signal levels is clear. The linear part of the characteristic causes a drop by 2dB in the SNR_Q for input levels below $1/A$. The system is designed so the rms. value of the smallest signal is equal to $1/A$. The SNR_Q is a function of both A and M which are chosen to give acceptable performance over a desired dynamic range (ratio between strongest and weakest signal). The CCITT recommended value of A is 87.6 which gives a useful volume range (UVR) of 26dB. This value with an 8-bit code gives an SNR_Q of 38 dB. An equivalent linear encoder would require 12-bits to have the same performance so companding is effectively a bit reduction technique which reduces the required transmission bandwidth by 33%.

In practice coders use linear quantisation with different step sizes over segments of the compression characteristic, as illustrated in figure 63). This is technologically easier to realise. 14 segments are used, 7 for positive values and 7 for negative values. The step size in segment 7 is 56 times that in segment 1 giving the characteristic shown above. The total number of input levels is 2^{13}. In segment 1 64 input levels are mapped unto 32 output levels so low level values are quantised at the equivalent of 12 bits linear quantisation.

Nearly Instantaneous Companded Audio Multiplex (NICAM)

Companding compression reduces the resolution (and thus increases the absolute quantisation noise) for the higher signal levels. An alternative form of compression can be based on *range coding* in which the input waveform is divided into a fixed number of ranges with different groups of bits transmitted for the different ranges. This is the technique used in the NICAM *Nearly Instantaneous Companded Audio Multiplex* compression commonly used for high quality television stereo sound signals. It has a superior noise performance compared to A-law for higher signal amplitudes. For low input amplitudes only the least significant bits are transmitted while for high input amplitudes only the most significant bits are transmitted.

In the case of NICAM 14 bit resolution is used which is compressed to 10 bits with for transmission using 5 ranges as shown in the table left. In addition to the data bits a 3 bit range code must also be transmitted. Bit reduction is achieved because the range code is transmitted for a block of amplitude samples adding only a fraction of a bit per sample. NICAM has a sampling frequency of 32HKz and transmits a range code every 32 samples representing a time interval of 1ms which is *nearly instantaneous* as far as the audio signal is concerned. To avoid clipping the range code corresponds to the largest bit in the block. A schematic of a NICAM coder is illustrated in figure 65).

Since 3 bits can define 8 ranges this represents an inefficiency. This is reduced in NICAM by taking 3x32 bit blocks which would have $5^3 = 125$ range combinations and transmitting these as a 7 bit code producing an overall delay of 3ms in the audio signal. Errors in the range code would cause substantial distortion of the signal so 4 parity bits are added to the 7 bit range code to give it a Hamming distance of 4.

NICAM effectively transmits the signal with 10 bits and a constant signal to noise ratio as signal amplitude varies. It has an SNR_Q improvement of

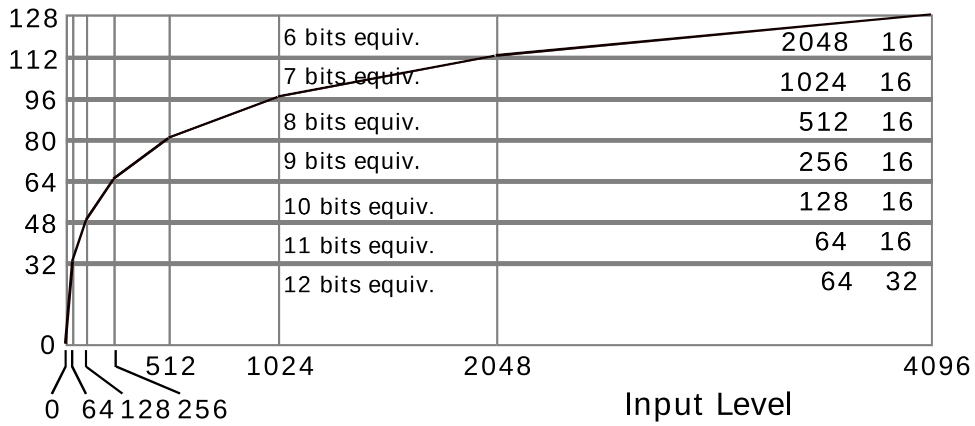

Figure 63: Segmented Compression Characteristic

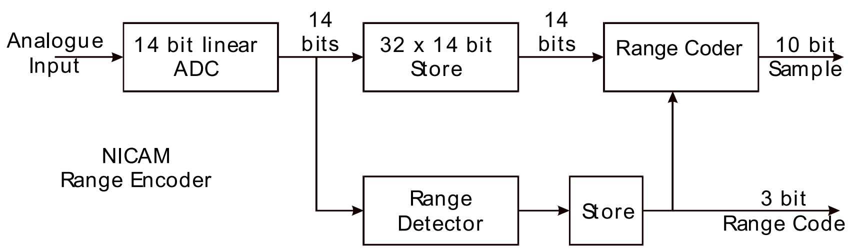

Figure 65: Nicam Encoder

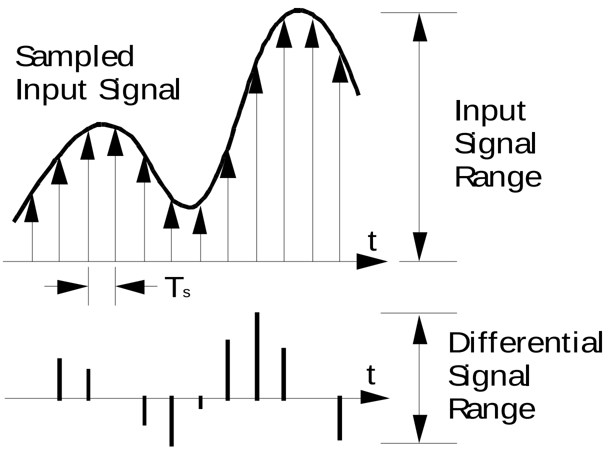

Figure 66: DPCM Concept

approximately 12dB over simple A-law compression but requires a more complex encoder and decoder. A 10 bit linear quantiser would have an SNR_Q of 60 dB but with range coding this is increased to a subjective equivalent of 80dB.

Differential Pulse Code Modulation (DPCM)

In PCM each sample of the waveform is encoded independently of all the others. However most source signals sampled at the Nyquist rate or faster exhibit significant correlation between successive samples and so an encoding scheme exploiting the redundancy in the samples will result in a lower bit rate for the source output.

The simplest solution would be to encode the differences between successive samples which being smaller than the actual sampled amplitudes would result in fewer bits being required. This is illustrated in figure 66) where we can see clearly that the range of the differential signal is less than the input signal range.

In differential pulse code modulation this technique is refined to predict the current sample based upon p previous samples and the difference or error between the actual sample and the prediction is encoded as illustrated here.

The process is reversed at the decoder

Let x_n be the current sample and $\hat{x}_n$ the pre-

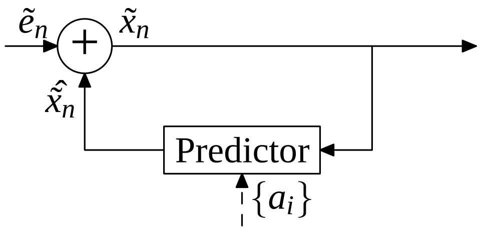

Figure 68: DPCM Decoding

dicted value can be given as a weighed linear combination of p previous samples using the $\{a_i\}$ predictor coefficients

$$\hat{x}_n = \sum_{i=1}^{p} a_i x_{n-i}$$

The transmitted difference signal is then

$$e_n = x_n - \sum_{i=1}^{p} a_i x_{n-i}$$

The predictor coefficients $\{a_i\}$ are chosen to minimise some function of error between x_n and $\hat{x}_n$ i.e. minimise the output quantisation noise $\mathrm{SNR}_O = \sigma_x^2/\sigma_Q^2$ where σ_x^2 is the variance of the original input and σ_Q^2 the variance of the quantisation error. We may rewrite this as

$$\mathrm{SNR}_O = \frac{\sigma_x^2}{\sigma_E^2}\frac{\sigma_E^2}{\sigma_Q^2} = G_P \mathrm{SNR}_Q$$

where σ_E^2 is the variance of the prediction error. The quantity G_p, called the *processing gain* then represents the gain in signal to noise ratio due to the differential quantisation scheme. For a given baseband signal σ_x^2 is fixed so G_p is maximised by minimising the variance σ_E^2 of the predictor. It can be shown that minimising σ_E^2 with respect to the predictor coefficients $\{a_i\}$ results in a set of linear equations called the *normal* or *Yule-Walker* equations (see Proakis p.128)

$$\sum_{i=1}^{p} a_i \phi(i-j) = \phi(j),\ j = 1, 2, \ldots, p$$

where $\phi(n)$ is the autocorrelation function of the sampled signal sequence x_n. It may be estimated from the finite set of samples $\{x_n\}$ using the relation

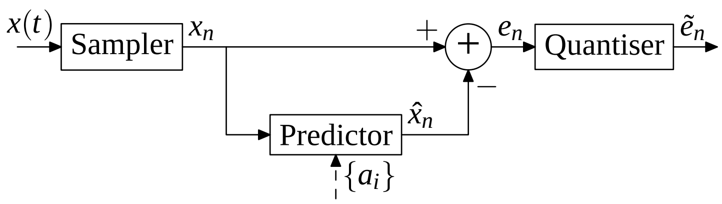

Figure 67: DPCM Encoding

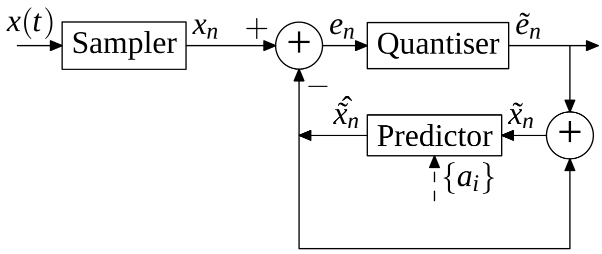

Figure 69: A Practical DPCM Encoder

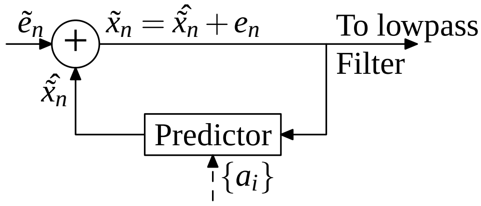

Figure 70: A Practical DPCM Decoder

$$\hat{\phi}(n) = \frac{1}{N}\sum_{i=1}^{N-n} x_i x_i + n,\ n = 0, 1, \ldots, p$$

In the DPCM encoder as demonstrated the difference between the transmitted sample and the received sample is given by

$$\begin{aligned} \tilde{x}_n - x_n &= \left(\tilde{e_n} + \sum_{i=1}^{p} a_i \tilde{x}_{n-i}\right) - \left(e_n + \sum_{i=1}^{p} a_i x_{n-i}\right) \\ &= q_n + \sum_{i=1}^{p} a_i(\tilde{x}_{n-i} - x_{n-i}) \end{aligned}$$

where $q_n = \tilde{e_n} - e_n$ is the quantisation error i.e. there is an accumulation of quantisation errors at the receiver.

Practical DPCM Shown below are block diagrams for the encoder and decoder of a practical DPCM system which overcomes the problem of accumulating quantisation error associated with the simplistic DPCM discussed previously.

In this implementation (see figure 69)) the predictor is implemented with the feedback loop around the quantiser. The input to the predictor $\tilde{x_n}$ is the sample signal x_n modified by the quantisation process and the output of the predictor is given by

$$\hat{\tilde{x}}_n = \sum_{i=1}^{p} a_i \tilde{x}_{n-i}$$

The difference

$$e_n = x_n - \hat{\tilde{x}}_n$$

is the input to the quantiser and $\tilde{e_n}$ the output which is encoded to binary digits and transmitted to the destination. The quantised error $\tilde{e_n}$ is also added to the predicted value $\hat{\tilde{x}}_n$ to yield $\tilde{x_n}$.

At the destination the same predictor is used and its output $\hat{\tilde{x}}_n$ added to the incoming error $\tilde{e_n}$ to yield the output quantised signal $\tilde{x_n}$ which is used by the predictor and which after filtering provides the output signal $x(t)$.

Using feedback around the quantiser ensures that the error in $\tilde{x_n}$ is simply the quantisation error $q_n = x_n - \hat{\tilde{x}}_n$ and there is no accumulation of previous quantisation errors in the implementation of the decoder.

The difference between received sample and transmitted sample is given by

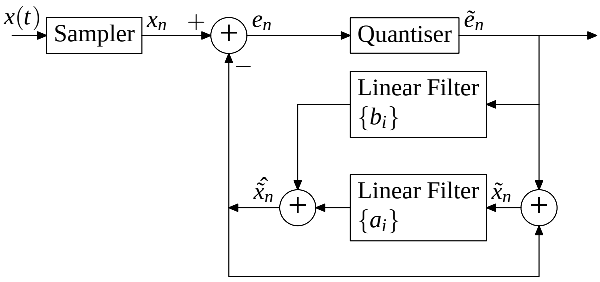

Figure 71: DPCM Encoder Using a linearly filtered error sequence

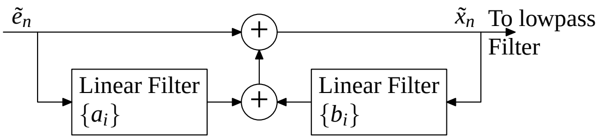

Figure 72: DPCM Decoder Using a linearly filtered error sequence

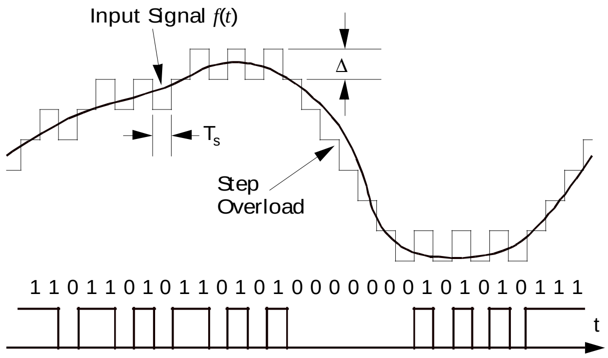

Figure 73: Delta Modulation

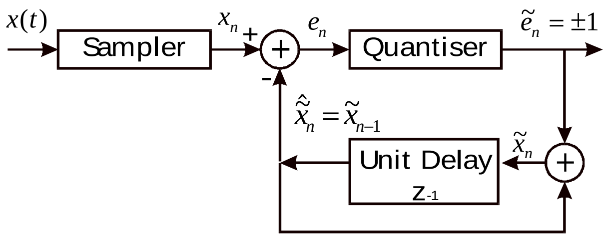

Figure 74: Delta Modulation Encoder

$$
\begin{aligned}
\tilde{x}_n - x_n &= \left(\tilde{e_n} + \sum_{i=1}^{p} a_i \tilde{x}_{n-i}\right) - \left(e_n + \sum_{i=1}^{p} a_i \tilde{x}_{n-i}\right) \\
&= q_n
\end{aligned}
$$

i.e. it is just the quantisation error and there is no accumulation of errors at the receiver.

An improvement in the quality of the estimate used in DPCM can be obtained by including linearly filtered past values of the quantised error. The $\hat{\tilde{x}}_n$ estimate may be expressed as

$$\hat{\tilde{x}}_n = \sum_{i=1}^{p} a_i \tilde{x}_{n-i} + \sum_{i=1}^{m} b_i \tilde{e}_{n-i}$$

where $\{b_i\}$ are the coefficients of the filter for the quantised error sequence. Both sets of coefficients $\{a_i\}$ and $\{b_i\}$ are chosen to minimise some function of the error $e_n = x_n - \hat{\tilde{x}}_n$ such as the mean square error.

This may be thought of in two ways. Firstly since the DPCM processing gain is proportional to $1/\sigma_E^2$ it is improved by minimising the variance in the transmitted errors and one way of doing that is to filter these signals. Alternatively and equivalently we could consider this as using a second predictor on the error signals to further improve our estimates of the signal.

Delta Modulation Delta modulation may be viewed as a simplified form of DPCM in which a 1-bit quantiser is used with a first order predictor. This is illustrated in figure 73) where a positive pulse (1) is transmitted if the signal is increasing and a 0 level is transmitted if the signal is decreasing.

Block diagrams of an encode and decoder are in figures 74) and 75). We note that

$$\hat{\tilde{x}}_n = \tilde{x_{n-1}} = \hat{\tilde{x}}_{n-1} + \tilde{e_{n-1}}$$

and since

$$q_n = \tilde{e_n} - e_n = \tilde{e_n} - \left(x_n - \hat{\tilde{x}}_n\right)$$

it follows that

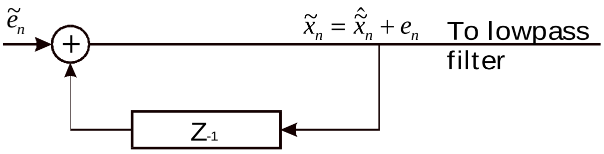

Figure 75: Delta Modulation Decoder

$$\hat{\hat{x}}_n = x_{n-1} + q_{n-1}$$

Thus the estimated (predicted) value of x_n is really the previous sample x_{n-1} modified by the quantisation noise q_{n-1}. In general the quantised error signal is scaled by some value by a *step size* Δ_1. In effect the encoder approximates a waveform $x(t)$ by a linear staircase function. For the approximation to be relatively good the waveform must change slowly relative to the sampling rate i.e. the sampling rate must be several times (e.g. 5) the Nyquist rate.

The quantiser input in delta modulation is an approximation to the derivative of the incoming message signal. Therefore disturbances such as noise result in an accumulative error in the demodulated signal. This drawback is overcome by integrating prior to delta modulation. This pre-emphasises the low frequency content of the input signal, increases the correlation between adjacent samples (thus reducing the value of the variance signal) and simplifies the design of the receiver which need only by an integrator. This is called *delta-sigma modulation* . A simple practical circuit implementing this is shown in figure 76). The circuit is based on a simple RC integrator. It will function if the output of the flip-flop is V volts (e.g. a CMOS device). The voltage across the capacitor will be a series of positive and negative exponential decays. The error voltage is the difference between the input voltage and the output voltage which is approximately zero when the input signal is zero. The receiver can simply be a simple RC integrator as shown in figure 76).

At any given sampling rate the performance of the DM encoder is limited by two types of distortion as shown in the figures 77) and 78).

Slope-overload Distortion occurs when the step size Δ_1 is too small to follow portions of

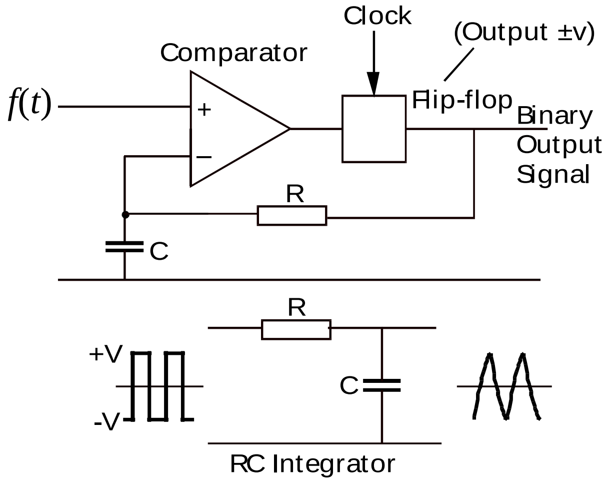

Figure 76: A Simple Delta Modulation Encoder and Decoder

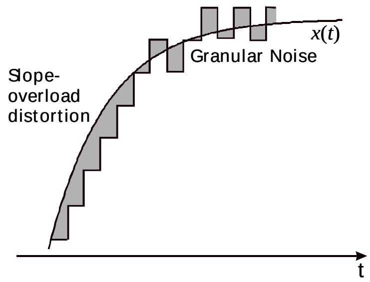

Figure 77: Delta Modulation Distortion: Fixed Step Size

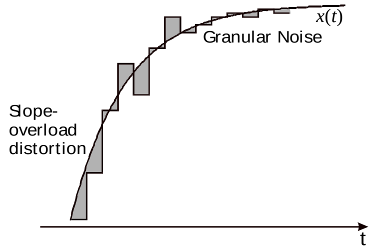

Figure 78: Delta Modulation Distortion: Variable Step Size

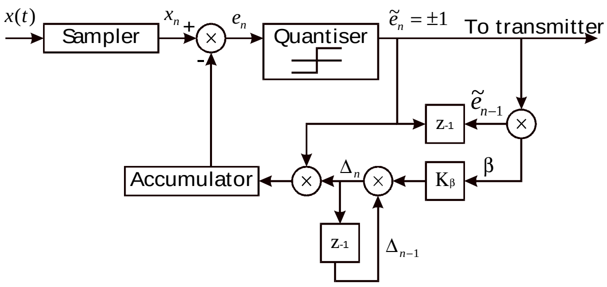

Figure 79: CSVD Encoder

the waveform that have a steep slope and the encoder output falls behind the input signal.

Granular noise results from using a step size that is too large in parts of the waveform having a small slope.

These two distortions place contradictory requirements on the step size of the encoder and a solution is to use a variable step size that adapts itself to the short term characteristic of the signal source i.e. it the step size is increased where the waveform has a steep slope and is reduced where it has a small slope.

A number of methods may be used to adaptively set the step size. The quantised error $\tilde{e_n}$ provides a good indication of the waveform slope characteristics. When successive values of $\tilde{e_n}$ are changing signs then the slope is small, when they have the same sign then the slope is large. A relatively simple rule to adaptively vary the step size according to the relation

$$\Delta_n = \Delta_{n-1} K^{\tilde{e_n} . \tilde{e_{n-1}}}$$

where $K \geq 1$ is a constant selected to minimise the total distortion.

A particularly popular technique is called *continuously variable slope modulation* (CVSD) in which the adaptive step size parameter is expressed as

$$\Delta_n = \alpha \Delta_{n-1} + k_1$$

if $\tilde{e_n}$, $\tilde{e_{n-1}}$ and $\tilde{e_{n-2}}$ have the same sign; otherwise

$$\Delta_n = \alpha \Delta_{n-1} + k_2$$

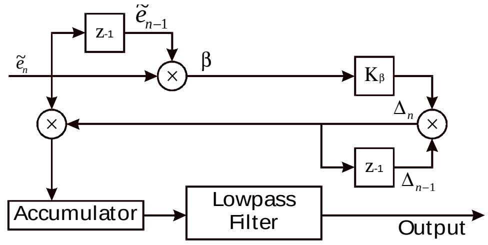

Figure 80: CSVD Decoder

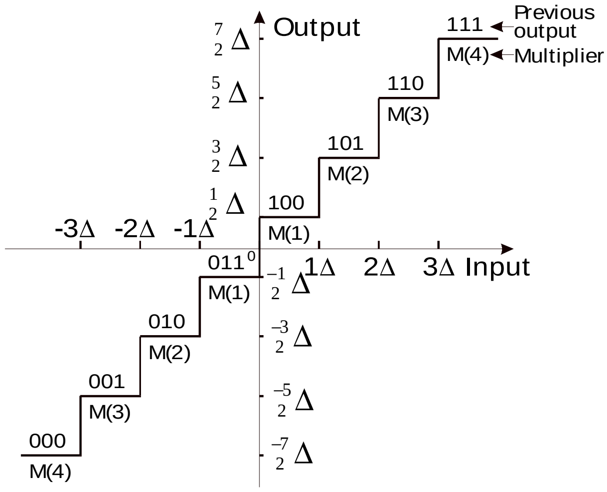

Figure 81: Adaptive Quantisation

where the parameters are chosen such that $0 < \alpha < 1$ and $k_1 \gg k_2 > 0$. Block diagrams representing the decoder and encoder for performing this type of modulation are illustrated in figures 79) and 80)

Adaptive Quantisation

Most real sources are quasi-stationary in nature i.e. their variance and autocorrelation function's vary slowly with time. PCM and DPCM encoders are designed on the basis that the output is stationary. Their efficiency and performance can be improved by having them adapt to the slow time-variant statistics of the source.

With both PCM and DPCM the quantisation error q_n operating on a quasi-stationary input signal will have a time-variant variance (quantisation

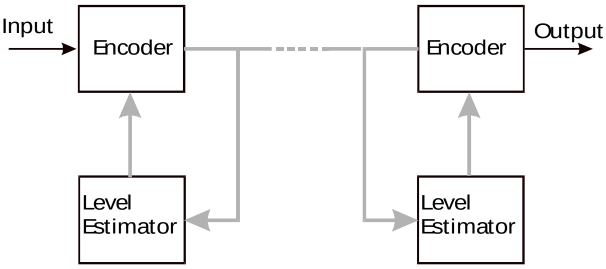

Figure 82: Adaptive Quantisation with backward estimation

noise power). An improvement that reduces the dynamic range of the quantisation noise is the use of an adaptive quantiser. A relatively simple method is to use a uniform quantiser that varies its step size according with the variance of the past signal samples. Is its simplest form only one previous signal sample may be used for the step size adjustment. Shown in figure 81) is an example of a 3-bit quantiser whose step size varies recursively with the relation $\Delta_{n+1} = \Delta_n M(n)$ where $M(n)$ is a factor whose value depends on the quantiser level for the sample x_n and Δ_n is the step size of the quantiser for sample x_n. The values of $M(n)$ optimised for speech are given in the table 1).

More generally we have a step size proportional to an estimate of the variance of the signal

$$\Delta_n = k\hat{\sigma}(x)$$

The problem is one of computing the estimated variance $\widehat{sigma}(x)$ continuously. This we may do in two ways

AQF Adaptive quantisation with forward estimation in which unquantised samples of the input signal are used to derive forward estimates of $\hat{\sigma}(x)$

AQB Adaptive quantisation with backward estimation in which samples of the quantised output are used to derive backward estimates of $\widehat{sigma}(x)$

AQF requires the use of a buffer to store unquantised samples and requires explicit transmission of level information to a remote decoder. It also introduces a delay (about 16ms for speech) in the encoding process which may not be acceptable. These problems are all avoided by AQB which is therefore usually preferred. It represents a non-linear feedback system which can be shown to be stable if the input is bounded.

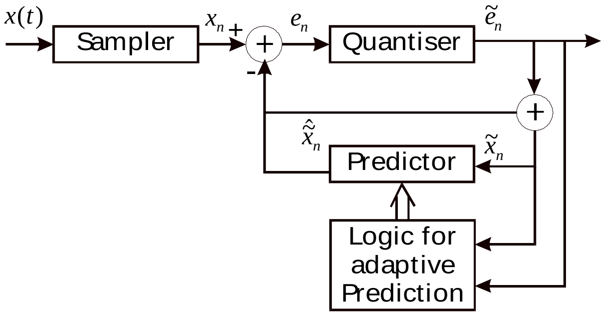

Figure 83: Adaptive Prediction with backward estimation (APB)

Adaptive Prediction

In DPCM the predictor can also be made adaptive when the source output is quasi-stationary i.e. the predictor coefficients can be changed periodically with time. They are calculated from the short term estimate of the autocorrelation function of x_n and transmitted along with the quantised error $\tilde{e_n}$ to the receiver which implements the same predictor.

Analogous to adaptive quantisation we may use forward or backward estimations of the autocorrelation functions of x_n

APF Adaptive prediction with forward estimation in which unquantised samples of the input signal are used to derive forward estimates of $\phi(n)$

AQB Adaptive prediction with backward estimation in which samples of the quantised output are used to derive backward estimates of $\phi(n)$

Transmission of the predictor offsets in part the lower data rate achieved by the reduction in the number of bits needed to transmit the lower dynamic range in the error e_n resulting from adaptive predication so as for adaptive quantisation the backward estimation AQB is preferred

In this case the receiver predictor may compute its own predictor coefficients from $\tilde{e_n}$ and $\tilde{x_n}$ where

Table 1: Values of $M(n)$ for an adaptive quantiser optimised for speech

	PCM			DPCM		
	2	3	4	2	3	4
M(1)	0.60	0.85	0.80	0.80	0.90	0.90
M(2)	2.20	1.00	0.80	1.60	0.90	0.90
M(3)		1.00	0.80		1.25	0.90
M(4)		1.5	0.80		1.70	0.90
M(5)			1.20			1.20
M(6)			1.60			1.60
M(7)			2.00			2.00
M(8)			2.40			2.40

$$\tilde{x_n} = \tilde{e_n} + \sum_{i=1}^{p} a_i \tilde{x_{n-i}}$$

If we neglect quantisation noise $\tilde{x_n}$ is equivalent to x_n and may be used to estimate the autocorrelation function $\phi(n)$ at the receiver. If the quantisation error is sufficiently small this is adequate for determining the predictor coefficients and the adaptive predictor results in a reduced source data rate.

Instead of using the block processing approach for determining the predictor coefficients we can do it on a sample by sample basis. Similar schemes may also be used for adapting the filter coefficients $\{a_i\}$ and $\{b_i\}$ for the second DPCM algorithm mentioned previously.

Adaptive Differential Pulse-Code modulation (ADPCM) using both adaptive quantisation and adaptive prediction is now internationally accepted for speech encoded transmissions at 32 kb/s along with standard PCM at 64 kb/s.

Assessment
Deadline: 2009-10-31 23:59:00
No Questions: 1
Time Allowed: 2 min

Adaptive Subband Coding

The coding techniques discussed so far digitally represent the temporal characteristics of the source waveform. The are however two other basic classes of source coding. In *spectral waveform encoding* the signal waveform is usually subdivided into different frequency bands and either the time waveform in each band or its spectral characteristics are encoded for transmission. In *Model-based encoding* the source is modelled as a filter that when excited by an appropriate input signal results in the observed source output. The parameters of the filter together with an appropriate excitation signal are transmitted and provided the number of parameters are sufficiently small a large reduction in data is achieved.

One form of spectral waveform encoding is *adaptive subband coding* (ASBC) in which the signal is divided into a number of frequency subbands each of which is coded separately. This technique can digitise speech at 16 kb/s with a quality comparable to 64 kb/s PCM. To achieve this it exploits the quasi-periodic nature of voiced speech which manifests itself in the fact that people speak with a characteristic pitch frequency. This permits pitch prediction reducing the level of prediction error requiring quantisation and thus greatly reducing the number of bits per sample. Further reductions are achieved by exploiting the fact that the human ear cannot hear noise below about 15dB below the signal level in that particular band

In ASBC noise shaping is accomplished bit adaptive bit assignment. The number of bits used to encode each subband is varied dynamically and shared with other subbands. Subbands with little or no energy may not be encoded at all. A block diagram of this is shown in figure 84).

The signal is split into a number (4 or 8) of contiguous bands by a bank of bandpass filters (typically each covering an octave) and these are fre-

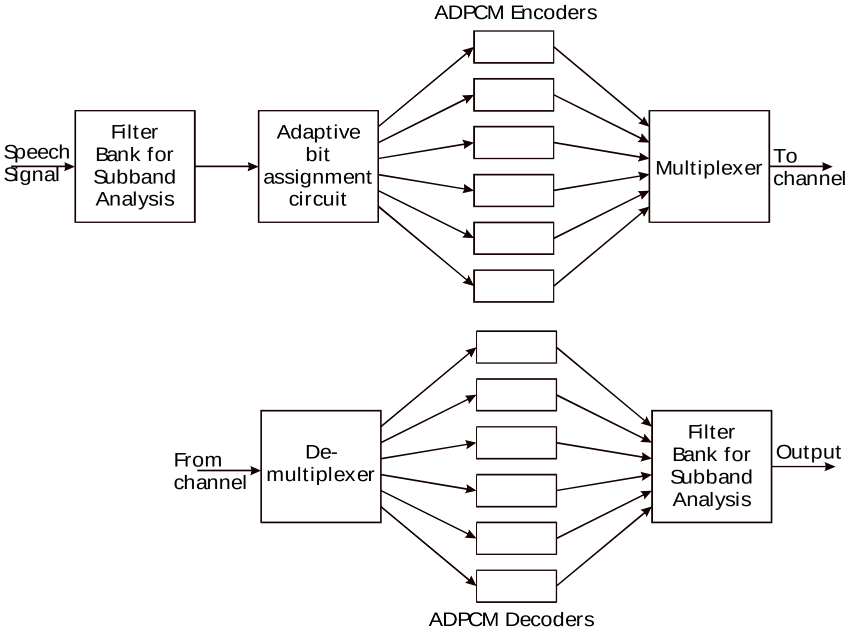

Figure 84: Adaptive Subband Transmitter and Receiver

quency shifted using single sideband modulation to a low-pass form before sampling, band specific ADPCM encoding and multiplexing. into a bit stream. The bit assignment information is transmitted to the receiver allowing it to decode the subbands individually and modulate them back into their appropriate frequencies which are summed to create the output signal. The complexity of the adaptive subband encoder means that there is a processing delay of about 25ms but this is no concern in voice storage e.g. voice mail.

Digital Baseband Transmission

Shown in figure 85) are the principle components of a digital baseband transmission system. We take an input binary data sequence b_k which is applied to a pulse amplitude modulator to produce a sequence of pulses a_k. These pulses are shaped prior to transmission using a transmit filter $g(t)$ and the resultant signal $s(t)$ transmitted over the transmission channel. The transmission channel has a characteristic impulse response $h(t)$ which will modify the shape of the pulses as they are transmitted and an additive source of noise $n(t)$. The received signal, modified by the channel, $x(t)$ is filtered using the receive filter with an impulse response $c(t)$ and the resulting signal $y(t)$ is sampled at integer multiples of the pulse duration. The samples are compared with a decision threshold and a decision made to determine if a 1 or 0 was transmitted.

In this tutorial we will be examining two channel effects, its response and additive noise separately. We will see that the channel bandwidth broadens pulses so they may overlap introducing intersymbol interference and limiting the maximum transmission rate and noise introduces errors into the received signal. In baseband systems, it is generally the case that the channel bandwidth is the dominant limitation.

Noise in Baseband Systems

The Matched Filter

Our problem is to design a receiver response to minimise the effects of noise. We model the receiver as a linear time-invariant filter of impulse response $h(t)$ followed by a sampler taking samples at intervals T_b as shown in figure 86). The known original pulse signal is $s(t)$ and the output

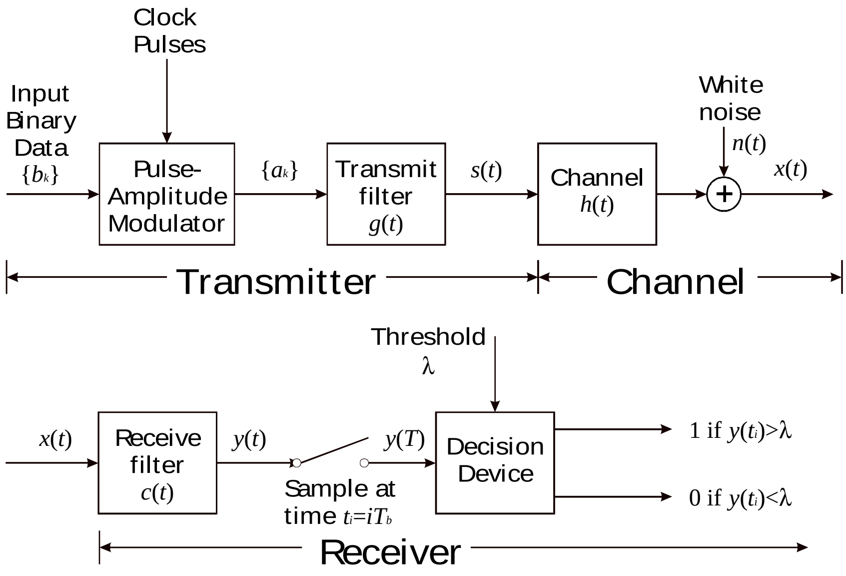

Figure 85: Components of a Digital baseband System

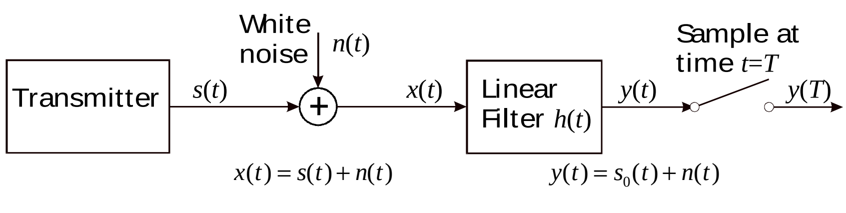

Figure 86: Model of a matched filter baseband transmission system

of the channel is $x(t) = s(t) + n(t)$ where $n(t)$ is the additive noise of zero mean and power spectral density $N_0/2$. We wish to determine the optimum filter response. Since the filter is linear its output may be represented as $y(t) = s_o(t) + n(t)$ where $s_o(t)$ and $n(t)$ are the signal and noise components respectively. We wish to maximise the instantaneous output signal component measured at time $t = T$ compared with the average power of the output noise $n(t)$. Assuming we can synchronise the receiver with the incoming data and take the samples at the optimum moment we chose the filter response to maximise the *peak pulse signal-to-noise ratio* defined as

$$\eta = \frac{|s_o(T)|^2}{E[n^2(t)]}$$

Signal Component We denote the Fourier transforms of the known signal and filter as $S(f)$ and $H(f)$ respectively. The Fourier Transform of the output signal from the filter $s_o(t)$ will be $S(f)H(f)$ so we have

$$s_o(t) = \int_{-\infty}^{\infty} H(f)S(f)e^{i2\pi ft}df$$

Hence when the output is sampled at $t = T$ we have for the signal component

$$|s_o(t)|^2 = \left|\int_{-\infty}^{\infty} H(f)S(f)e^{i2\pi fT}df\right|^2$$

Noise Component The power spectral density of the output noise from the filter is the power spectral density of the input noise $N_0/2$ times the squared magnitude of the filter transfer function $H(f)$

$$S_N(f) = \frac{N_0}{2}|H(f)|^2$$

The average power is therefore

$$E\left[n^2(t)\right] = \frac{N_0}{2}\int_{-\infty}^{\infty}|H(f)|^2df$$

We thus have for the peak pulse signal-to-noise

$$\eta = \frac{\left|\int_{-\infty}^{\infty} H(f)S(f)e^{i2\pi fT}df\right|^2}{\frac{N_0}{2}\int_{-\infty}^{\infty}|H(f)|^2df}$$

To find the particular form of $H(f)$ which maximises this expression we apply Schwarz's Inequality (Section) to the numerator of this which may then be rewritten as

$$\left|\int_{-\infty}^{\infty} H(f)S(f)e^{i2\pi fT}df\right|^2 \leq \int_{-\infty}^{\infty}|H(f)|^2df \int_{-\infty}^{\infty}|S(f)|^2df$$

We may thus redefine the peak pulse signal to noise as

$$\eta \leq \frac{2}{N_0}\int_{-\infty}^{\infty}|S(f)|^2df$$

This depends only on the signal energy and noise power and not on the filter transfer function. Thus η will be a maximum when the filter transfer function is chosen so that the equality holds and we have

$$H_{opt}(f) = kS^*(f)e^{-i2\pi fT}$$

Taking Fourier transforms of this equation gives the time domain relation

$$h_{opt}(t) = ks(T-t)$$

> **Important**
>
> The impulse of the optimum *matched filter* is a time reversed and delayed version of the input signal except for an arbitrary scaling factor.

We have for the optimum impulse response of a matched filter the expression $h_{opt}(t) = ks(T-t)$ i.e. it is uniquely defined except for an arbitrary delay and scaling factor by the waveform of the pulse signal $s(t)$. The most important result we wish to know is what the peak pulse signal-to-noise ratio is in this case. To calculate this we consider a filter matched to a known signal $s(t)$. The Fourier transform of the matched filter signal output is given by

$$\begin{aligned} S_o(f) &= H_{opt}(f)S(f) \\ &= kS^*(f)S(f)e^{-i2\pi fT} \\ &= k|S(f)|^2e^{-i2\pi fT} \end{aligned}$$

Taking the Fourier transform of this we get

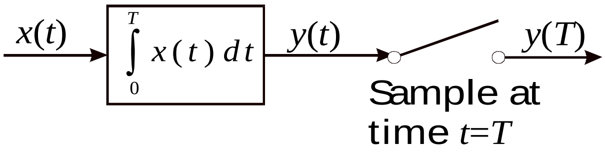

Figure 87: Integrate and Dump Filter

$$
\begin{aligned}
s_o(T) &= k\int_{-\infty}^{\infty} S_o(f)e^{i2\pi fT}df \\
&= k\int_{-\infty}^{\infty} |S(f)|^2 df \\
&= kE
\end{aligned}
$$

where E is the energy of the pulse signal. Similarly we may substitute out matched filter result into the expression for the noise power to get

$$
\begin{aligned}
E\left[n^2[t]\right] &= \frac{N_0}{2}\int_{-\infty}^{\infty} |H(f)|^2 df \\
&= \frac{k^2 N_0}{2}\int_{-\infty}^{\infty} |S(f)|^2 df \\
&= \frac{k^2 N_0 E}{2}
\end{aligned}
$$

and hence the peak pulse signal-to-noise ratio becomes

$$\eta_{max} = \frac{(kE)^2}{(k^2 N_0 E/2)} = \frac{2E}{N_0}$$

The peak pulse signal-to-noise ratio of a matched filter depends only on the ratio of the signal energy to the power spectral density of the white noise at the filter input i.e.

Important

All signals with equal energy are equally effective to combat additive white Gaussian noise with a matched filter.

As an example we show (figure 88)) the matched filter response for a rectangular pulse of amplitude A and duration T. In this case the impulse response

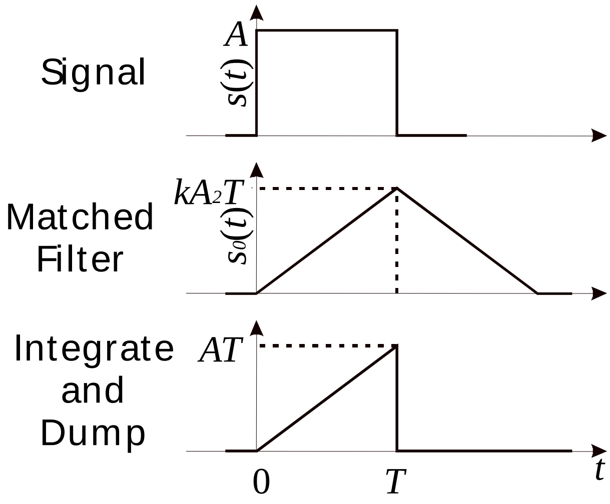

Figure 88: Matched Filter for a Rectangular Pulse

of the filter is exactly that of the waveform itself and the output is a triangle wave of amplitude kAT. With a maximum at $t = T$. This special case can be implemented using the *integrate-and-dump* circuit (figure 87)) which for $0 \le t \le T$ has the same waveform at the output as the ideal matched filter.

Schwarz's Inequality Given two complex functions $\phi_1(x)$, $\phi_2(x)$ of a real variable x which have finite energy i.e.

$$\int_{-\infty}^{\infty} |\phi_i(x)|^2 dx < \infty$$

then we may write

$$\left|\int_{-\infty}^{\infty} \phi_1(x)\phi_2(x)dx\right|^2 \le \int_{-\infty}^{\infty} |\phi_1(x)|^2 dx \int_{-\infty}^{\infty} |\phi_2(x)|^2 dx$$

The equality in this expression only holds if we have

$$\phi_1(x) = k\phi_2^*(x)$$

where k is an arbitrary constant.

Gaussian Noise Processes

The simplest mathematical model for a communication system is the additive noise channel where

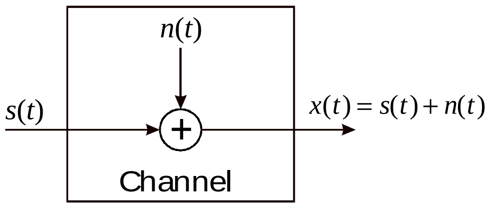

Figure 89: Additive Noise Channel

the transmitted signal $s(t)$ is corrupted by an additive white noise process $n(t)$. This is the most random noise process in that there is no correlation between samples. Physically the noise may arise from electronic components and amplifiers at the receiver of the communication system or from interference encountered in transmission (e.g. radio interference). If the noise primarily arises from electronic components it may be characterised as thermal noise which statistically is characterised as a *Gaussian noise process* . Hence the resulting model is called the *Additive Gaussian noise channel* .

This may be viewed as the result of the statistical central limit theorem which states that the distribution of a large number of identical random variables (i.e. electrons under the influence of thermal vibrations) will tend to Gaussian as the number of variables tends to infinity.

The probability distribution function of a Gaussian or normally distributed random variable is

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma_X} \exp\left(-\frac{(z-\mu_X)^2}{2\sigma_X^2}\right)$$

where μ_X is the mean and σ_X^2 the variance of the random variable X.

The cumulative distribution function is therefore given by

$$\begin{aligned} F(x) &= \int_{-\infty}^{x} p(u)du \\ &= \frac{1}{2}\left(1 + \text{erf}\left(\frac{x-\mu_X}{\sqrt{2}\sigma_X}\right)\right) \\ &= 1 - Q\left(\frac{x-\mu_X}{\sigma_X}\right) \end{aligned}$$

It is common to use the Q function, as defined above, when dealing with communications systems, and this is plotted below.

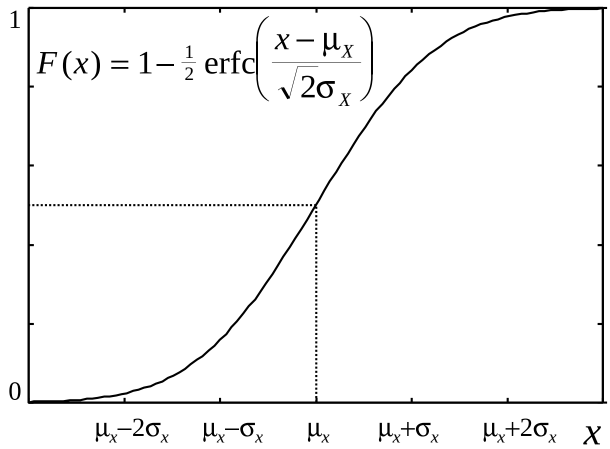

Figure 91: Cumulative Gaussian Probability Distribution

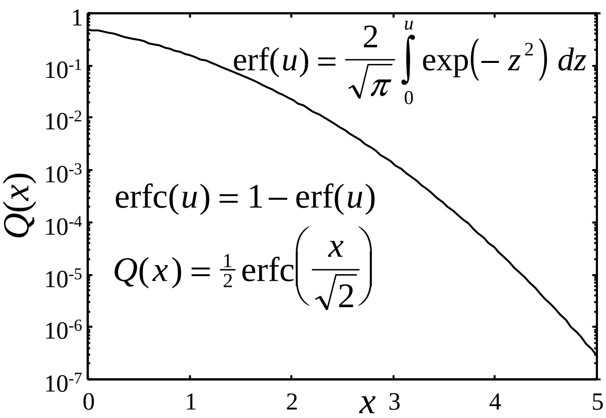

Figure 92: Plot of Q(x) versus SNR

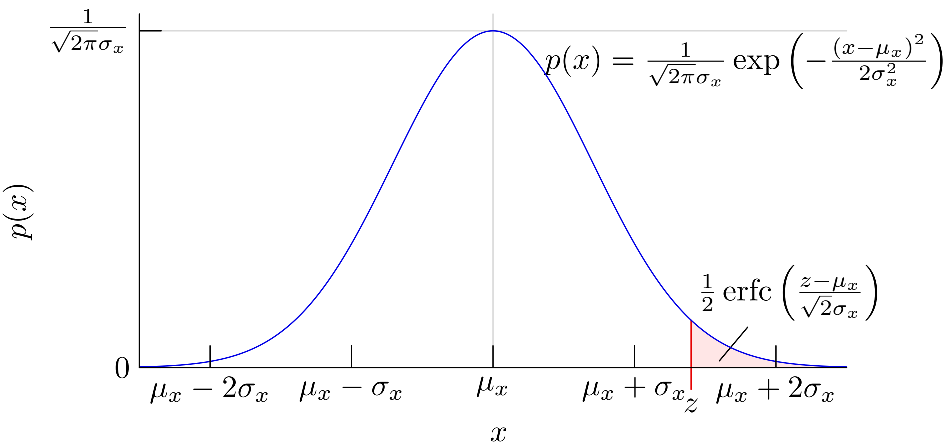

Figure 90: The Gaussian Probability Distribution Function

Other random distributions may be used to model other processes for example the Rayleigh and Rice distributions are used to model the fluctuations of signals received from a multiple path fading channel.

White Noise White noise is noise in which the power spectral density is independent of the operating frequency This is commonly used in the noise analysis of communication systems. The power spectral density of white noise is expressed as

$$S_W(f) = \frac{N_0}{2}$$

The parameter N_0 (W/Hz) is usually referenced to the input stage of a receiver and may be expressed as

$$N_0 = kT_e$$

where k is Boltzmann's constant and T_e is the equivalent noise temperature of the receiver. The equivalent noise temperature of a system is defined as the temperature at which a noisy resistor has to be maintained such that, by connecting the resistor to the input of a noiseless version of the system, it produces the same available noise power at the output of the system as that produced by all the sources of noise in the actual system. It depends only on the parameters of the system.

Since the autocorrelation function is the inverse Fourier transform of the power spectral density then for white noise

$$R_W(\tau) = \frac{N_0}{2}\delta(\tau)$$

Note that since this is zero for $\tau \neq 0$ any two samples of white noise no matter how closely together in time they are taken are uncorrelated. If the white noise is also Gaussian then the two samples are also statistically independent. Gaussian white noise may therefore considered to be the most random distribution.

Since white noise has infinite bandwidth it also has infinite average power and is therefore not physically realisable. However provided the bandwidth of the noise at the input of a system is significantly larger than that of the system itself we may model the noise process as white noise. Calibrated white noise may be used to characterise the spectral response of a system.

The Error Rate Due To Noise

We seek to determine the probability of error in a binary PCM system based on *non-return-to-zero (NRZ) signalling* due to additive white Gaussian noise of zero mean and power spectral density $N_0/2$. For this analysis the transmitted signal consists of pulses of constant A or $-A$ amplitude units and T_b seconds in duration.

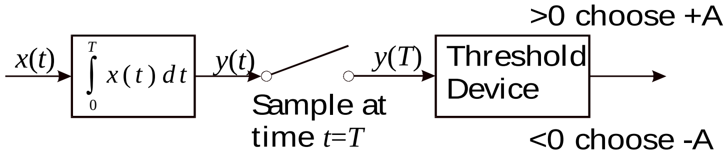

Figure 93: Baseband receiver using an integrate and dump filter

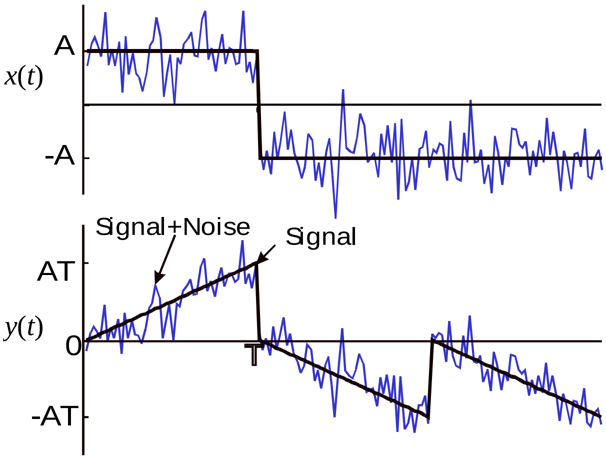

Figure 94: The effect of the integrate and dump filter on signal and noise.

The receiver used consists of a matched (integrate and dump) filter (see The Matched Filter (Section)) and a threshold decision device as shown in figure 93). The matched filter output is sampled at the end of each signalling interval. Over the sampling interval T_b the signal is integrated so that the signal accumulates while the variance of the random noise σ_Y^2 will decrease. This effect is illustrated in figure 94).

The variance is then given by

$$\sigma_T^2 = \frac{N_0}{2T_b} \tag{7}$$

See Haykin p. 420 for the proof of this.

In figure 95) we can plot a graph showing the probability distribution function versus amplitude for the two signals. We can see that it consists of two Gaussian distributions separated by $2A$ and with variances as given in equation 7).

We denote the conditional probability of error given that symbol 0 was sent as P_{e0} and given that we set the decision threshold midway between the two levels at zero this is given by the labelled shaded area under that tail of the Gaussian distribution as shown in figure 95).

$$P_{e0} = \frac{1}{\sqrt{\pi N_0/T_b}} \int_0^\infty \exp\left(-\frac{(y+A)^2}{N_0/T_b}\right) dy$$

By defining a new variable

$$z = \frac{y+A}{\sqrt{N_0/T_b}}$$

this may be rewritten as

$$P_{e0} = \frac{1}{\sqrt{\pi}} \int_{\sqrt{E_b/N_0}}^\infty e^{-z^2} dz$$

where E_b is the transmitted energy per bit given by

$$E_b = A^2 T_b$$

Thus P_{e0} may be given in terms of the complementary error function as

$$P_{e0} = \frac{1}{2}\text{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right)$$

We may perform exactly the same calculation for the conditional probability of error provided a 1 is sent P_{e1}. In this case of a binary symmetric channel with the threshold exactly half way between the two signal levels it will be identical to P_{e0}. The average probabilityof symbol error $P(_e$ for the channel will depend on the a priori probabilities of binary symbols 0 and 1, p_0 and p_1 which if we assume are equiprobable $p_0 = p_1 = 1/2$ we get

$$\begin{aligned} P_e &= p_0 P_{e0} + p_1 P_{e1} \\ &= \frac{1}{2}\text{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right) \end{aligned}$$

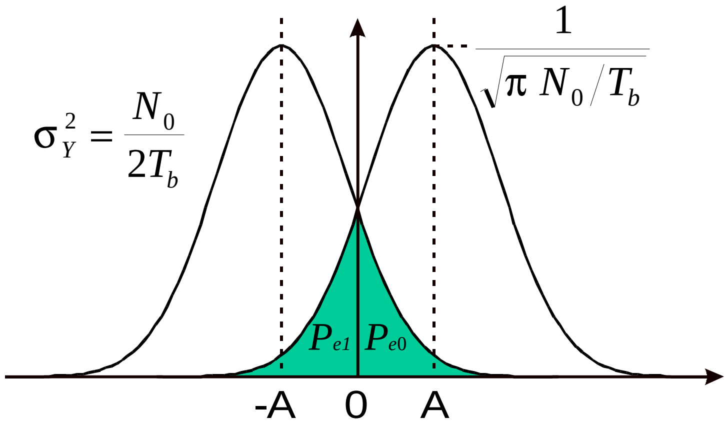

Figure 95: Signal space diagram for the bipolar NRZ system

or in terms of the more usually used Q function

$$P_e = Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$$

which is plotted in figure 96)

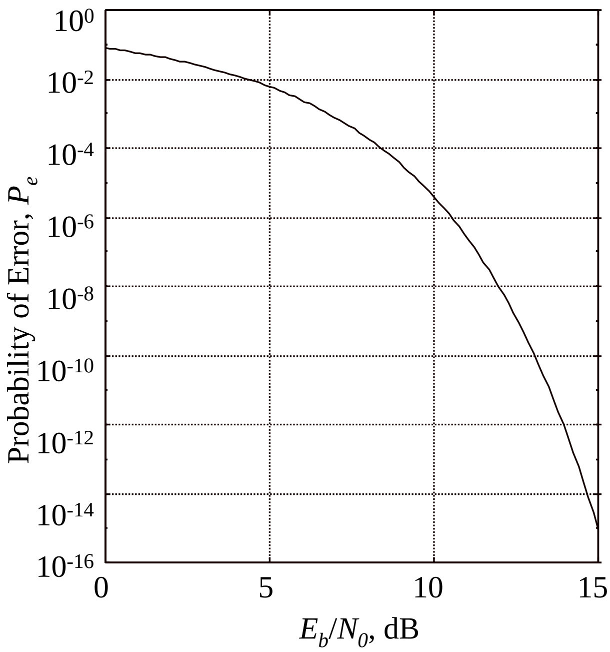

Figure 96: Probability of Error vs SNR for bipolar NRZ

Assessment
Deadline: 2009-11-14 23:59:00
No Questions: 3
Time Allowed: 10 min

Complementary Error Function Tables

Intersymbol Interference

The main source of errors in most baseband communication systems is intersymbol interference arising from the *dispersive* nature of the communications channel i.e. pulse distortion arising from the non-ideal filtering characteristics of the transmission channel leading to interference between symbols. In baseband transmission discrete pulse-amplitude modulation (PAM) is the most efficient modulation scheme and for the analysis carried out here we will be looking at binary systems. The analysis can easily be generalised for M-ary data transmission.

Shown in figure 98) is the generic model of the system which we will be considering. An incoming binary sequence $\{b_k\}$ of symbols 0 and 1 each of

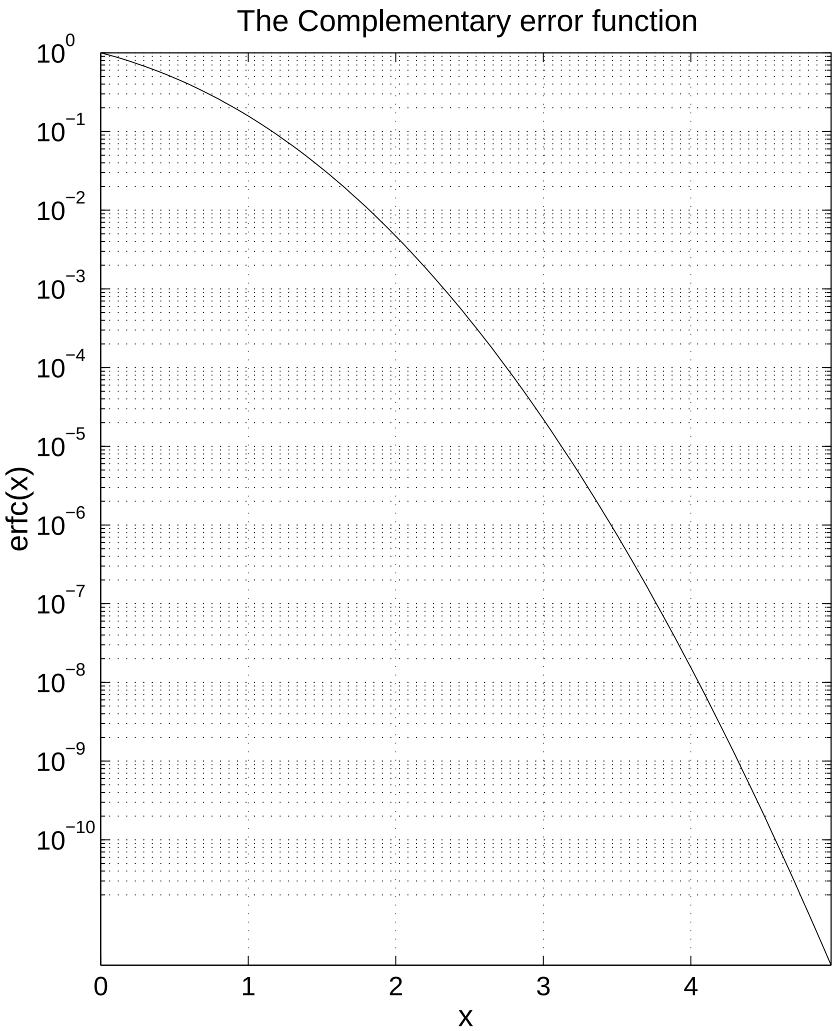

Figure 97: Graph of the complementary error function erfc

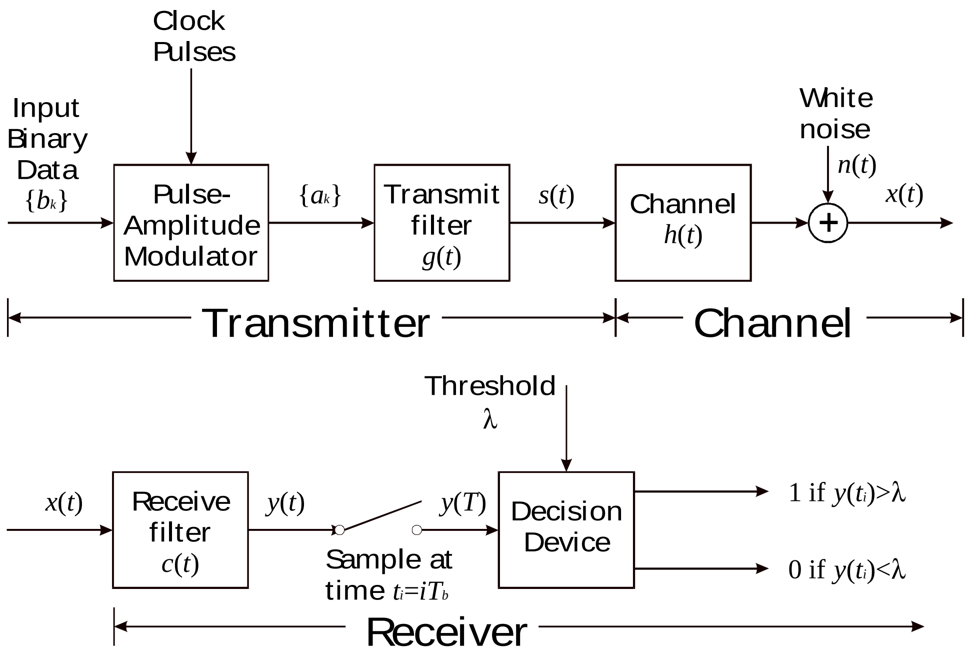

Figure 98: Generic model of a baseband transmission system

duration T_b is transformed by the pulse amplitude modulator into a sequence of short pulses $\{a_k\}$ of amplitudes +1 and -1. These short pulses are applied to a *transmit filter* of impulse response $g(t)$ producing the transmitted signal

$$s(t) = \sum_k a_k g(t - kT_b)$$

The signal $s(t)$ is modified by transmission through a channel of impulse response $h(t)$ and also has some noise $n(t)$ added to it. At the receiver the noisy signal $x(t)$ is passed through a receiver filter $c(t)$ and the resulting filtered output $y(t)$ is synchronously sampled at intervals of T_b and the samples are passed to a threshold device which outputs a 1 if the sample is greater than the threshold λ and 0 if less.

The receive filter output can be written in the form

$$y(t) = \mu \sum_k a_k p(t - kT_b) + n(t)$$

where μ is an arbitrary scaling factor taking account of amplitude changes through the transmission process and $p(t)$ the received pulse will given by the convolution of the pulse shape, the channel response and the receive filter:

$$\mu p(t) = g(t) * h(t) * c(t)$$

where we can normalise $p(t)$ such that $p(0) = 1$

The receive filter output is sampled at time intervals $t_i = iT_b$ giving

$$\begin{aligned} y(t_i) &= \sum_{k=-\infty}^{\infty} a_k p((i-k)T_b) + n(t_i) \\ &= \mu a_i + \mu \sum_{k=-\infty, k\neq i}^{\infty} a_k p((i-k)T_b) + n(t_i) \end{aligned} \tag{8}$$

where μa_i represents the contribution of the i^{th} transmitted bit. The second term represents the residual effect of all the r transmitted bits on decoding the i^{th} bit - the *intersymbol interference* . The last term represents the noise signal at time t_i .

x	erfc(x)	x	erfc(x)	x	erfc(x)	x	erfc(x)
0	1	1.25	0.0771	2.5	0.00040695	3.75	1.1373e-07
0.05	0.94363	1.3	0.065992	2.55	0.00031066	3.8	7.7004e-08
0.1	0.88754	1.35	0.056238	2.6	0.00023603	3.85	5.1886e-08
0.15	0.832	1.4	0.047715	2.65	0.00017849	3.9	3.4792e-08
0.2	0.7773	1.45	0.040305	2.7	0.00013433	3.95	2.3217e-08
0.25	0.72367	1.5	0.033895	2.75	0.00010062	4	1.5417e-08
0.3	0.67137	1.55	0.028377	2.8	7.5013e-05	4.05	1.0188e-08
0.35	0.62062	1.6	0.023652	2.85	5.5656e-05	4.1	6.7e-09
0.4	0.57161	1.65	0.019624	2.9	4.1098e-05	4.15	4.3847e-09
0.45	0.52452	1.7	0.01621	2.95	3.0203e-05	4.2	2.8555e-09
0.5	0.4795	1.75	0.013328	3	2.209e-05	4.25	1.8506e-09
0.55	0.43668	1.8	0.010909	3.05	1.608e-05	4.3	1.1935e-09
0.6	0.39614	1.85	0.008889	3.1	1.1649e-05	4.35	7.6594e-10
0.65	0.35797	1.9	0.0072096	3.15	8.3982e-06	4.4	4.8917e-10
0.7	0.3222	1.95	0.0058207	3.2	6.0258e-06	4.45	3.1089e-10
0.75	0.28884	2	0.0046777	3.25	4.3028e-06	4.5	1.9662e-10
0.8	0.2579	2.05	0.0037419	3.3	3.0577e-06	4.55	1.2374e-10
0.85	0.22933	2.1	0.0029795	3.35	2.1625e-06	4.6	7.7496e-11
0.9	0.20309	2.15	0.0023614	3.4	1.522e-06	4.65	4.8297e-11
0.95	0.17911	2.2	0.0018628	3.45	1.0661e-06	4.7	2.9953e-11
1	0.1573	2.25	0.0014627	3.5	7.431e-07	4.75	1.8485e-11
1.05	0.13756	2.3	0.0011432	3.55	5.1548e-07	4.8	1.1352e-11
1.1	0.11979	2.35	0.00088927	3.6	3.5586e-07	4.85	6.9375e-12
1.15	0.10388	2.4	0.00068851	3.65	2.4448e-07	4.9	4.2189e-12
1.2	0.089686	2.45	0.00053058	3.7	1.6715e-07	4.95	2.5531e-12

The Ideal Nyquist Channel

Recalling our equation for the output of the transmission system (eqn 8)) we see that to eliminate the effects of Intersymbol Interference (Section) and ensure perfect reception in the absence of noise we must control the received pulse shape such that

$$p(iT_b - kT_b) = \begin{cases} 1, & i = k \\ 0, & i \neq k \end{cases}$$

in which case $y(t_i) = \mu a_i$ for all i. For analysis we wish to transform this into the frequency domain. Since this is periodically sampled at time intervals T_b the Fourier transform will be periodic in frequency intervals $R_b = 1/T_b$ and may be written in the form

$$P_\delta(f) = R_b \sum_{n=-\infty}^{\infty} P(f - nR_b) = \int_{-\infty}^{\infty} \sum_{m=-\infty}^{\infty} \left[p(mT_b)\delta(t - mT_b)\right] e^{-j2\pi ft} dt$$

Where $m = i - k$. Imposing our condition for zero intersymbol interference gives

$$\begin{aligned} P_\delta(f) &= \int_{-\infty}^{\infty} p(0)\delta(t) e^{-j2b\pi ft} dt \\ &= p(0) \end{aligned}$$

and we get

$$\sum_{n=-\infty}^{\infty} P(f - nR_b) = T_b \qquad (9)$$

This is the *Nyquist criterion for distortionless baseband transmission in the absence of noise* .

Note $P(f)$ is the filter characteristic of the whole system including transmit, channel and receive filters.

As an illustration of the Nyquist criterion we plot in figure 99) the filter response for the three cases of increasing sample period $T < 1/(2W)$, $T = 1/(2W)$ and $T > 1/(2W)$. We see that for $T < 1/(2W)$ we cannot achieve the Nyquist criterion as specified in equation 9) in that the filter response falls to zero for some frequencies. For

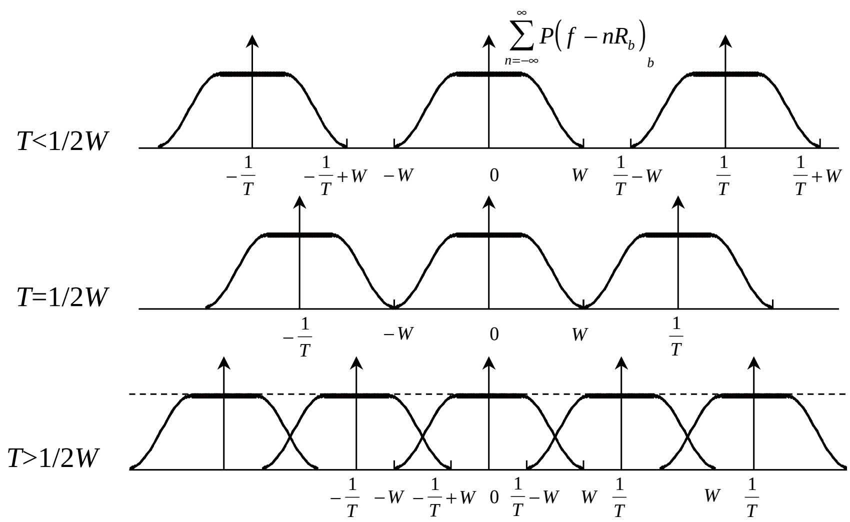

Figure 99: Illustration of the Nyquist criterion showing how the spectra vary with sample rate

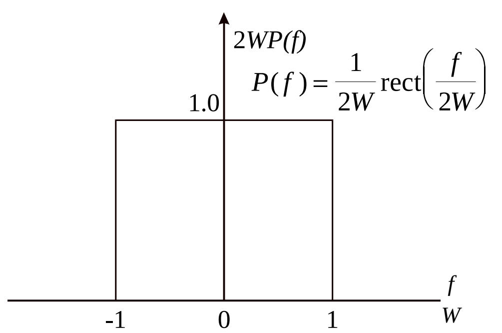

Figure 100: Ideal Nyquist Filter Response

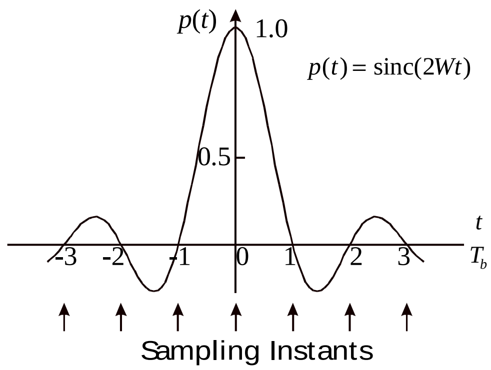

Figure 101: Ideal Nyquist Pulse

$T = 1/(2W)$ we can only achieve the Nyquist condition for a rectangular filter which gives a constant response across all frequencies.

The simplest solution to the Nyquist condition is therefore a *rectangular* filter function having a constant value between $-W$ and W.

$$
\begin{aligned}
P(f) &= \begin{cases} \frac{1}{2W} & -W < f < W \\ 0 & |f| > W \end{cases} \\
&= \frac{1}{2W}\mathrm{rect}\left(\frac{f}{2W}\right)
\end{aligned}
$$

where the over-all system bandwidth W is defined by

$$W = \frac{R_b}{2} = \frac{1}{2T_b} \tag{10}$$

i.e. no frequencies exceeding half the bit-rate are needed. This is shown in figure 100).

Taking the Fourier transform of this thus gives one signal waveform which produces zero intersymbol interference. This is shown in figure 101).

$$p(t) = \mathrm{sinc}(2Wt) \tag{11}$$

The bit rate $R_b = 2W$ is called the *Nyquist rate* and W is the *Nyquist bandwidth* corresponding to

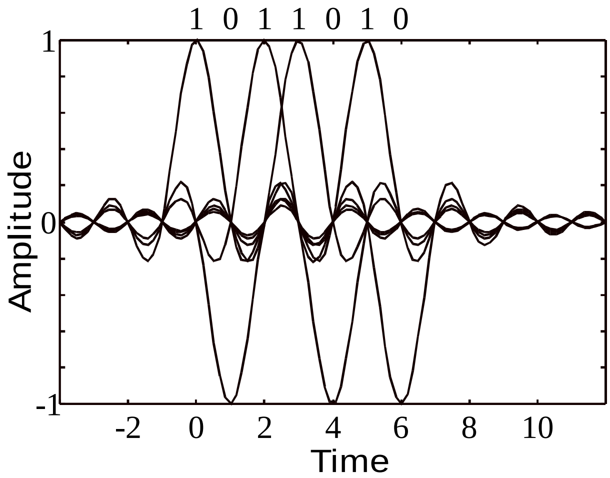

Figure 102: Zero interchannel interference for the ideal Nyquist Channel with the sequence 1011010

the *ideal Nyquist channel* given by equations 10) and 11).

Examining the function $p(t)$ we see that it has a maximum at $t = 0$ and goes through zero at integer multiples of T_b so that when the received waveform is sampled at intervals of T_b there is zero interference between symbols. This effect is shown for the sequence 1011010.

The ideal Nyquist channel solves the problem of zero intersymbol interference with the minimum bandwidth possible but has serious practical difficulties. First of all the filter function has abrupt transitions at the band edges $\pm W$ which is physically unrealisable. Secondly the sidelobes in the function $p(t)$ decrease only as $1/|t|$ which for large t means that there is little margin of error in the sampling times at the receiver. Indeed the sum of the contributions from successive bits where there is timing error may diverge causing erroneous decisions in the receiver.

The Raised Cosine Filter

The practical difficulties of the ideal Nyquist channel may be overcome using a filter function with an extendible bandwidth between W and $2W$ and without the sharp transitions. One such frequency characteristic is the *raised cosine spectrum* given by

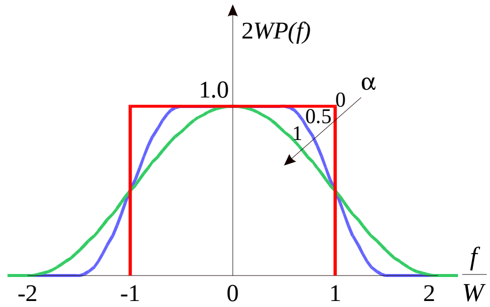

Figure 103: The raised Cosine Filter Characteristics

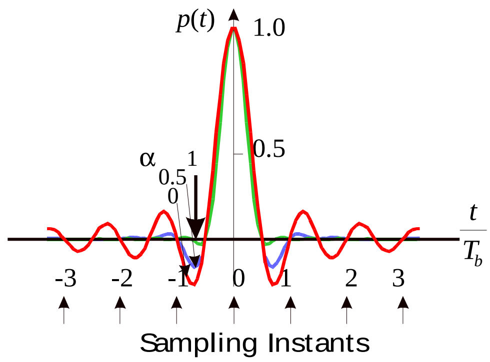

Figure 104: The corresponding raised cosine waveforms

$$P(f) = \begin{cases} \dfrac{1}{2W}, & 0 \leq |f| < f_1 \\ \dfrac{1}{4W}\left(1 - \sin\left(\dfrac{\pi(|f| - W)}{2W - 2f_1}\right)\right), & f_1 \leq |f| < 2W - f_1 \\ 0, & |f| \geq 2W - f_1 \end{cases}$$

or taking the inverse Fourier transform, in the time domain

$$p(t) = \text{sinc}(2Wt)\frac{\cos(2\pi\alpha Wt)}{1 - 16a^2W^2t^2}$$

where $\alpha = a - f_1/W$ is called the *rolloff factor* and represents the *excess bandwidth* over the ideal solution W . The frequency and time response for this frequency characteristic are given in figures 103) and 104) for values of $\alpha = 0,\ 0.5,\ 1$

Looking at the impulse response we see that it has a part $\text{sinc}(2Wt)$ which ensures that zero cross-

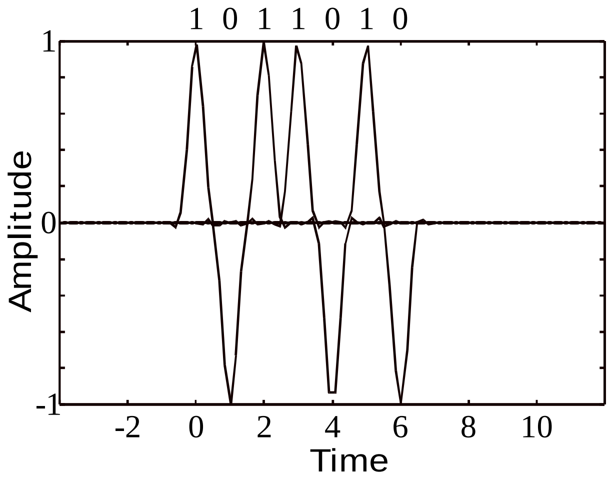

Figure 105: Zero interchannel interference for the raised cosine filter with the sequence 1011010.

ings of $p(t)$ occur at the desired sampling intervals $t = iT_b$ and a second factor decreasing as $1/t^2$ which attenuates the tails considerably reducing the sensitivity of the receiver to sampling time errors. The *full cosine-roll-off* with $\alpha = 1$ has the most gradual roll-off and the smallest tails in $p(t)$. Additionally its pulse width at half amplitude is equal to the bit duration T_b and their are zero crossings at half integer intervals of T_b as well as at the sampling times which aid timing signal extraction for synchronisation purposes. However these desirable properties are at the expense of a channel bandwidth double that required by the ideal Nyquist channel. The effect of the raised cosine filter on the waveform corresponding to binary sequence is shown in figure 105). This clearly demonstrated the superior properties of the raised cosine filter compared to that of the ideal Nyquist channel (figure 102).

Assessment
Deadline: 2009-11-14 23:59:00
No Questions: 2
Time Allowed: 15 min

Duobinary Coding

Correlative-level coding or partial-response signalling schemes involve adding intersymbol interference to a transmitted signal in a controlled way so as to achieve a signalling rate closer to the Nyquist rate of 2W symbols per second in a channel of bandwidth W using realisable filters. Since the intersymbol interference is known its effect can be interpreted at the receiver in a deterministic way. The simplest such scheme is *duobinary signalling* also known as *class I partial response* .

In duobinary signalling we apply the incoming sequence of uncorrelated binary symbols $\{b_k\}$ of duration T_b to a pulse amplitude modulator to produce a two-level sequence of short pulses $\{a_k\}$ of corresponding amplitudes ± 1. This two-level pulse sequence is passed to a *duobinary encoder* which transforms it to a sequence of three-level correlated pulses $\{c_k\}$ consisting of the sum of the present input value a_k and the previous value a_{k-1}.

$$c_k = a_k + a_{k-1}$$

It does this by adding a T_b delayed version of the signal to itself. This is the introduction of intersymbol interference under the designers control, the basis of correlative level coding.

The overall transfer function of the duobinary signalling scheme is given by the simple delay line filter cascaded with the ideal Nyquist channel given by

$$\begin{aligned} H_I(f) &= H_{\text{Nyquist}}(f)\left[1 + e^{-j2\pi f T_b}\right] \\ &= 2H_{\text{Nyquist}}(f)\cos(\pi f T_b)e^{-j\pi f T_b} \\ &= \begin{cases} 2\cos(\pi f T_b)e^{-j\pi f T_b} & |f| \le 1/(2T_b) \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

This function can easily be approximated in practice as there is continuity at the band edges i.e. this is a realisable filter.

In the time domain it will be two sinc pulses time displaced by T_b seconds and given by

$$\begin{aligned} h_I(t) &= \frac{\sin(\pi t/T_b)}{\pi t/T_b} + \frac{\sin(\pi(t-T_b)/T_b)}{\pi(t-T_b)/T_b} \\ &= \frac{T_b^2 \sin(\pi t/T_b)}{\pi t(T_b - t)} \end{aligned}$$

We see that the impulse response has only two distinguishable values at the sampling intervals. It

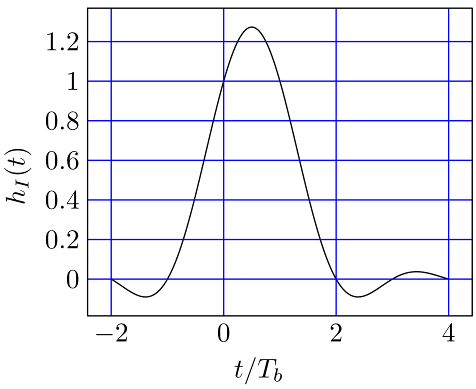

Figure 106: The Duobinary Impulse response

Table 2: Table of signals for duobinary coding

Binary Sequence $\{b_k\}$		0	0	1	0	1	1	0
Precoded Sequence $\{d_k\}$	1	1	1	0	0	1	0	0
Two-Level sequence $\{a_k\}$	+1	+1	+1	-1	-1	+1	-1	-1
Duobinary Coder output $\{c_k\}$		+2	+2	0	-2	0	0	-2
Decoded Sequence		0	0	1	0	1	1	0

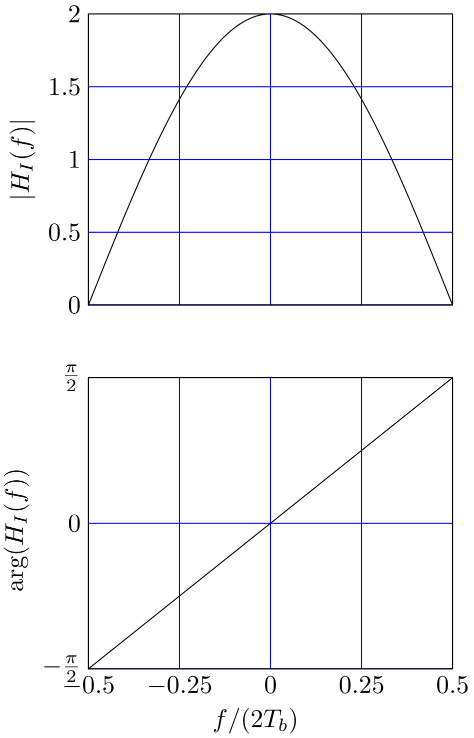

Figure 107: Duobinary Spectral Response

is called partial-response signalling because the response in any single signalling interval is only partial. Note also that the tails in the response decay as $1/|t^2|$, faster than in the ideal Nyquist channel. An estimate from the original pulse in the two level sequence $\hat{a}_k$ can be obtained from $\hat{a}_k = c_k - \hat{a}_{k-1}$ so if the previous estimate was stored estimate was correct the current estimate will be correct too. This is called *decision feedback*. A drawback of this technique is that the output estimate depends on previous output estimates and therefore errors will tend to propagate through the output. This may be overcome by *precoding* before the duobinary coding converting the original binary sequence $\{b_k\}$ into another $\{d_k\}$ using

$$d_k = b_k \otimes d_{k-1}$$

which is the applied to the pulse modulator. Figure 108) shows a schematic of a duobinary encoder with precoding.

The combined use of this non-linear operation with the duobinary coding yields

$$c_k = \begin{cases} 0 & \text{if } b_k = 1 \\ \pm 2 & \text{if } b_k = 0 \end{cases}$$

and so a simple threshold decision rule may be used for detecting the original binary sequence from the duobinary sequence and there is no propagation of errors.

A table illustrating all the different signals for duobinary coding with precoding is given in figure 2).

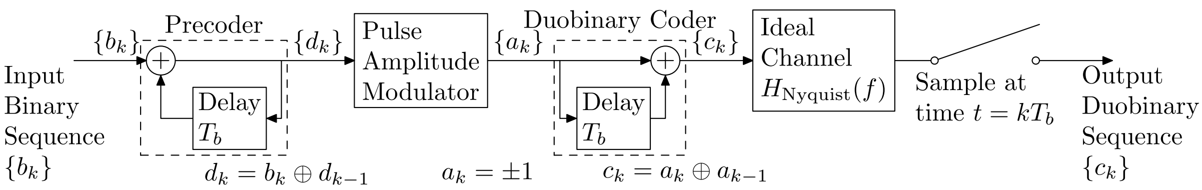

Figure 108: Duobinary coder with precoding

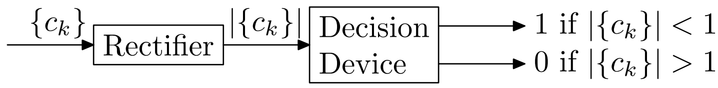

Figure 109: Duobinary Receiver

The benefit of duobinary encoding arises because the pulse representing 1 bit extends over two periods therefore reducing its bandwidth. However there is an overlap, one symbol with the next, so that the symbol rate is maintained at one bit per interval.

General correlative-level coding

General correlative level or partial response schemes use a tapped-delay-line filter with tap-weights $w_0, w_1, \ldots w_{n-1}$ to use a weighted linear combination of N ideal Nyquist pulses given by

$$h(t) = \sum_{n=0}^{N-1} w_n \text{sinc}\left(\frac{t}{T_b} - n\right)$$

This is illustrated in figure 110). An appropriate choice of tap-weights results in a variety of spectral shapes. Given in figure 3) is a table of tap-weights for different classes of partial response signalling schemes.

The duobinary signalling scheme has a power spectral density which is nonzero at the zero frequency i.e., has a d.c. component which is undesirable in many applications. Another common example is the so called *modified duobinary signalling* scheme which has the weighting factors $w_0 = +1$, $w_1 = 0$, $w_2 = -1$. Its transfer characteristics are shown in figure 111). As can be seen it has no d.c. component. As with duobinary coding precoding is used to remove error propagation.

Table 3: Table of tap weights for different partial response signalling schemes

Type N of class	w_0	w_1	w_2	w_3	w_4	w_5	
I	2	1	1				Duobinar
II	3	1	2	1			
III	3	2	1	-1			
IV	3	1	0	-1			Modified Duobinary
V	5	-1	0	2	0	-1	

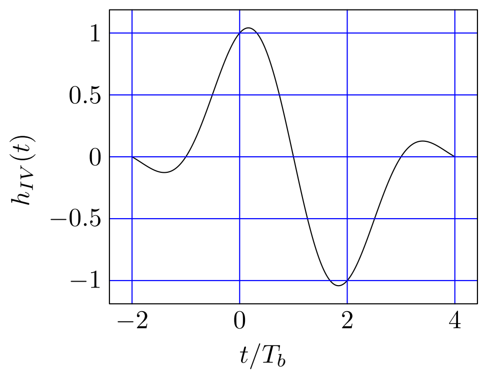

Figure 111: Impulse responses of the modified duobinary conversion filter

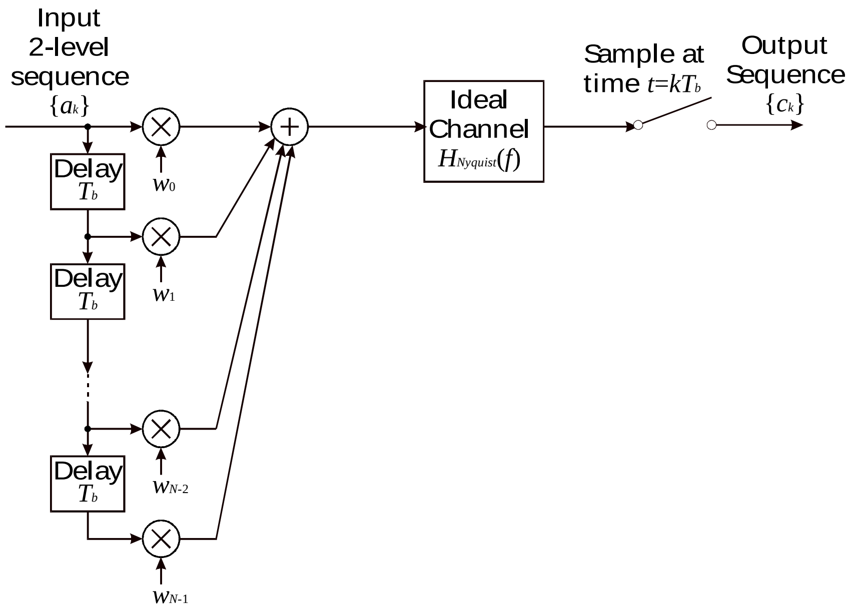

Figure 110: General Correlative Coder

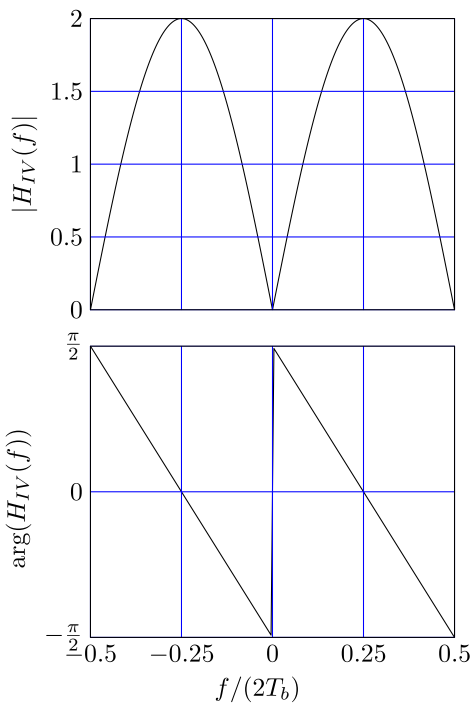

Figure 112: Spectral responses of the modified duobinary conversion filter

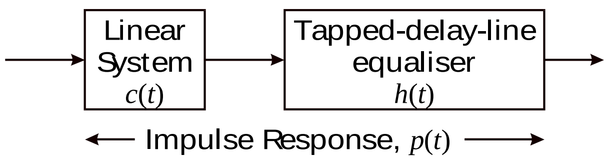

Figure 113: Equalisation

Tapped-delay-line equalisation

In the characteristics of a communication channel are known precisely it is possible to choose transmit and receive filters to make the intersymbol interference at the sampling instants arbitrarily small. In practice however the communicational channel may not be known precisely and there may be limitations in the precision to which the filters can be manufactured. In this case there will be residual distortion and this can be compensated or equalised using an *equaliser* which is placed after the receive filter as shown in figure 113).

A common of equaliser, illustrated in figure 114), is the tapped-delay-line filter consisting of a symmetric arrangement of $2N+1$ taps with weights

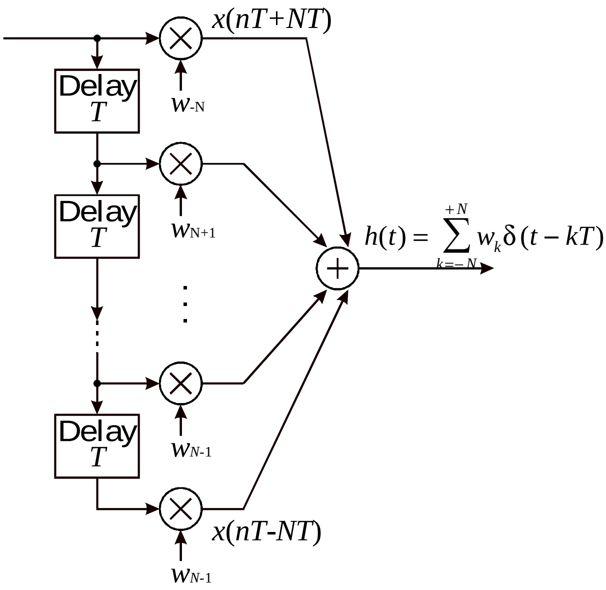

Figure 114: Tapped Delay Line Equaliser

$w_{-N}, \ldots, w_{-1}, w_0, w_1, \ldots, w_N$ giving an impulse response

$$h(t) = \sum_{k=-N}^{+N} w_k \delta(t-kT)$$

For a linear system of impulse $c(t)$ followed by the equaliser we have the total impulse function

$$\begin{aligned} p(t) &= c(t) * h(t) \\ &= c(t) * \sum_{k=-N}^{+N} w_k \delta(t-kT) \\ &= \sum_{k=-N}^{+N} w_k c(t) * \delta(t-kT) \\ &= \sum_{k=-N}^{+N} w_k c(t-kT) \end{aligned}$$

Evaluating at sampling times $t = nT$ we get the *discrete convolution sum*

$$p(nT) = \sum_{k=-N}^{+N} w_k c((n-k)T)$$

To eliminate intersymbol interference completely we require that there be no contributions from the current pulse signal at the sampling time from all other pulse signals. We note that here we have only $2N+1$ coefficients and can only satisfy the ideal condition for the $2N$ nearest neighbouring pulses. To achieve this we have $2N+1$ simultaneous equations relating the filter weighting factors to the signal contributions at discrete time intervals

$$\sum_{k=-N}^{+N} w_k c((n-k)T) = \begin{cases} 1, & n = 0 \\ 0, & n = \pm 1, \pm 2, \ldots, \pm N \end{cases}$$

or in matrix form

$$\begin{bmatrix} c_0 & \ldots & c_{-N+1} & c_{-N} & c_{-N-1} & \ldots & c_{-2N} \\ \vdots & & & \vdots & & & \vdots \\ c_{N-1} & \ldots & c_0 & c_{-1} & c_{-2} & \ldots & c_{-N-1} \\ c_N & \ldots & c_1 & c_0 & c_{-1} & \ldots & c_{-N} \\ c_{N+1} & \ldots & c_2 & c_1 & c_0 & \ldots & c_{-N+1} \\ \vdots & & & \vdots & & & \vdots \\ c_{2N} & \ldots & c_{N+1} & c_N & c_{N-1} & \ldots & c_0 \end{bmatrix} \begin{bmatrix} w_{-N} \\ \vdots \\ w_{-1} \\ w_0 \\ w_1 \\ \vdots \\ w_N \end{bmatrix} = \tag{12}$$

A tapped delay line equaliser described by this equation is referred to as a *zero-forcing equaliser*. It is optimum in the sense that it minimises the peak distortion and is relatively easy to implement. The longer we make the equaliser (the larger N is) the more closely the equalised system approaches the ideal of the Nyquist condition for distortionless transmission.

Adaptive Equalisation. In a telecommunications environment the channel is usually time varying. For example in a switched telephone network there may be differences between individual transmission links and in the number of transmission links switched in a connection. A fixed equaliser may not therefore be adequate to eliminate intersymbol interference and there is need for *adaptive equalisation* where the equaliser adjusts the weighting coefficients continuously and automatically. Generally the equaliser is used at the receiving end of the transmission system (*postchannel equalisation*). Prior to data transmission a suitable *training sequence* is transmitted through the channel allowing the filter to adjust its parameters. This is called precall equalisation and is usually sufficient as the average telephone channel changes little during a call. Generally the tapped delay line filter is synchronous in that the tap delay spacing is the same as the symbol duration of the transmitted signal.

The adaptation may be achieved by observing the error between the desired pulse shape and the actual pulse shape at the filter output measured at the sampling instants, and using this error to estimate the direction in which the tap-weights need to be changed to approach the optimum. We may use a *peak distortion criterion* minimising the worst-case intersymbol interference at the equaliser output similar to the zero-forcing concept described previously. Such an equaliser is optimum only when the peak distortion is not too severe and suffers from sensitivity to timing perturbations. More commonly a *mean-square error criterion* is used that is for minimum mean-square error, the cross-correlation between the output error sequence and the input sequence must have zeros for the $2N + 1$ components with integer lags corresponding to the index values of the available tap-weights of the filter. The algorithm used to achieve this is the *least-mean-square (LMS) algorithm* (see Haykin p. 455). During the training mode a known sequence, usually a pseudo-noise sequence i.e. deterministic with noise like characteristics, is transmitted. A synchronised version of this is generated at the receiver and compared with the received data to adjust the equaliser according to the LMS algorithm. The equaliser is then switched to decision directed mode. In this mode the decisions made by the equaliser are correct with high probability and the equaliser can use its estimate of the symbols to determine the error signal. The equaliser can therefore track relatively slow changes in channel characteristics.

Implementations of the LMS algorithm may be carried out using hardwired digital electronics, using programmable digital electronics or where the symbol weight is too high for digital implementation using *charge-coupled device* (CCD) technology.

Baseband M-ary PAM Transmission

In a *baseband M-ary PAM system* the pulse amplitude modulator is used to produce of one of M possible amplitude levels. Illustrated in figure 115)

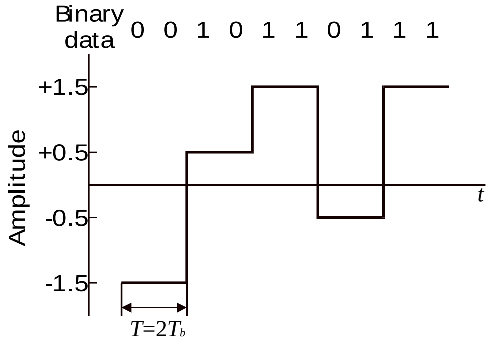

Figure 115: M-ary Baseband Transmission

Table 4: Quaternary Modulation.

Dibit	**Amplitude**
00	-1.5
01	-0.5
10	+0.5
11	+1.5

is the quaternary case with $M = 4$ and the binary sequence 00 10 11 01 11. In an M -ary system the information source emits an alphabet with M symbols and each amplitude at the modulator output corresponds to a distinct symbol so there are M distinct amplitude levels to be transmitted. The table 4) shows the amplitude mapping used in the example quaternary system.

The symbol duration is denoted by T and we refer to $1/T$ as the signalling rate in symbols per second or *baud* . In general in an M -ary system each baud is equal to $\log_2 M$ bits per second and the symbol duration may be related to the bit duration T_b of an equivalent binary PAM system as

$$T = T_b \log_2 M$$

Therefore in a given channel bandwidth we find that by using an M -ary PAM we are able to transmit information at a rate that is $\log_2 M$ times faster than the corresponding binary system. However to realise the same average probability of error an M -ary PAM system requires more transmitted power. For $M \geq 2$ and the average probability of error much less than 1 the transmitted power must be

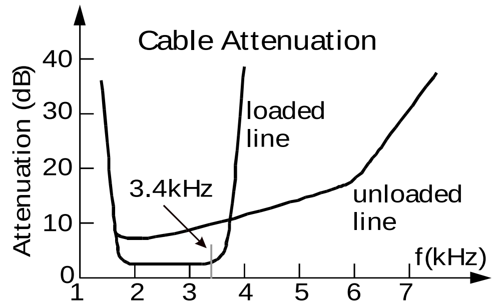

Figure 116: Cable Attenuation

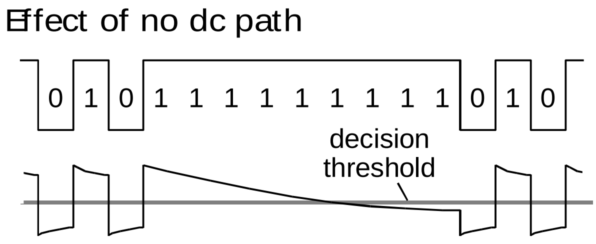

Figure 117: Effect of no DC path

increased by a factor $M^2/\log_2 M$ compared to a binary PAM system.

The transmitter and receiver for an M-ary PAM system are necessarily more complex than the equivalent binary PAM system. In particular at the receiver the signal must be compared to a number of *threshold* or *slicing* levels. The procedures used in designing the transmission and receive filters and for calculating the error rate due to noise are similar to those used for baseband binary PAM. Not that such modulation schemes involve an energy variation i.e. symbols have varying energies. They are therefore generally not suitable for transmission channels which contain a non-linear response.

Assessment
Deadline: 2009-11-14 23:59:00
No Questions: 3
Time Allowed: 15 min

Line code requirements

After information has been source coded to remove redundancy and error-correct coded the signal may require further encoding to be give good baseband transmission characteristics. In particular the frequency spectrum should 'match'; the channel spectrum.

PCM signals are often carried over cables originally installed for analogue telephony. In order to minimise distortion in such cables they are artificially loaded with inductive coils at 2km intervals which introduces high losses at frequencies above 4kHz. The inductors must be removed to improve the high frequency response.

Another serious problem for PCM is that 2km sections of the analogue cable are coupled together using transformers meaning that there is no dc path through the transmission system. The effect of this is to introduce 'droop' of the signal level when long sequences of a constant amplitude occur resulting in drop below the decision threshold and errors. Appropriate line coding would generate a signal with no d.c. component.

It is often also important that the spectrum of the transmitted waveform have a component at the bit rate frequency to enable bit-rate clock recovery at the receiver. A summary of the requirements on a line code are given below.

Transparency The code must be independent of bit sequence to impose no restrictions on message content

Efficiency Each symbol of the code should contribute to information transmission

Unique Decodability Each symbol must be able to be decoded without ambiguity to yield the original bit sequence

Suitable Energy Spectrum Zero dc component and small low frequency content to avoid baseline wandering; small high frequency content to minimise effects of inter-channel interference

Timing Information Content For ease of clock extraction, coded signal should have high energy content at clock frequency

In a *binary code* each binary symbol may be either of two distinct values, normally denoted as 0 and 1. In a *ternary code* each symbol may be one of three distinct values or kinds. *Pseudo-ternary* codes also use three levels but carry no more information per bit than a binary code.

Binary Signalling Formats

Shown in figure 118) are examples of the different binary signalling formats.

Of-Off Signalling Each Symbol 1 is represented by transmitting a pulse of constant amplitude for the duration of the symbol and each 0 is represented by turning off the symbol.

Nonreturn-to-zero (NRZ) Signalling Symbols 1 and 0 are represented by pulses of equal positive and negative amplitudes.

Return-to-zero (RZ) Signalling Each signal 1 is represented by a positive pulse of width less than the symbol width (here a rectangular pulse of half symbol width) and each symbol 0 is represented by transmitting no pulse.

Bipolar Return-to-zero (BRZ) signalling It uses three levels (a pseudo-ternary code). Positive and negative pulses of equal amplitude are use alternately for symbol 1 and no pulse is used for symbol 0. This is also known as alternate mark inversion (AMI) encoding. On average this coding scheme produces a signal with no d.c. component allowing transmission through lengths of line that are transformer coupled and therefore have no d.c. signal path. The symbol rate does not increase and decoding is achieved by rectification. In regenerators the positive marks and the negative marks have to be handled separately. The efficiency is rather low, each ternary symbol being capable of transmitting 1.6 bits, and as the code carries 1 bit per symbol the redundancy is 60%. It has an error detecting capability since errors can violate the simple coding rule. The pulses may be of width less than the symbol width. This is also called alternate mark inversion.

Split-phase (Manchester code) Each symbol 1 is represented by a positive pulse followed by a negative pulse with both pulses being of equal amplitude and half symbol width. For symbol 0 the polarities of the two pulses are reversed. The d.c. component is suppressed and the Manchester code has relatively insignificant low-frequency components independent of signal statistics.

Table 5: Miller Code

Input Data Bits	Output Coded Sequence
0	x 0
1	0 1
$x = 0$, if preceding input bit is 1	
$x = 1$, if preceding input bit is 0	

Differential Encoding The information is encoded in terms of signal transitions. In the example shown a transition represents a symbol 0 and no transition represents a 1. The coded signal may be inverted without loss of information. Recovery is performed by comparing the polarity of adjacent symbols to establish whether or not a transition has occurred.

Runlength Limited Codes

Codes that have a restriction on the number of consecutive 1s or 0s in a sequence are generally called *runlength-limited codes* . These codes are generally described by two parameters, say d and κ where d denotes the minimum number of 0s between 2 1s in a sequence and κ denotes the maximum number of 0s between 2 1s in a sequence. When used with NRZI modulation, the effect of placing d 0s between successive 1s is to spread the transitions farther apart, thus reducing the overlap in the channel response due to successive transitions and thus reducing of inter-symbol interference. Setting an upper limit κ on the runlength of 0s occurs that transitions occur frequently enough so that symbol timing information can be recovered from the received modulated signal.

Runlength limited codes are commonly denoted (d, κ). Such codes may be represented by a finite state sequence machine with $\kappa + 1$ states. A code commonly used in magnetic recording is the Miller code which is coded as shown in figure 36 which en-

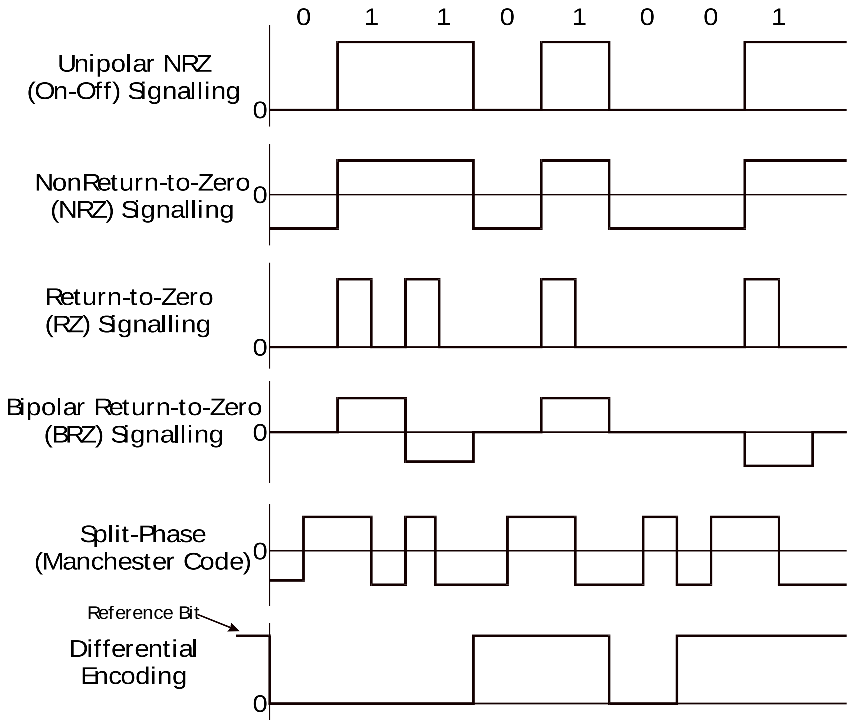

Figure 118: Binary Signalling Formats

Table 6: HDB3 violation encoding

Polarity of Preceding Pulse	Number of pulses since last bipolar violation	
	ODD	EVEN
+	000+	-00-
−	000-	+00+

sures appropriate transitions for the magnetic pick-up head.

High Density Bipolar 3 This code is a variant on alternate mark inversion (AMI) which ensures that no more than 3 zeros are transmitted together. It is so called because it has higher density of marks than AMI, it is bipolar like AMI, i.e. has both +1 and -1 as well as zero, while the 3 in its name is the maximum number of zeros.

The coding is as follows. When four zeros occur in the binary (NRZ) signal the AMI waveforms, which should be entirely zero is replaced by a code which violates the AMI polarity rule to distinguish it from a real mark representing a 1. The actual code transmitted depends on the polarity of the preceding binary 1 and on whether the number of binary 1s transmitted since the last HDB3 code (bipolar violation) is odd or even as shown in figure 6). Figure 119) is an example showing HDB3 coding of a waveform and comparing it with AMI, RZ and NRZ waveforms.

Error detection is still available with this code, because violations of the code rules will be observed, albeit after a few digits have been transmitted beyond the violation because decoding cannot be done instantaneously.

The Eye Diagram

An important diagnostic technique used in the operational environment for evaluating the performance of communication system is the eye pattern or eye diagram. It is the synchronised superposition of all possible realisations of the signal of interest viewed within a particular signalling interval and is recorded by superimposing multiple sweeps in a storage oscilloscope as illustrated in figure 120). The interior region of the eye diagram is called the eye opening.

The eye pattern provides a good deal of information about the performance of a data transmission system.

- The width of the eye opening (A) defines the time interval over which the received signal can be sampled without error from intersymbol interference, it is apparent that the optimum sampling time is the instant where the eye is open the widest (t*).
- The sensitivity of the system to timing errors is determined by the rate of closure of the eye as the sampling time is varied (slope D above).
- The height of the eye opening, (B), at a specified sampling time, defines the noise margin of the system.
- The distortion of zero crossings (c)
- The maximum distortion (E)

When the effect of intersymbol interference is severe the upper traces cross the lower traces resulting in a closed eye. In such a situation it is impossible to avoid errors will occur due to the combined effects of noise and intersymbol interference in the system.

In the case of an M-ary system the eye pattern contains (M-1) eye openings vertically above each other. In linear systems with random data all eye openings should be identical however in practice nonlinearities in the communication channel will lead to asymmetries in the eye pattern.

Digital Passband Transmission

In this section we will be looking at requirements for the modulation processes of modulation and demodulation, and the transmission and receiver filters for passband transmission. We will concentrate on the effects of noise which dominates the performance of passband systems, and look at the various trade-offs associated with different passband modulation techniques.

In *digital passband transmission* the incoming data stream is modulated onto a (sinusoidal) carrier with fixed frequency limits imposed by a band-pass channel of interest. The major issue of concern is

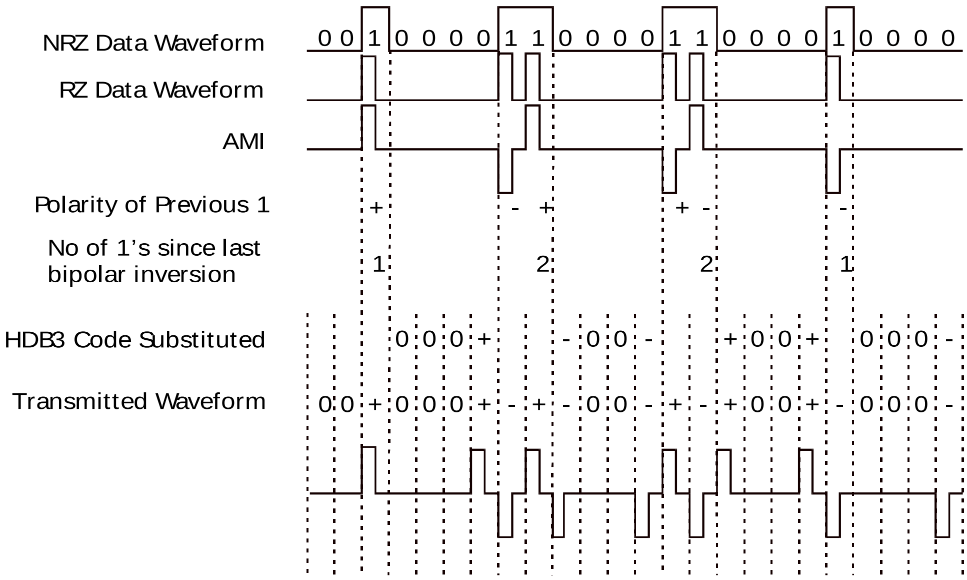

Figure 119: HDB3 Coding

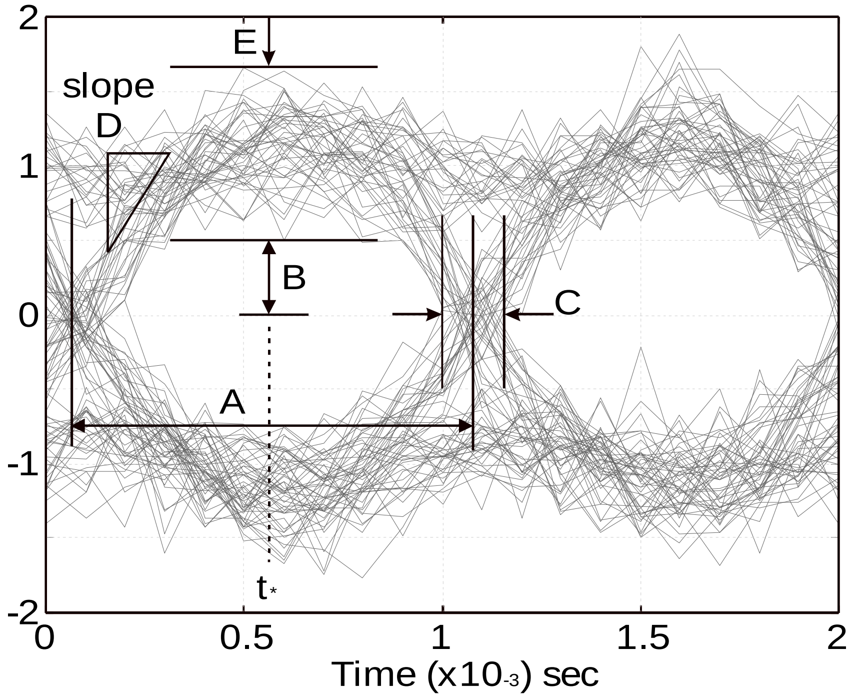

Figure 120: The Eye Diagram

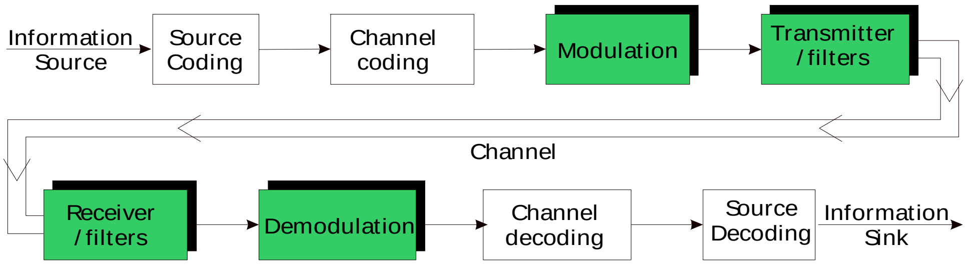

Figure 121: A Passband Transmission System

the optimum design of the receiver to minimise the average probability of symbol error in the presence of noise. The communications channel may be a microwave radio link, satellite channel etc. The modulation process involves switching (keying) the amplitude, phase or frequency in some way according to the incoming data. The three basic signalling schemes *amplitude-shift keying* (ASK), *frequency-shift keying* (FSK) and *phase-shift keying* (PSK) are special cases of the analogue amplitude modulation, frequency modulation and phase modulation respectively and are studied in the laboratory experiments associated with this course. Both PSK and FSK have constant envelope making them impervious to amplitude nonlinearities commonly found on microwave channels and they are therefore usually preferred to ASK signals.

In our passband model we will assume a *message source* emits a *symbol* every T seconds with the symbols belonging to an alphabet $(m_1, m_2, \ldots, m_N)$ of M symbols. Generally M is a power of two: $M = 2^k$, k is a integer. e.g. a quaternary PCM encoder has alphabet of 4 symbols (00,01,10,11).

Without prior information we assume that all symbols are equally likely so we have the symbol probabilities as

$$p_i = P(m_i) = \frac{1}{M} \text{for all } \ i$$

The M -ary output of the message source is send to a *signal transmission encoder* which produces a corresponding vector $\vec{S}_i$ made up of N real elements, one set for each of the M symbols of the source alphabet and $N \leq M$. With the vector $\vec{S}_i$ the modulator constructs a *distinct* signal of duration T seconds as the representation of the message m_i produced by the message source. This signal $s_i(t))$ must be of finite energy,

$$E_i = \int_0^T s_i^2(t)dt, \ i = 1, 2, \ldots, M$$

is real valued and one such signal is emitted every T seconds. The particular signal chosen depends on the message to be transmitted and possibly also on the signals transmitted in previous time slots. With a sinusoidal carrier the modulator will generally introduce a step change in the amplitude, phase, frequency or some combination thereof to distinguish the signals.

In our passband communication channel we will assume that it is linear, has a bandwidth large enough to support the modulated signal $s_i(t)$ without distortion and that the transmitted signal is perturbed by an *additive white Gaussian noise* (AWGN) process $w(t)$. The received signal will then be given by

$$x(t) = s_i(t) + w(t), \ \begin{cases} 0 \leq t \leq T \\ i = 1, 2, \ldots, M \end{cases}$$

The receiver observes the received signal $x(t)$ for T seconds and makes a best estimate of the transmitted signal $s_i(t)$ or message m_i. This is accomplished first by operating on the received signal $x(t)$ to produce an observation vector $\vec{x}$ which is then used together with prior knowledge of the modulation format used in the transmitter and the probabilities $p(m_i)$ to produce an estimate of the message $\hat{m}$. Noise from the channel makes the decision process statistical and prone to errors. The goal is to design a receiver to minimise the *average probability of symbol error* :

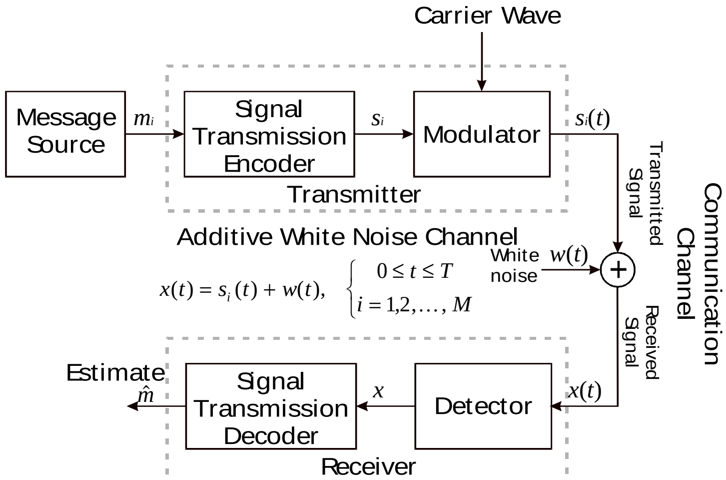

Figure 122: Model of a Passband Transmission System

$$p_e = \sum_{i=1}^{M} p(\hat{m} \neq m_i)p(m_i)$$

We assume that the receiver is time synchronised with the transmitter and knows the instant of time in which a particular signal is transmitted. In *coherent detection* we also assume that the detector is *phase locked* to the transmitter and knows the absolute phase of the signal. If this phase synchronism is not required then we have *non-coherent detection* .

We use the above construction as the basis of designing the optimum receiver exploiting a geometric representation of the known set of signals which gives insight and a simplification of detail.

Gram-Schmidt Orthogonalisation

Any set of M energy signals $\{s_i(t)\}$ may be represented as a linear combination of some N orthonormal basis functions $\phi_j(t)$ where $N \leq M$ in a process called *Gram-Schmidt Orthogonalisation* i.e. we represent our signals as

$$s_i(t) = \sum_{j=1}^{N} s_{ij}\phi_j(t), \begin{cases} 0 \leq t \leq T \\ i = 1, 2, \ldots, M \end{cases}$$

where the coefficients of the expansion are defined by

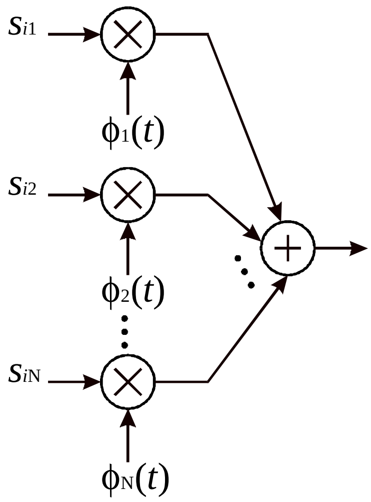

Figure 123: Gram-Schmidt Signal Synthesis

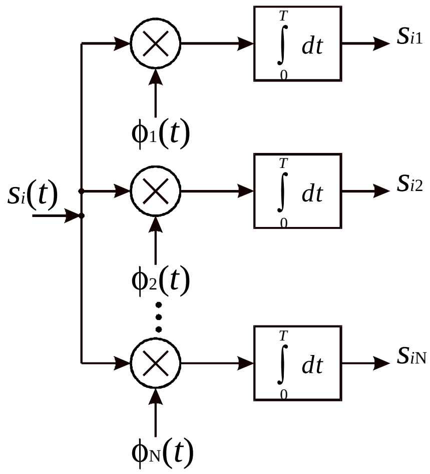

Figure 124: Gram-Schmidt Signal Decomposition

$$s_i j = \int_0^T s_i(t)\phi_j(t)dt \begin{cases} i = 1, 2, \ldots, M \\ j = 1, 2, \ldots, N \end{cases}$$

This is ilustrated in figure 123).

This overlap integral may be thought of as determining a measure of how like basis function $\phi_j(t)$ signal $s_i(t)$ is. The basis functions are *orthonormal* i.e. they are each normalised to have unit energy and they are orthogonal to each other over the interval $0 \le t \le T$.

$$\int_0^T \phi_i(t)\phi_j(t)dt = \begin{cases} 1, & i = j \\ 0, & i \ne j \end{cases}$$

Orthogonality simply means that none of the basis functions can be represented by a linear superposition of others. The Fourier sine and cosine function sequences over a finite interval are one example of an orthogonal set of basis functions.

The coefficients s_{ij} are the coefficients of the N -dimensional vector $\vec{S}_i$ which is used to generate the signal $s_i(t)$ using a modulator which consists of N multipliers, each provided with its own basis function, followed by a summer.

Given a signal $s_i(t)$ as an input we can use a bank of N *product-integrators* or *correlators* each with its own basis function input to determine the set of coefficients s_{ij} in the vector $\vec{S}_i(t)$ and this forms the first stage of a detector in the receiver. This is illustrated in figure 124).

To perform the Gram-Schmidt Orthogonalisation we begin by defining the first basis function as

$$\phi_1(t) = \frac{s_1(t)}{\sqrt{E_1}}$$

where E_1 is the energy of the signal $s_1(t)$. The clearly we have $s_{11} = \sqrt{E_1}$ and $\phi_1(t)$ has unit energy as required. Given a second signal $s_2(t)$ we can calculate the coefficient s_{21} as

$$s_{21} = \int_0^T s_2(t)\phi_1(t)dt$$

and we introduce a new intermediate function which is remainder of this second signal not represented by the first basis function

$$g_2(t) = s_2(t) - s_{21}\phi_1(t)$$

which must be orthogonal to $\phi_1(t)$ over the sampling interval. Normalising this gives us the second basis function

$$\begin{aligned} \phi_2(t) &= \frac{g_2(t)}{\sqrt{\int_0^T g_2^2(t)dt}} \\ &= \frac{s_2(t) - s_{21}\phi_1(t)}{\sqrt{E_2 - s_{21}^2}} \end{aligned}$$

where E_2 is the energy of the signal $s_2(t)$. The basis function form an orthonormal set as required. We may continue in this fashion and in general define

$$g_i(t) = s_i(t) - \sum_{j=1,i-1} s_{ij}\phi_j(t)$$

where the coefficients themselves are defined by

$$s_{ij} = \int_0^T s_i(t)\phi_j(t)dt, \; j = 1, 2, \ldots, i-1$$

and from which we define a set of basis functions

$$\phi_i(t) = \frac{g_i(t)}{\sqrt{\int_0^T g_i^2(t)dt}}, \; i = 1, 2, \ldots, N$$

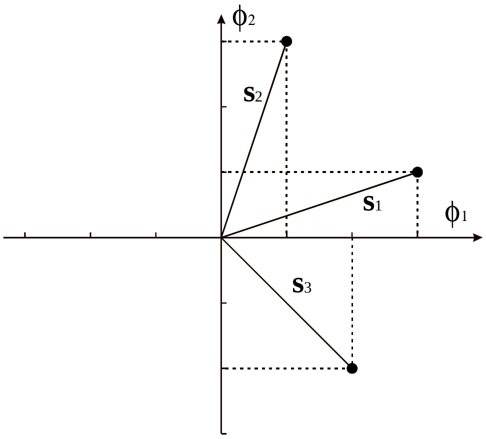

Figure 125: Geometric representation of $M = 3$ signals on $N = 2$ basis functions.

which form an orthonormal set. If the set of signals are a linearly independent set then $N = M$, if not then we will have less basis functions than signals $N < M$.

The Geometric Interpretation of Signals

Once we have obtained a convenient set of basis functions $\{\phi_j(t)\},\ j = 1, 2, \ldots, N$ then each signal in the set $\{s_i(t)\},\ i = 1, 2, \ldots, M$ is determined by the vector of its coefficients.

$$\vec{S}_i = \begin{bmatrix} s_{i1} \\ s_{i2} \\ \vdots \\ s_{iN} \end{bmatrix},\ i = 1, 2, \ldots, M$$

This vector $\vec{S}_i$ is called the *signal vector* and we may conceptually think of our signals as a set of M points in N *-dimensional Euclidean space* called *signal space* with the axis corresponding to the N orthonormal basis functions. This enables us to visualise our set of signals geometrically and as we will see simplifies the analysis when noise is introduced. Figure 125) illustrates this with the representation of 3 signals, $\vec{S}_1, \vec{S}_2, \vec{S}_3$ on 2 basis functions ϕ_1, ϕ_2.

In our geometric interpretation of signals we can introduce lengths of vectors and angles. The length of a signal vector $\vec{S}_i$ is defined by the inner product of the signal with itself

$$|\vec{S}_i|^2 = \vec{S}_i^T \vec{S}_j = \sum_{j=1}^{N} \vec{S}_{ij}^2$$

This is equal to the energy of the signal E_i which can be proved by expanding the signals in term of the basis functions and using the properties of basis function orthogonality as follows:

$$\begin{aligned} E_i &= \int_0^T s_i^2(t)dt \\ &= \int_0^T \left[\sum_{j=1}^{N} s_{ij}\phi_j(t)\right] \left[\sum_{k=1}^{N} s_{ik}\phi_k(t)\right] dt \\ &= \sum_{j=1}^{N}\sum_{k=1}^{N} s_{ij}s_{ik} \int_0^T \phi_j(t)\phi_k(t)dt \\ &= \sum_{j=1}^{N} s_{ik}^2 \end{aligned}$$

For two signal vectors we can calculate the Euclidean distance between them.

$$|s_i - s_k|^2 = \sum_{j-1}^{N}(s_{ij} - s_{kj})^2$$

We can also calculate the cosine of the angle between the two signal vectors defined by

$$\cos\theta_{ij} = \frac{\vec{S}_i^T \vec{S}_j}{|\vec{S}_i||\vec{S}_h|}$$

Two vectors are thus orthogonal or perpendicular to each other if their inner produce is zero giving $\theta_{ij} = 90°$.

Signals and Noise

If instead of receiving in our bank of N product integrators the transmitted signal $s_i(t)$ we receive signal received from an AWGN channel $x(t)$ given by

$$x(t) = s_i(t) + w(t),\ \begin{cases} 0 \le t \le T \\ i = 1, 2, \ldots, M \end{cases}$$

where $w(t)$ is the sample function of a white Gaussian noise process of zero mean and power spectral

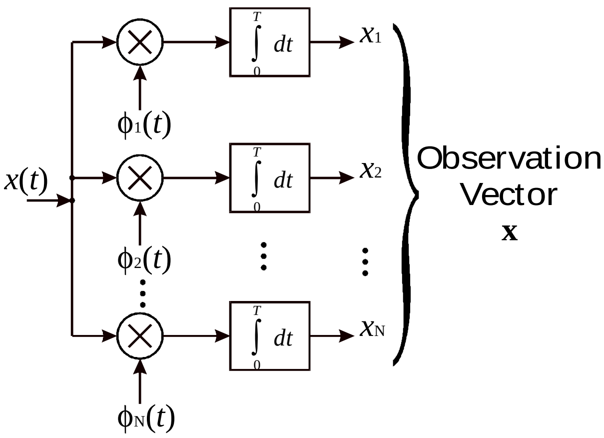

Figure 126: Correlation type demodulator

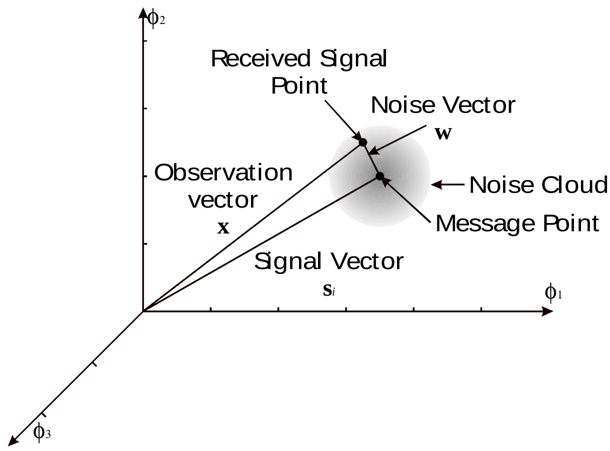

Figure 127: A signal with a Gaussian distributed noise cloud

density $N_0/2$ then the output of correlator j in the demodulator will be a sample value of a random variable given by

$$
\begin{aligned}
x_j &= \int_0^T x(t)\phi_j(t)dt \\
&= s_{ij} + w_j
\end{aligned}
$$

The first component s_{ij} is a deterministic quantity contributed by the transmitted signal $s_i(t)$ and the second component is a sample of a random variable that arises in the presence of noise at the receiver input

$$w_j = \int_0^T w(t)\phi(t)dt$$

Since $w(t)$ is due to Gaussian white noise and the basis function has unit energy we can get the simple result (see standard texts e.g. Haykin p. 487) that the variance is given by

$$\sigma^2_{w_j} = \frac{N_0}{2}$$

The complete received signal vector may then be considered to be a vector of N Gaussian random variables with mean values s_{ij} and variances $N_0/2$ which can be written in the form

$$\vec{x} = \vec{S}_i + \vec{w}$$

$$\mu_{x_j} = \sigma_{ij}$$

$$\sigma^2_{x_j} = \frac{N_0}{2} \text{ for all} j$$

This received signal point may lie anywhere inside an observation cloud which is a Gaussian-distributed cloud centred on the message point as illustrated right.

We may write a conditional probability density function which will be the product of the Gaussian probability density function of each of the elements. The likelihood functions of an AWGN channel will therefore be defined by

$$f_{\vec{X}}(\vec{X}|m_i) = \frac{1}{\sqrt{\pi N_0}} \exp\left(-\frac{1}{N_0}\sum_{j=1}^{N}(x_j - s_{ij})^2\right) \quad i = 1, 2, \ldots, M \tag{13}$$

This is the probability after receiving observation vector $\vec{X}$ given that the message m_i was transmitted.

Equivalence of Matched Filter and Correlation Demodulator

Suppose we have a bank of N filters with their impulse responses matched to each of the N basis functions. The N matched filters are time reversed

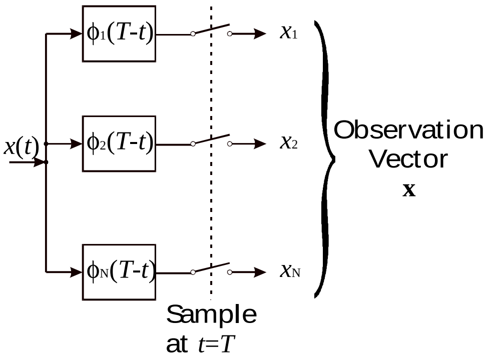

Figure 128: Matched Filter Demodulator

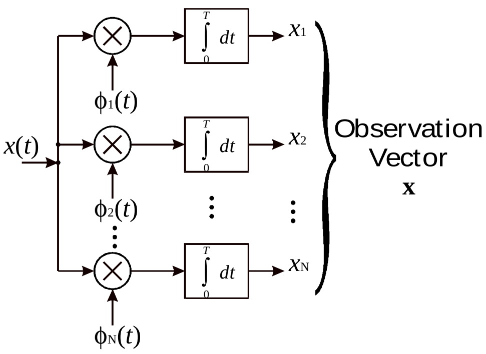

Figure 129: Correlation Detector

and delayed versions of inputs and will thus have impulse responses given by

$$h_j(t) = \phi(T - t)$$

The filter output will be the convolution of the input signal and filter responses

$$\begin{aligned} y_j(t) &= \int_{-\infty}^{\infty} x(\tau)h_j(t-\tau)d\tau \\ &= \int_{-\infty}^{\infty} x(\tau)h_j(T-t+\tau)d\tau \end{aligned}$$

Suppose then that we sample the output of these filters at the time $t = T$. The outputs will then be given by

$$\begin{aligned} y_j(T) &= \int_{-\infty}^{\infty} x(\tau)\phi_j(\tau)d\tau \\ &= \int_{0}^{T} x(\tau)\phi_j(\tau)d\tau \end{aligned}$$

Since by definition the basis signals are zero outside the bit interval.

Thus we see that the matched filter demodulator shown in here produces the identical output to the correlation demodulator. The outputs of both these demodulators can then be interpreted by the minimum distance signal transmission decoder to determine the most likely transmitted message for a given received signal.

Probability of Error for a passband system

The minimum distance decision rule implies that we can divide the N dimensional observation space Z into a number of decision regions Z_i which are bounded by N -1 dimensional hyperplane boundaries. For received signals lying inside a particular decision region the most likely transmitted message is the one inside that region. Shown here is the example for $M = 4$ signals and $N = 2$ dimensions assuming that the signals are transmitted with equal energy and probability.

An error will be deemed to have occurred if when a symbol m_i is sent the received vector $\vec{x}$ does not lie in the associated decision region Z_i. Averaging

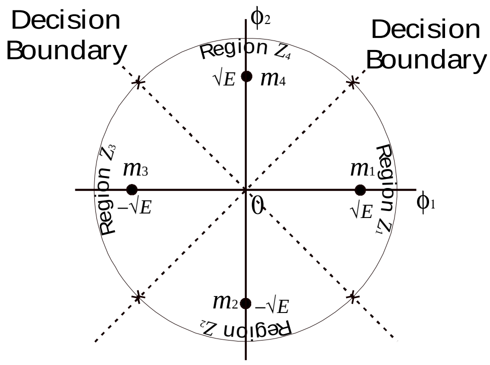

Figure 130: Division of Observation Space

over all possible transmitted symbols we obtain the average probability of symbol error P_e :

$$\begin{aligned} P_e &= \sum_{i=1}^{M} p(\vec{x} \text{ does not lie in } Z_i | m_i \text{ sent}) p(m_i \text{ sent}) \\ &= \frac{1}{M} \sum_{i=1}^{M} p(\vec{x} \text{ does not lie in } Z_i | m_i \text{ sent}) \\ &= 1 - \frac{1}{M} \sum_{i=1}^{M} p(\vec{x} \text{ lies in } Z_i | m_i \text{ sent}) \end{aligned}$$

which can be written in terms of the conditional probability function as

$$P_e = 1 - \frac{1}{M} \sum_{i=1}^{M} \int_{Z_i} f_{\vec{X}}(\vec{X}|m_i) d\vec{X}$$

The integral is N -dimensional. While this is conceptually straightforward it may be numerically impractical except for a few simple cases which we will now examine further. For more complicated systems that can be simplified to obtain a maximum upper bound called the union bound of on the probability of error (See the standard texts.)

The Maximum Likelihood Decoder

At the receiver we wish to choose the most probable transmitted symbol given a particular received signal. We maximise the *maximum a posteriori probability* (MAP) criterion for the optimum detector: i.e. we

$$\text{Set} \hat{m} = m_i \text{ if} P(m_i \text{sent}|\vec{X}) \geq P(m_k \text{sent}|\vec{X}) \text{for all} k \neq i$$

Using Bayes theorem we can rewrite this in terms of transmitted symbol probabilities and likelihood functions getting

$$\text{Set} \hat{m} = m_i \text{ if} \frac{p_k f_{\vec{X}}(\vec{X}|m_k)}{f_{\vec{X}}(\vec{X})} \text{is a maximum for all} k = i$$

where p_k is the probability of occurance of the transmitted symbols which we will consider to be the same for all symbols and thus independent of the transmitted signal., $f_{\vec{X}}(\vec{X}|m_k)$ is the likelihood function when m_k is transmitted and $f_{\vec{X}}(\vec{X})$ is the unconditional joint probability of random vector $\vec{X}$ and is independent of the transmitted signal. The MAP rule can then become

$$\text{Set} \hat{m} = m_i \text{if} \ln \left(f_{\vec{X}}(\vec{X}|m_k \right) \text{ is a maximum for all} k = i$$

Using natural logarithms simplifies the calculations. Substituting in our Gaussian probability density function from equation we get

$$\ln \left(f_{\vec{X}}(\vec{X}|m_i) \right) = -\frac{1}{2} N \ln(\pi N_0) - \frac{1}{N_0} \sum_{j=1}^{N} (x_j - s_{ij})^2$$

The first term in this is a constant and the second term is a constant times the Euclidean distance between the received vector and the signal vector. Clearly then if all signals are transmitted with equal probability the most likely transmitted signal vector is that whose signal point is nearest to the received vector. This is called *minimum distance detection.*

In practice the need for squarers in performing this decision rule is removed if we recognise that

$$\sum_{j=1}^{N} (x_j - s_{ij})^2 = \sum_{j=1}^{N} x_j^2 - 2 \sum_{j=1}^{N} x_j s_{ij} + \sum_{j=1}^{N} s_{ij}^2$$

The first term in this expansion is independent of i and can be ignored. The second term is the inner product of the observation vector $\vec{X}$ and the signal

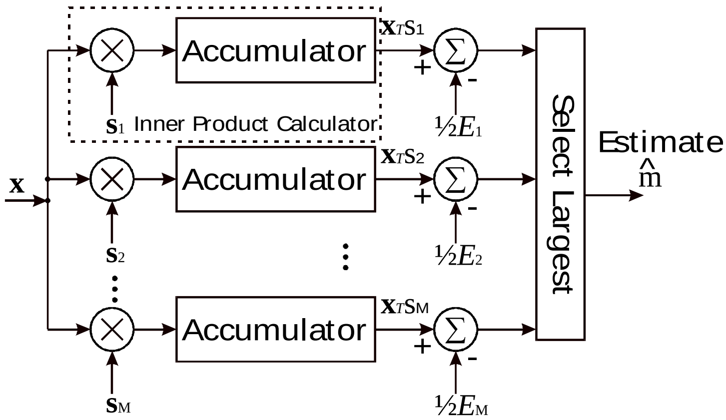

Figure 131: MAP detector

vector $\vec{S}_i$ and represents the projection of the received signal on each of the message signal vectors. The final term is the energy of the transmitted signal $s_i(t)$ and may be viewed as a bias term serving as a bias term for signal sets that have unequal energies, such as PAM. The decoder based on this decision rule then takes on the form as shown in figure 131).

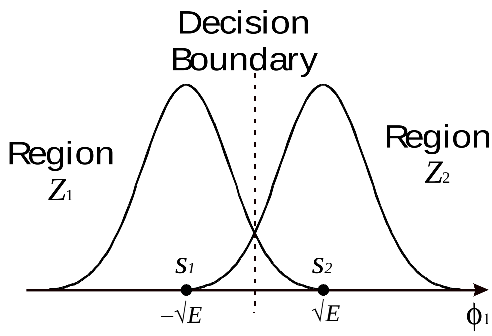

Figure 132: Signal Space Diagram for Binary PSK

Coherent Binary Phase Shift Keying (PSK)

In coherent binary phase shift keying (PSK) a phase shift of ϕ in a carrier modulated signal is used to distinguish the binary symbols 0 and 1. The two signals are therefore given by

$$\begin{aligned} s_1(t) &= \sqrt{\frac{2E_b}{T_b}}\cos(2\pi f_c t) \\ s_2(t) &= \sqrt{\frac{2E_b}{T_b}}\cos(2\pi f_c t + \pi) \\ &= -\sqrt{\frac{2E_b}{T_b}}\cos(2\pi f_c t) \end{aligned}$$

where f_c is the carrier frequency. This system may be represented using a single basis function

$$\phi_1(t) = \sqrt{\frac{2}{T_b}}\cos(2\pi f_c t),\ 0 \le t \le T_b$$

with the two signal vectors amplitude amplitudes given by

$$\begin{aligned} s_1 &= -\sqrt{E_b} \\ s_2 &= \sqrt{E_b} \end{aligned}$$

where E_b is the signal energy which is also in this case the energy per bit

A coherent binary PSK system is therefore characterised by having a signal space that is one dimensional with a signal constellation consisting of two message points of equal and opposite amplitude Such signals with equal energy and a cross correlation coefficient of -1 are called *antipodal*. This signal space diagram is shown in figure 132). If the symbols are equiprobable then the rule for deciding which symbol was transmitted is to choose the closest message point. There is therefore a decision threshold midway between the two signal points.

As can be seen from the signal space diagram the probability of error of for a symbol will be given by the area under the probability density function lying at the wrong side of the decision boundary. For additive white Gaussian noise with a power spectral density of $N_0/2$ this will be given by

$$P_e = \frac{1}{2}\text{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right)$$

and if the symbols are equiprobable this will also be the average probability of error per symbol. If we set the carrier frequency f_c to zero we have a bipolar NRZ baseband transmission system and so not surprisingly we have the same average probability of symbol error as calculated for such a system.

Generation of a coherent binary PSK may simply be done by encoding the binary data as a bipolar NRZ signal and multiplying the result by the carrier basis function as shown in figure 133). Detection is carried out by multiplying the received signal by a locally generated coherent reference signal $\phi_1(t)$, integrating over the symbol period and comparing the output with a threshold amplitude of zero to determine if a 1 or 0 was sent (see figure 134)).

Assessment
Deadline: 2009-11-28 23:59:00
No Questions: 1
Time Allowed: 10 min

Coherent Binary Amplitude Shift Keying (ASK)

Amplitude shift keying, also called Pulse Amplitude Modulation (PAM) involves modulating the amplitude of a sinusoidal carrier to represent the signals. There is one (carrier) basis function given by

$$\phi_1(t) = \sqrt{\frac{2}{T_b}}\cos(2\pi f_c t),\ 0 \leq t \leq T_b$$

In the case of binary PAM an amplitude of zero represents binary zero and an amplitude of $\sqrt{E}$ represents a 1. The signal space diagram is therefore one dimensional as shown, with signal vectors

$$\begin{aligned} s_1 &= 0 \\ s_2 &= \sqrt{E} \end{aligned}$$

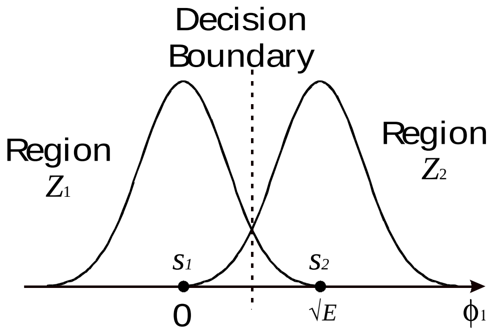

Figure 135: Signal Space Diagram for Binary ASK

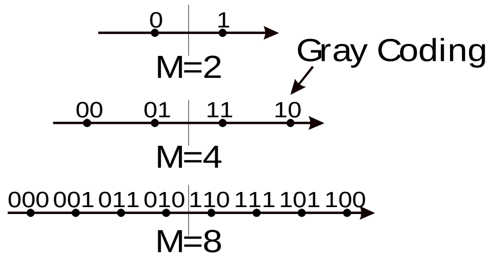

Figure 136: Gray coding

Note that with PAM the different signals are of different energies. We usually wish to work with the energy per bit which for binary PAM as described here will be given by.

$$E_b = E/2$$

From the signal space diagram it is obvious that the signal points are at half the distance of those for PSK and the probability of error per symbol will be given by

$$\begin{aligned} P_e &= \frac{1}{2}\text{erfc}\left(\frac{1}{2}\sqrt{\frac{E}{N_0}}\right) \\ &= \frac{1}{2}\text{erfc}\left(\sqrt{\frac{E_b}{2N_0}}\right) \end{aligned}$$

i.e. such a binary amplitude shift keyed system requires twice the energy per bit to achieve the same probability of symbol error as the coherent binary phase shift keyed system.

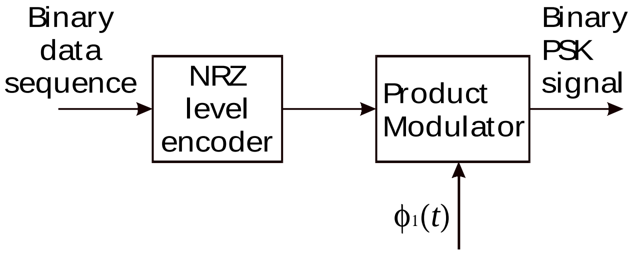

Figure 133: Binary PSK Transmitter

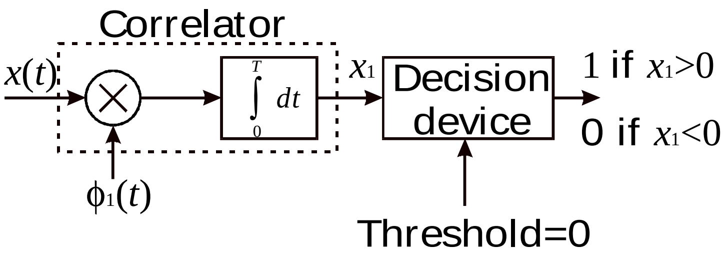

Figure 134: Binary PSK Receiver

M-ary Amplitude Shift Keying The amplitude shift keying system can obviously be extended to transmit M -ary data using M different signal levels. Generally we choose to transmit a power of 2, $M = 2^k$ levels in an M -ary transmission system for compatibility with binary data systems. Shown in figure 136) is the signal space diagram for some M -ary ASK systems. Note that in this figure the Gray coding is used for the binary codes. Such coding ensures that adjacent signals differ by only one bit. Since the most probable errors are for adjacent signal points Gray coding of the message signals ensures that signal bit errors are the most likely thus minimising the bit error rate for a particular symbol error rate.

There is one basis function representing the carrier with the signal vectors values are given by

$$s_i = (2m - 1 - M)d, \; m = 1, 2, \ldots, M$$

where d is half the distance between adjacent signal points. The energy of the M^{th} signal point will be given by

$$E_m = (2m - 1 - M)^2 d^2$$

from which we calculate the average error per signal (assuming that all signals are equal probable) as follows

$$\begin{aligned} E_{av} &= \frac{1}{M}\sum_{m=1}^{M} E_m = \frac{d^2}{M}\sum_{m=1}^{M}(2m - 1 - M)^2 \\ &= \frac{d^2}{M}\left(\frac{1}{3}M(M^2 - 1)\right) \\ &= d^2(M^2 - 1)/3 \end{aligned}$$

Each M -ary signal carries $\log_2 M$ bits of information so the average transmission energy per bit is given by.

$$E_b = \frac{d^2(N^2 - 1)}{3\log_2 M}$$

In order to calculate the symbol error rate for equiprobable symbols we see that the M signal points are separated by $m - 1$ decision thresholds midway between adjacent points. Therefore for the outside levels the probability of symbol error will be given by

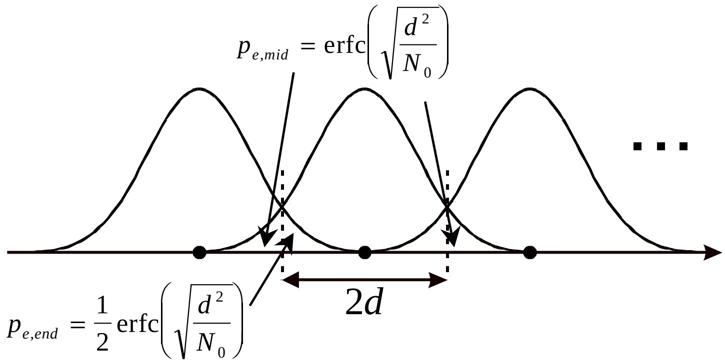

Figure 137: Calculation of symbol error rate for M-ary ASK

$$P_e = \frac{1}{2}\text{erfc}\left(\sqrt{\frac{d^2}{N_0}}\right)$$

The inside signal points, however, can have an error on either side and therefor have twice this probability of error. The total probability of symbol error per transmission will therefore be given by

$$\begin{aligned} P_m &= \frac{M-1}{M}\text{erfc}\left(\sqrt{\frac{d^2}{N_0}}\right) \\ &= \frac{M-1}{M}\text{erfc}\left(\sqrt{\frac{(3\log_2 M)E_b}{(M^2-1)N_0}}\right) \end{aligned}$$

when substitution is made for the average energy per bit. The symbol error rate versus signal to noise ratio for some different values of M is plotted in figure 138).

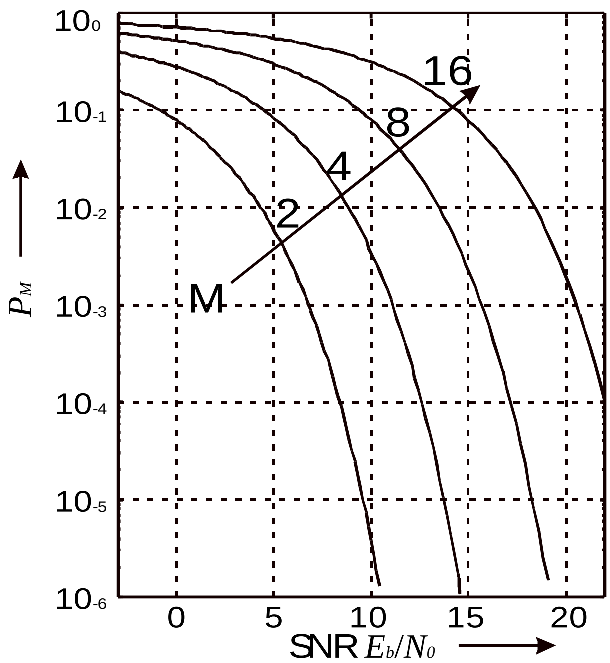

Figure 138: Probability of symbol error rate versus signal to noise ratio for M-ary ASK

Assessment
Deadline: 2009-11-28 23:59:00
No Questions: 1
Time Allowed: 5 min

Coherent Binary Frequency Shift Keying (FSK)

In a binary frequency shift keyed system (FSK) the signals are distinguished from each other by having different frequencies and may be described by

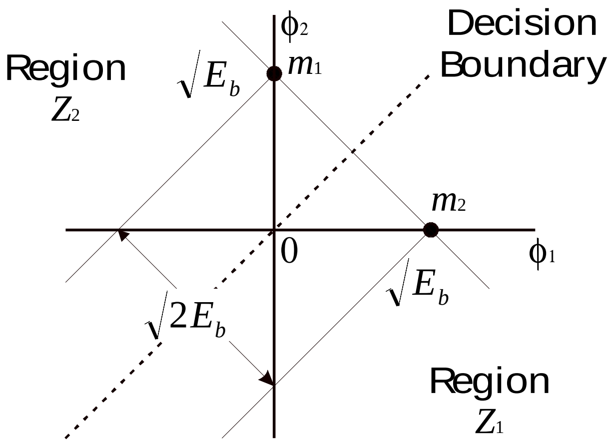

Figure 139: Signal Space Diagram for coherent Binary FSK

$$s_i(t) = \sqrt{\frac{2E}{T_b}}\cos(2\pi f_i t)$$

where $i = 1, 2$ and the transmitted frequency is given by

$$f_f = \frac{n_c + i}{T} \tag{14}$$

where n_c is some fixed integer. This form is an example of continuous-phase frequency shift keying (CPFSK) and is also called *Sunde's FSK* . The condition expressed in equation 14) that the carrier frequencies must be an integral multiple of the symbol frequency ensures that the basis functions given by

$$\phi(t) = \sqrt{\frac{2}{T_b}}\cos(2\pi f_i t)$$

satisfy the orthogonality conditions.

The signal space diagram for binary FSK is therefore two dimensional with signal vectors given by

$$s_{ij} = \begin{cases} \sqrt{E} & i = j \\ 0 & i \neq j \end{cases}$$

as shown.

Note that the number of frequencies to be transmitted can be increased to M for an M -ary transmission system producing an M dimensional signal space diagram.

The Euclidean distance between the two message points for binary FSK is equal to $\sqrt{2E_b}$ and the decision boundary for equiprobable symbols will lie as a diagonal line mid way between the two message points. The average probability of symbol error for equal energy binary signals depends only on the distance between the message points in signal space and so will be given by

$$P_e = \frac{1}{2}\text{erfc}\left(\sqrt{\frac{E_b}{2N_0}}\right)$$

We thus see that we have to double the bit signal-to-noise ratio in order to maintain the same bit error rate as a coherent binary PSK system.

Generation of coherent binary FSK simply requires an on-off level encoder alternatively switching the two frequency basis functions which are then transmitted as shown. In this case we are assuming that the two oscillators are synchronised appropriately to maintain the continuous phase condition. We could alternatively use a single voltage controlled oscillator whose frequency is shifted in accordance with the continuous phase requirement.

To detect the original binary sequence given a noisy received signal $x(t)$ we use two correlators supplied with locally generated coherent reference versions of the basis signals $\phi_1(t)$ and $\phi_2(t)$. The correlator outputs are subtracted and the difference compared with a zero threshold to determine which symbol was most likely sent.

Coherent Minimum Shift Keying (MSK) In the coherent detection of a binary FSK signal the phase information in the received signal was not fully exploited other than to synchronise the receiver with the transmitter. With proper utilisation of the phase information when performing detection the noise performance can be improved significantly at the expense of increased receiver complexity.

This is achieved by using a deviation ratio of a half as compared to one in FSK The signal space diagram for MSK has four signal points with two representing symbol 1 and two representing symbol 0. The average probability of is the same as for PSK

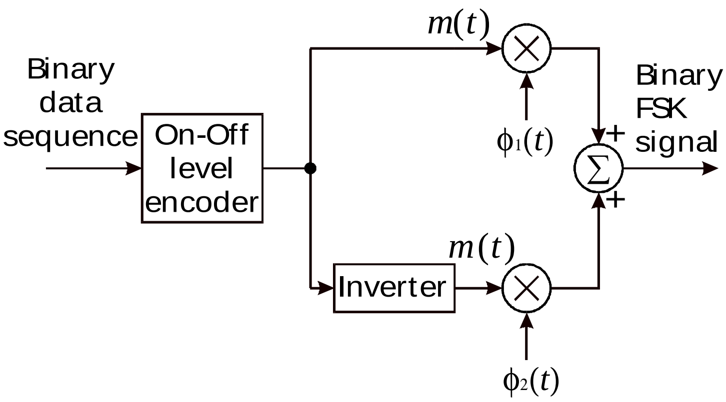

Figure 140: Transmitter for coherent Binary FSK

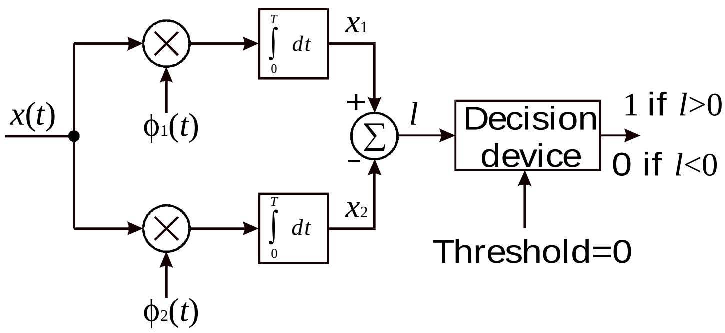

Figure 141: Receiver for coherent Binary FSK

however this improved performance is achieved by detector observation over two symbol periods for each symbol. MSK is an example of a modulation scheme with memory - there is a correlation from one symbol period to the next and as such is considered beyond the scope of this course. It is however an important technique used in practice.

Coherent M-ary Phase Shift Keying (MPSK)

Digital phase shift keying can be extended from binary to an M -ary transmission system by using M different phases of the carrier to represent the signal. The set of signals will then take the form

$$s_m(t) = \sqrt{\frac{2E}{T}} \cos\left(2\pi f_c t + \frac{2\pi}{M}(m-1)\right)$$

where E is the signal energy, f_c the carrier frequency and $m = 1, 2, \ldots, M$. Such a set of signals can be represented using two basis functions with the same carrier frequency but in quadrature (with a $\pi/2$ phase shift) as follows

$$\begin{aligned} \phi_1(t) &= \sqrt{\frac{2}{T}} \cos(2\pi f_c t) \\ \phi_2(t) &= -\sqrt{\frac{2}{T}} \sin(2\pi f_c t) \end{aligned}$$

The m signal vectors will take the co-ordinates

$$\begin{aligned} s_{m1} &= \sqrt{E} \cos\left(\frac{2\pi}{M}(m-1)\right) \\ s_{m2} &= \sqrt{E} \sin\left(\frac{2\pi}{M}(m-1)\right) \end{aligned}$$

We therefore have a two-dimensional signal space map with the M signal points equally spaced around a circle of radius $\sqrt{E}$. Shown in figure 142) are the cases for $M = 4$ (called *quadriphase-shift keying* (QPSK)) and $M = 8$. Note the use of Gray encoding again to minimise the bit error rate for a given symbol rate.

Shown in figure 143) is a block diagram for a QPSK transmitter. The incoming binary sequence is transformed by a non-return to zero level encoder into a polar form which is then split into two channels off odd and even bits which are then used to

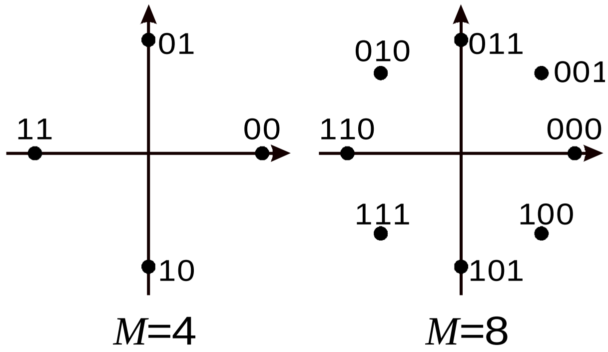

Figure 142: Some dignal Space Diagrams for M-ary PSK

modulate a pair of quadrature carriers. The result of a pair of PSK signals which may be detected independently due to their orthogonality. We will use this in determining the probability of error for QPSK The two PSK signals are added to produce the transmitted QPSK signal.

The QPSK receiver has a pair of correlators provided with a locally generated pair of coherent synchronised quadrature reference signals. The output of these are compared with a threshold of zero to determine if a 1 or 0 was sent and the two binary sequences are multiplexed together to generate the original binary sequence. The receiver is essentially two PSK receivers in parallel using quadrature frequency references and generating alternate odd and even bits.

Except for the cases of $M = 2$ and $M = 4$ the calculation of the probability of symbol error for M -ary PSK does not reduce to a simple form. We will perform this calculation for the case $M = 4$ (QPSK).

Calculation of Error Rate for QPSK We show here again the signal space diagram for QPSK indicating the decision regions and the distance between adjacent signal points which is given by

$$E_c = \sqrt{2}E$$

If we look at each point we see that we can consider it to have two orthogonal PSK systems with the distance between channels as above. The probability

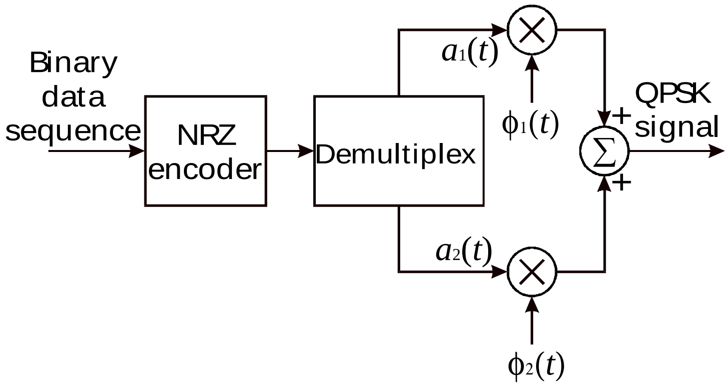

Figure 143: Transmitter for QPSK

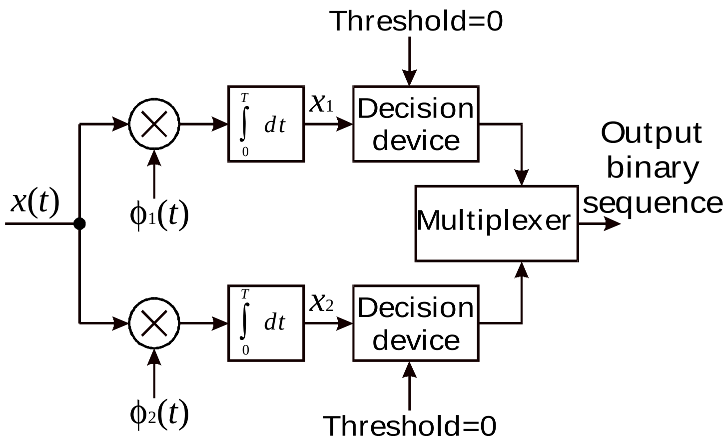

Figure 144: Receiver for QPSK

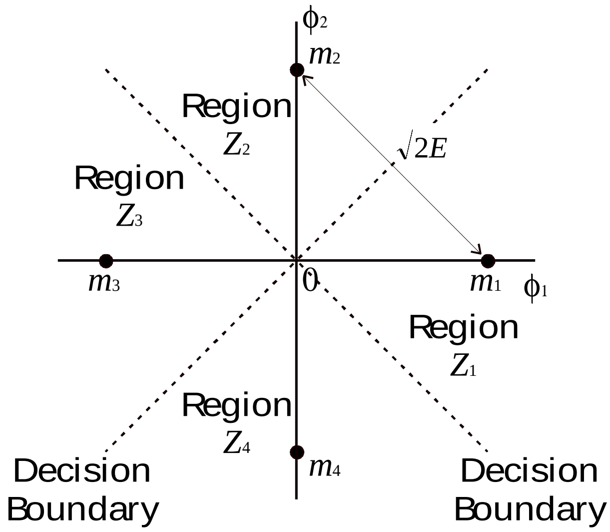

Figure 145: Signal space diagram for QPSK

of error for each of these independent channels will be given by

$$P_{e,2} = \frac{1}{2}\text{erfc}\left(\sqrt{\frac{E}{2N_0}}\right)$$

From this we can determine the probability of there being a correct decision for this message point to be

$$P_{c,4} = (1 - P_{e,2})^2$$

and thus the probability of symbol error for QPSK is given by

$$P_4 = \text{erfc}\left(\sqrt{\frac{E}{2N_0}}\right) - \frac{1}{4}\text{erfc}^2\left(\sqrt{\frac{E}{2N_0}}\right)$$

Substituting for the average energy per bit $E_b = E/2$ and taking the approximation that the signal to noise ratio is not small we find that for QPSK the probability of symbol error will be given by

$$\text{if} E/N_0 \gg 1 \text{then} P_e \approx \text{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right)$$

With Gray encoding of the incoming dibits (symbols) we find that the bit error rate of QPSK is exactly

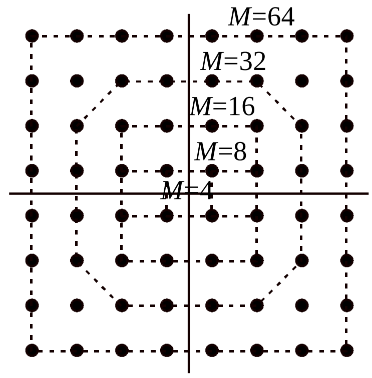

Figure 146: Rectangular QAM Constellations

$$P_e = \frac{1}{2}\text{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right)$$

which is the exactly the same as for PSK. This we expect since Gray encoded QPSK essentially consists of the transmission of two independent PSK channels. However QPSK achieves this bit error rate with half the bandwidth requirements of PSK.

Quadrature Amplitude Modulation (QAM)

In M-ary PSK the signal amplitude is the same for all signals thus constraining the signal points to a circular constellation. If this constraint is removed and the in-phase and quadrature signal components are allowed to vary independently we have a scheme called quadrature amplitude modulation (QAM). In this case we can have any signal constellation we choose. QAM waveforms are a combination of PAM and PSK. The basis functions are as for PSK.

We may choose any combination of M_1 -level PAM and M_2 -level PSK to give an $M = M_1 M_2$ level QAM signal constellation. Figure 146) shows some rectangular signal constellations for different values of M. The optimum signal constellation will be that which requires the least average power for a given minimum distance between signal points. For $M = 4$ this will be when the four points lie on a circle and the resulting signal constellation is the same as that for QPSK. For $M = 8$ the optimal signal constellation consists of the points lying on two circles as indicated in figure 147). For $M \geq 16$

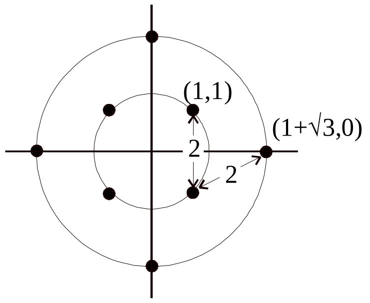

Figure 147: Optimum M=8 QAM Constellation

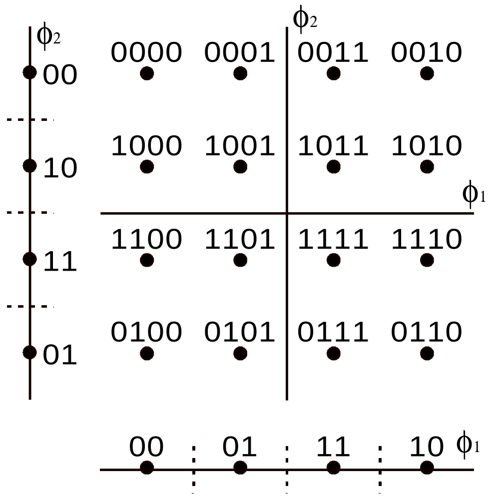

Figure 148: Square Signal space diagram for M=16, showing division into two 4-ary PAM channels

there are many more ways of building up the signal constellation however rectangular signal constellations have the distinct advantage that they can be generate as two PAM signal impressed on the phase quadrature carriers and they are easily demodulated. Also for $M \geq 16$ the average transmitter power required for the rectangular constellation is only slightly worse than the best M -ary QAM signal constellation.

Calculation of symbol error rate for QAM

Since the two quadrature components are orthogonal they can be detected independently without interference. The rectangular QAM constellation can thus divided into two independent polar ASK signals, and the probability of error calculated separately for each before combining to find the overall probability of symbol error as was done for the example of QPSK. Here we will perform the calculation for a square QAM constellation as shown in figure 148) for $M = 16$. When Gray encoded we see that this constellation consists of two independent L-ary PAM signals such that

$$L = \sqrt{M}$$

From our calculation for the probability of error for M-ary PAM we have the probability of error for each channel to be given by

$$P_L = \frac{L-1}{L}\text{erfc}\left(\sqrt{\frac{d^2}{N_0}}\right)$$

where 2 d is the distance between adjacent signal points. From this we calculate the probability of a correct decision for each channel and thus find that the overall probability of error for the M-ary QAM signal is given by

$$\begin{aligned} P_M &= 1-(1-P_L)^2 \\ &\simeq 2P_L \end{aligned}$$

The average energy per transmission for each PAM channel is given previously, and since QAM has two PAM channels the average energy for the QAM will be twice that and thus given by

$$E_{av} = \frac{2d^2(L^2-1)}{3}$$

From this we can calculate the average energy per bit given that each signal transmits $\log_2 M$ bits and substituting for L we get

$$E_b = \frac{2d^2(M-1)}{3\log_2 M}$$

Putting all these together we find for a square M -ary QAM constellation the probability of symbol energy as a function of bit energy to noise ratio to be given by

$$P_M \simeq 2\frac{\sqrt{M}-1}{\sqrt{M}}\text{erfc}\left(\sqrt{\frac{(3\log_2 M)E_b}{2(M-1)N_0}}\right)$$

Substituting in $M = 4$ gives the same probability of symbol error as was calculated for QPSK as expected.

Assessment
Deadline: 2009-11-28 23:59:00
No Questions: 2
Time Allowed: 5 min

Non-coherent Modulation

Coherent detection requires knowledge of the carriers waves phase reference thus providing optimum error performance with the digital modulation format of interest. This knowledge may not be practicable in which case we resort to non-coherent detection. Here we will consider non-coherent orthogonal modulation including differential phase shift keying and non-coherent binary frequency shift keying as specific examples.

If we transmit a signal

$$s_i(t) = \sqrt{\frac{2E}{T}} \cos(2\pi f_i t), \; 0 \le t \le T$$

Then the received signal, from an AWGN channel, with an unknown phase shift θ will be given by

$$\begin{aligned} s(t) &= \sqrt{\frac{2E}{T}} \cos(2\pi f_i t + \theta) + w(t) \\ &= \sqrt{\frac{2E}{T}} \begin{pmatrix} \cos(\theta)\cos(2\pi f_i t) \\ \quad - \sin(\theta)\sin(2\pi f_i t) \end{pmatrix} \\ & quad + w(t), \; 0 \le t \le T \end{aligned}$$

where we have applied well a known trigonometric identity. If we apply this to a pair of correlators with reference signals $\sqrt{2/T}\cos(2\pi f_i t)$ and $\sqrt{2/T}\sin(2\pi f_i t)$ we will get output signals $\sqrt{E}\cos\theta$ and $-\sqrt{E}\sin\theta$. Squaring these, adding and taking the square root will remove the dependence on the unknown phase θ. The *quadrature receiver* is thus as in figure 149).

We can then use this receiver to detect two orthogonal signals $s_1(t)$ and $s_2(t)$ which are sent over a noisy channel which shifts the carrier by an unknown amount. We assume that the phase shifted signals, denoted $g_1(t)$ and $g_2(t)$, remain orthogonal and have an energy E regardless of the phase shift and that the channel has additive white gaussian noise with power spectral density $N_0/2$. The received signal will then be

$$x(t) = \begin{cases} g_1(t) + w(t) & 0 \le t \le T \\ g_2(t) + w(t) & 0 \le t \le T \end{cases}$$

The generalised receiver, shown in figure 150), will be used to distinguish between $s_1(t)$ and $s_2(t)$ regardless of carrier phase. This consists of a pair of non-coherent (quadrature) filters matched to the basis functions. The filter outputs are envelope detected, sampled and compared to product the output thus removing the need for a phase reference.

The average probability of error for such a non-coherent receiver can be shown to be

$$P_e = \frac{1}{2}\exp\left(-\frac{E}{N_0}\right)$$

where E is the energy per symbol. Derivation of this is considered beyond this course.

The need for the envelope detector in the non-coherent receiver is apparent if we consider the output of the quadrature receiver for signals with a phase of 0 and 180 degress.

In the first case there is a peak at the sampling instant while in the latter a trough. To avoid poor sampling that arises without prior knowledge of θ we thus use an envelope detector as its output is completely independent of the phase mismatch.

Non-coherent FSK

In binary FSK the signal is given as

$$s_i(t) = \sqrt{\frac{2E}{T_b}} \cos(2\pi f_i t)$$

where $i = 1, 2$ and the transmitted carrier frequency is given by

$$f_i = \frac{n_c + i}{T}$$

The non-coherent receiver for this kind of signal consists of a pair of filters - one matched to each of the transmitted frequencies followed by an envelope detector. The output envelopes are sampled and compared to determine the most likely transmitted symbol as shown.

This form of non-coherent binary FSK is a special case of non-coherent orthogonal modulation for

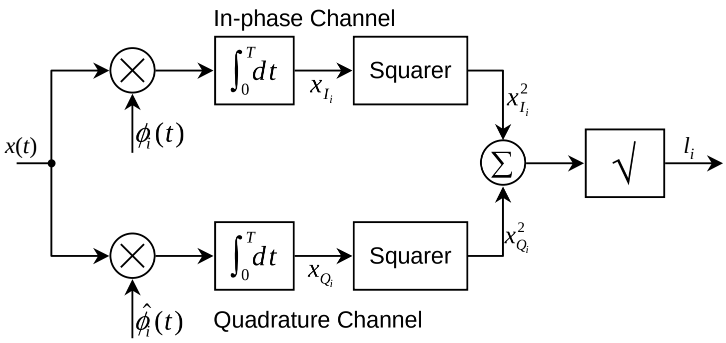

Figure 149: Quadrature Receiver

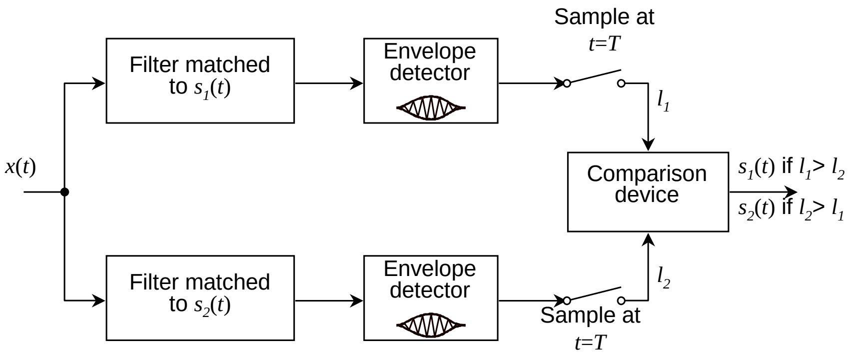

Figure 150: Generalised non-coherent binary receiver

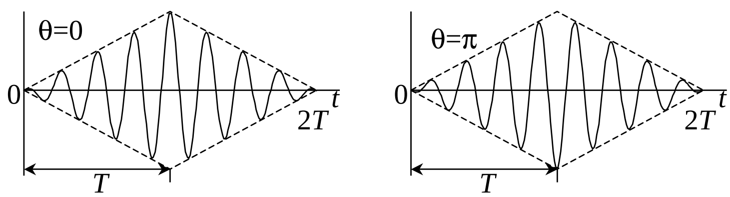

Figure 151: Output of non-coherent matched filter for differing phase mismatches.

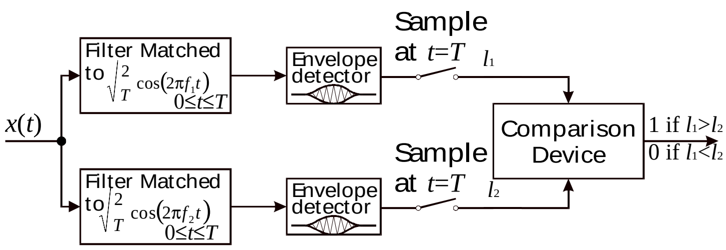

Figure 152: Non-coherent Binary FSK Receiver

which we find the average probability of symbol error (and also the bit error rate) as

$$P_e = \frac{1}{2}\exp\left(-\frac{E_b}{2N_0}\right)$$

Differential Phase Shift Keying (DPSK)

Differential phase shift keying may be viewed as a non-coherent version of PSK. The transmitter eliminates the need for a coherent reference by differentially coding the input binary signal and then employing phase shift keying. Effectively to send a 0 we advance the phase by π and to send a 1 we leave the phase unchanged. The receiver measures the relative phase difference between the waveforms received during two successive symbol intervals. If the phase error changes sufficiently slowly it can be considered to be constant over the two bit intervals.

DPSK may be considered as an example of non-coherent orthogonal modulation when it is considered over two bit intervals. We can define then define the two signals representing symbols 1 and 0 respectively as

$$s_1(t) = \begin{cases} \sqrt{\frac{E_b}{2T_b}}\cos(2\pi f_c t) & 0 \le t \le T_b \\ \sqrt{\frac{E_b}{2T_b}}\cos(2\pi f_c t) & T_b \le t \le 2T_b \end{cases}$$

for where the phase is unchanged between signal intervals and

$$s_2(t) = \begin{cases} \sqrt{\frac{E_b}{2T_b}}\cos(2\pi f_c t) & 0 \le t \le T_b \\ \sqrt{\frac{E_b}{2T_b}}\cos(2\pi f_c t + \pi) & T_b \le t \le 2T_b \end{cases}$$

where there is a π phase change across signal intervals.

These two signals are orthogonal over the interval $0 \le t \le 2T_b$ and the bit error rate for DPSK is thus given by

$$P_e = \frac{1}{2}\exp\left(-\frac{E_b}{N_0}\right)$$

In figure 7) we show an example illustrating the DPSK encoding of a binary signal. Note that the sequence starts with an (arbitrary) reference bit.

The transmitter (figure 153)) consists of a logic network connected to a 1 bit delay element to generated the differentially encoded binary sequence. This sequence is amplitude encoded and used to modulate the carrier signal to generate the DPSK signal.

The optimum receiver for binary DPSK is shown in figure 154).

This receiver equipped with an in-phase and quadrature channel which measures the co-ordinates of the signal at time $t = T_b$ and time $t = 2T_b$ respectively. These two signal points are compared to determine if they map to the same signal point or they are π out of phase.

To complete the test below you may need the Complementary Error Function Tables (Section).

Assessment
Deadline: 2009-11-28 23:59:00
No Questions: 1
Time Allowed: 5 min

Comparison Of Modulation Techniques

We can compare the various modulation techniques on the basis of the SNR required to achieve a specified probability of error subject to some constraint such as a fixed data rate of transmission. The results are shown in figure 155).

For multiple phase signals the channel bandwidth required is simply that of the equivalent low pass signal. We assume that our pulse has duration T and occupies a bandwidth $B = 1/T$. Since $T = (\log_2 M)/R$ so as M is increased the channel bandwidth required decreases and we have a bandwidth efficiency, defined as bit rate to bandwidth ratio of

$$\frac{R}{B} = \log_2 M$$

For PAM the most efficient transmission is single-sideband (SSB) and we have approximately $B = 1/(2T)$ so the bandwidth efficiency is given by

$$\frac{R}{B} = 2\log_2 M$$

In the case of QAM we have two orthogonal carriers each having a PAM signal however the signal must be transmitted via double sideband. Thus QAM and PAM have the same bandwidth efficiency

Table 7: Generation of a DPSK Signal

Message Sequence $\{b_k\}$		1	0	0	1	1	1	0	0	0
Encoded Sequence $\{d_k\}$	1	1	0	1	1	1	1	0	1	0
Transmitted Phase	0	0	π	π	π	π	π	0	0	π

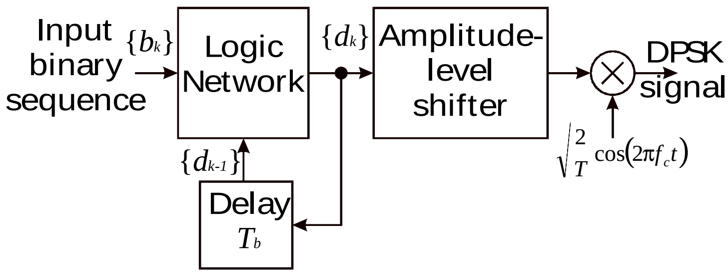

Figure 153: DPSK Transmitter

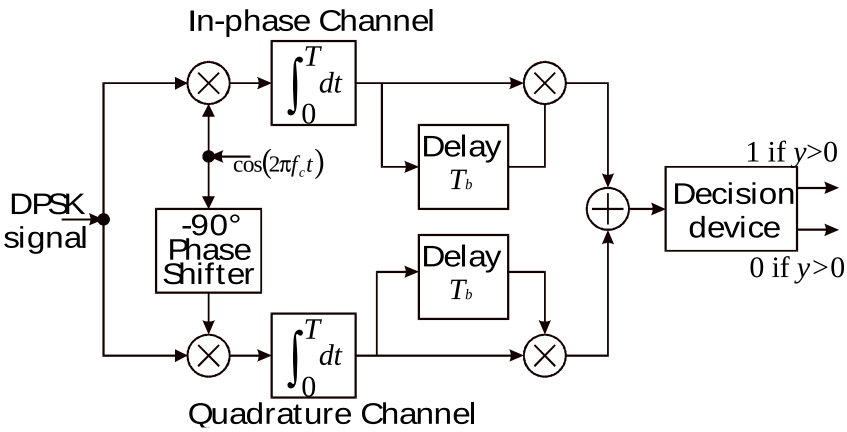

Figure 154: DPSK Receiver

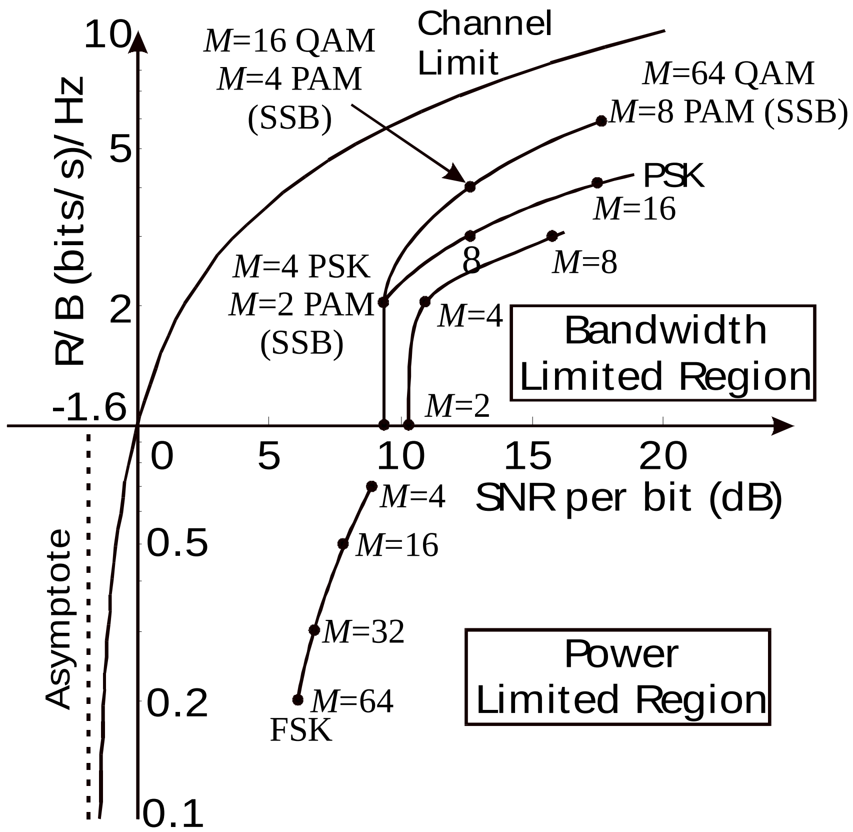

Figure 155: Comparison of modulation techniques at 1×10^{-5} symbol error probability

when bandwidth is referenced to the bandpass signal.

Orthogonal signals (such as FSK) have completely different bandwidth requirements. If we use $M = 2^k$ orthogonal carriers with a minimum frequency separation of $1/(2T)$ for orthogonality the bandwidth is given by $M/(2T)$ giving a bandwidth efficiency of

$$\frac{R}{B} = \frac{2\log_2 M}{M}$$

A meaningful comparison of these results is based on the normalised data rate R/W versus the SNR per bit required to achieve a given probability of error. This is plotted in here for $P_M = 10^{-5}$. We observe that in the case of PAM, QAM and PSK increasing M results in a higher bit rate to bandwidth ratio however this is achieved at the expense in the SNR per bit. Thus these modulation techniques are appropriate for bandwidth limited communication channels where there is sufficiently high SNR to support increases in M. Telephone channels and digital microwave channels are examples of such bandwidth limited channels.

In contrast M-ary orthogonal signals such as FSK have an increasing bandwidth requirement as M increases giving a decreasing channel bandwidth. However the SNR per bit required to achieve a given error probability decreases as M increases. Thus such signals are appropriate for power limited channels that have sufficiently large bandwidth to accommodate a large number of signals. As M increases the error probability can be made as small as required provided that the SNR is grater than the Shannon limit of -1.6dB.

Assessment
Deadline: 2009-11-28 23:59:00
No Questions: 1
Time Allowed: 3 min

Synchronisation in Passband Systems

Carrier Synchronisation

When coherent detection is used knowledge of both the frequency and phase of the carrier is required. If the power spectrum of the signal contains a discrete component at the carrier frequency then a

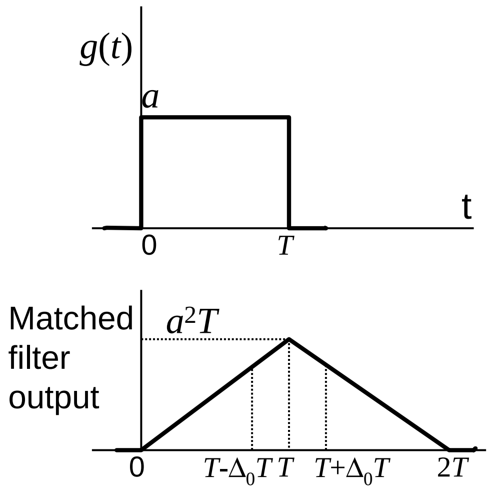

Figure 157: Early-Late detection using a matched filter

narrow-band phased locked loop could be used to provide the carrier reference however such a discrete reference component carries no information and its transmission represents a waste of power.

Without a d.c. component the receiver requires a *suppressed carrier-tracking loop* for providing a coherent subcarrier reference. One example of this, shown in figure 156), is the M th power loop for M-ary PSK. (For $M = 2$ this is called a *squaring loop*). The problem with this type of loop is that there is are M phase ambiguities of in the generated reference. The M th order *Costas Loop* which may also be used for carrier recovery also has this phase ambiguity problem. One method of resolving this is to use differential encoding resulting in *coherent detection of differentially encoded M-ary PSK*. This has a small degradation with respect to the noise performance.

Symbol Synchronisation

To perform demodulation the receiver has to know when symbols start and end so as to determine the correct time to sample and quench the product integrators. Estimation of the symbol timing is called *clock recovery* or *symbol synchronisation*. A clock could be transmitted along with the data signal. This minimises the time for clock recovery but results in a waste of transmitted power.

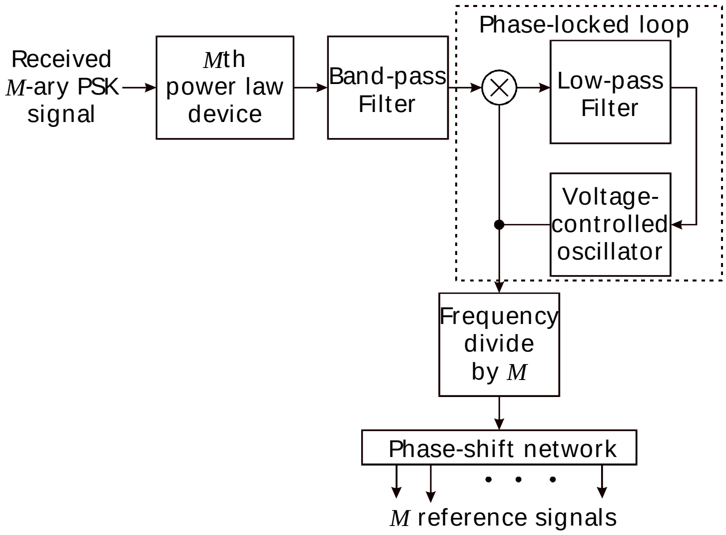

Figure 156: Mth Power Loop

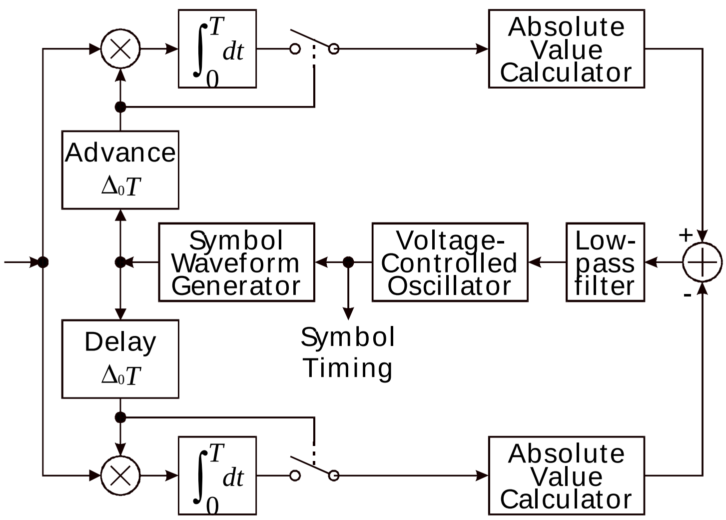

Figure 158: Early-late gate type of symbol synchroniser

A good method for extracting the clock from the demodulated data signal is to use non-coherent detection. If we consider a rectangular pulse as shown in figure 157), we observe that the matched filter output is at a maximum at time $t = T$ and that it is symmetric about that point. We obviously want to sample at that point. If we take an early sample at $T - \Delta_0 T$ and a late sample at $T + \Delta_0 T$ then obviously the two samples will be equal to each other and smaller than the peak value. The *error signal* or difference between these two values will be zero and the proper sampling time is mid-way between the early and late points. Figure 158) shows an example of symbol synchroniser which exploits this idea. Two correlators integrate the data signal integrated over the full time T with one starting early by $\Delta_0 T$ and the other late by $\Delta_0 T$. The difference in the output of these two correlators generates an error signal which is low-pass filtered and used to drive a voltage controlled oscillator at the symbol rate. When this local clock is at the correct frequency we will be at the equilibrium point and the error signal will be zero.

Spread-spectrum Modulation

So far we have been looking at the importance of providing efficient utilisation of bandwidth and power in digital transmission systems. Sometimes these requirements are sacrificed to enable other objectives to be met. For example we may wish to provide secure transmission in a hostile environment so that the signal is not easily detected by unauthorised listeners. One class of signalling techniques used to achieve this is called *spread-spectrum modulation* . The primary advantage of spread spectrum modulation is its ability to reject interference either unintentional or intentional from a hostile jamming transmitter.

Spread spectrum modulation may be defined by

1. Spread spectrum is a means of transmission in which the data sequence occupies a bandwidth in excess of the minimum bandwidth necessary to sent it

2. The spectrum spreading is accomplished before transmission by use of a code that is independent of the data sequence. The same code is used in the receiver (in synchronism with the transmitter) to despread the received signal to recover the original data sequence.

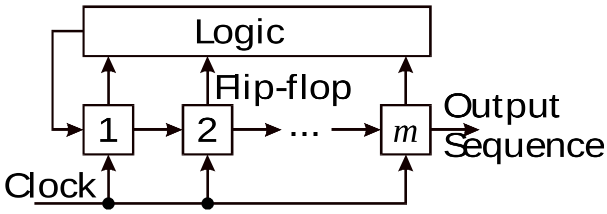

Figure 159: Feedback shift register

Spread spectrum may be not only for its resistance to jamming or interference but may also be used to minimise multiple path effects in ground based mobile radio applications and to allow multiple access communications allowing a number of independent users to share a common channel without external synchronisation.

There are two main types of spread-spectrum modulation - *direct-sequence spread spectrum* where a narrow band code modulates a pseudo-random wideband code which is transmitted and *frequency-hopping spread-spectrum* where the carrier frequency carrying the data is change in a pseudo-random manner. Both these rely on the frequency spreading properties of a noise-like spreading code called a *pseudo-random* or *pseudo-noise sequence* .

Pseudo-Random Sequences

A pseudo-noise (PN) sequence has a noise-like waveform and is usually generated using a *feedback shift register* shown schematically in figure 159). This consists of an shift register on m flip-flop stages and a logic circuit interconnected to form a multiple-loop feedback circuit. A single timing clock regulators the shift register stages and at each pulse the state of each flip-flop is shifted to the next one down the line. With each clock the logic computes a function of the Boolean state of the flip-flops which is fed back into the shift-register input. If the feedback logic consists only of modulo-2 adders then the feedback shift register is said to be linear. In this case the zero state (all flip-flops set

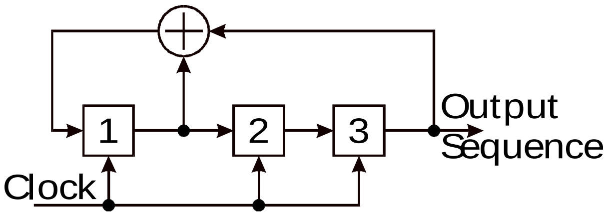

Figure 160: Maximum length m=3 sequence generator

to zero) is not permitted as the feedback logic output would the be zero and the output would be a train of zeros. Thus the period of a PN sequence produced by a linear feedback shift register with m flip-flops cannot exceed $2^m - 1$ and if the period is exactly $2^m - 1$ it is called a *maximum-length-sequence* or *m-sequence*.

Figure 160) shows an example of a maximum length $m = 3$ sequence generator. The input is equal to the modulo sum of the first and third flip-flops. If the initial state of the shift register is 100 the succession of states will be as follows: 100, 110, 111, 011, 101, 010, 001, 110, generating the output binary sequence 00111010 which repeats with a period $N = 2^3 - 1 = 7$.

Maximum length sequences have many of the properties of a truly *random binary sequence* (one in which the probability of 1 and 0 are equal). The most useful property for spread spectrum communications is that they have similar envelopes for their power spectral densities - a sinc^2 function which is continuous for a random binary wave but which comprises a series of delta functions $1/(NT)$ apart for a pseudo-random sequence.

In general we want to find for a particular m the feedback logic to generate a maximum length sequence. This can be done by looking up extensive tables found in the literature.

Assessment
Deadline: 2009-11-28 23:59:00
No Questions: 1
Time Allowed: 5 min

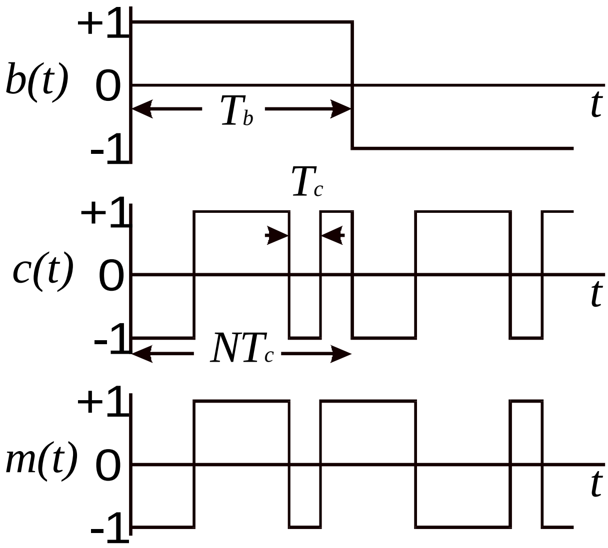

Figure 161: Waveforms illustrating direct-sequence spread spectrum

Direct-Sequence Spread Spectrum

One method of widening the bandwidth of a data sequence is to use modulation, multiplying the narrow band data sequence waveform $b(t)$ by a broadband pseudo-noise sequence signal $c(t)$ to generate the transmitted signal

$$m(t) = c(t)b(t)$$

The product signal will have a spectrum equal to the convolution of the two separate spectra and which will therefore be nearly the same as the wideband PN signal spectrum. Thus the PN sequence is called a spreading code. The process is illustrated in figure 161). We see that information bit is chopped into a number of small time increments (called *chips*).

The received signal $r(t)$ consists of the transmitted signal plus some additive interference $i(t)$ and will be thus be given by

$$\begin{aligned} r(t) &= m(t) + i(t) \\ &= c(t)b(t) + i(t) \end{aligned}$$

To recover the original message signal the received signal is applied to a demodulator consisting of a multiplier followed by an integrator. The multiplier is supplied with a locally generated PN sequence which is an exact replica of that used in the transmitter and is in synchronism with it. The multiplier output will then be given by

$$\begin{aligned} z(t) &= c^2(t)b(t) + c(t)i(t) \\ &= b(t) + c(t)i(t) \end{aligned}$$

where we have made use of the fact that $c^2(t) = 1$ for all t. The original signal is multiplied twice by the PN sequence and is recovered back to its narrowband form while the interference is multiplied once and is spread in spectrum at the multiplier output. An integrator after the multiplier can therefore filter out the original data sequence from the interference.

We thus see that use of the spreading code produces a wide-band transmitted signal which appears noise like to a receiver which has no knowledge of the spreading code. The longer the pseudo-random sequence used the more noise-like the transmitted signal appears and the harder it is to detect.

We have shown baseband direct-sequence spread spectrum for illustration however the technique may be incorporated into other passband modulation schemes such PSK (giving a direct-sequence spread binary phase-shift keyed (DS/BPSK) signal).

An important parameter in spread-spectrum modulation which we may want is the gain in SNR obtained. This is called the processing gain. On the simple argument that the noise is spread in spectrum by $1/T_c$ while the signal has a spectral width of $1/T_b$ we can see that

$$PG = \frac{T_b}{T_c}$$

This expression is properly derived in Haykin p.592

Frequency-hop Spread Spectrum

Another technique for accomplishing spread-spectrum communications is to use a PN sequence to randomly hop the data-modulated carrier from one frequency to the next. In this case the spectrum is spread sequentially rather than instantaneously as for direct-sequence spread spectrum. This is called *frequency-hop (FH) spread spectrum* and a common modulation format is to use M -ary FSK giving rise to the combination technique FH/MFSK

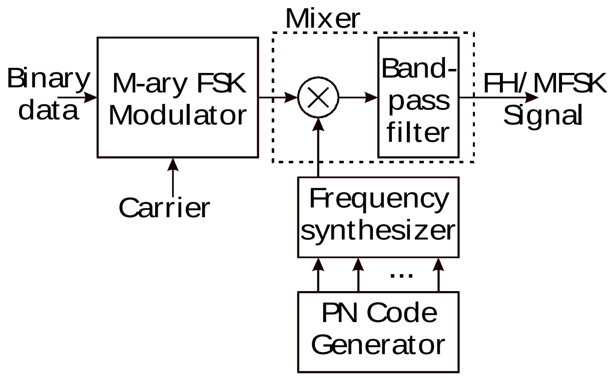

Figure 162: Frequency Hop Transmitter

In *slow-frequency hopping* the symbol rate R of the MFSK signal is an integer multiple of the hop rate R_h that is several symbols are transmitted per frequency hop. Figure 162) shows an example of a transmitter an receiver for achieving this. It involves frequency modulation followed by mixing with a frequency generated by a frequency synthesiser controlled by the pseudo-random code generator. The output is filtered to select the sum term in the mixing signal. A successive k bits from the PN code generator control synthesiser allowing the frequency to hop over 2^k distinct values. For a single hop the spectrum as the same as for the corresponding MFSK signal however over the complete range of 2^k frequency hops the spectrum occupies a much larger bandwidth- up to several GHz with current synthesiser technologies which is an order of magnitude larger than with direct-spread spread spectrum. In the receiver the received signal is mixed down using a frequency synthesiser driven exactly as the one in the transmitter and the resultant output bandpass filter before normal MFSK detection is used.

In *fast-frequency hopping* the hop rate R_h is an integer multiple of the MFSK symbol rate R and the carrier frequency will change several times during the transmission of one symbol. This is used to defeat a smart jammer which might measure the

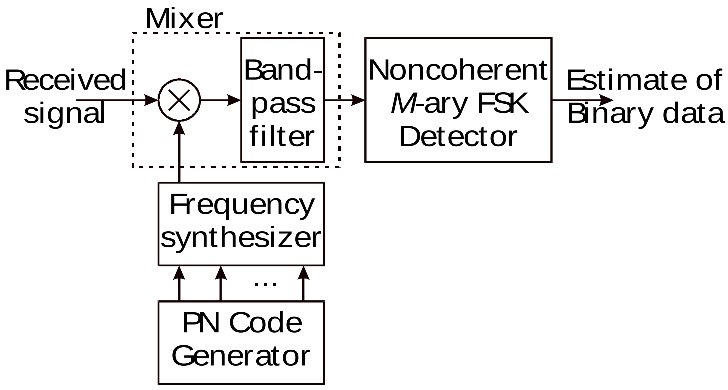

Figure 163: Frequency Hop Receiver

spectral content of the transmitted signal and retune the jamming frequency to that portion of the frequency band. Non-coherent detection is used at the receiver with either a majority rule made on the frequency hop chips of the dehopped MFSK symbol or, optimally, likelihood functions are computed as functions over the total energy received over the K chips of a signal and the larger one selected.

Spread Spectrum Synchronisation

A spread spectrum sequence communication system relies on synchronisation between the locally generated PN sequence at the receiver and that used to transmit the sequence. This may be done in two parts: acquisition and tracking. The acquisition stage aligns the two PN codes quickly by measuring the correlation between the received signal and local PN code. Once aligned to within a fraction of a chip in this way a *decision-rule and search strategy* is used to determine whether the two codes are in synchronism and what to do if they are not. Once aligned tracking is performed using phased-lock loop techniques.

Code-division multiplexing

This technique relies on using separate PN codes for individual users allowing a multitude of users on a common communication channel without frequency or time allocation. However an important concern in a code-division multiplex system is the partial cross-correlation of PN sequences which gives rise to cross-talk between any two users sharing a common communication channel. Maximum length sequences do not generally exhibit a good partial cross-correlation and for this reason so called *Gold Sequences* are used which exhibit a very low cross correlation.

Modems

The word 'Modem' is derived from two words; 'MODulator' and 'DEModulator'. From a data communications perspective, a modem is a device that converts the digital bit stream into analogue signals that are suitable for transport over standard voice circuits.

Typical modems operate using Frequency Shift Keying (FSK), Phase Shift Keying (PSK), Amplitude Shift Keying (ASK), or a combination of basic schemes.

Frequency Shift Keying (FSK)

Bell 103 Modem This modem supports asynchronous rates up to 300 baud using Frequency Shift Keying (FSK) modulation. Different carrier frequencies are used at each end of a 103 modem link, allowing full-duplex operation on a 2-Wire switched voice circuit.

The Originating modem transmits signals of either 1070 Hz (Space) or 1270 Hz (Mark). The Answering modem transmits signals of either 2025 Hz (Space) or 2225 Hz (Mark).

This modem was predominant until the early

1980s, when the Bell 212 modems became available.

V.21 Modem This modem supports asynchronous transmission at rates up to 300 baud. Modulation is Frequency Shift Keying (FSK). In this modulation scheme, different carrier frequencies are used between the Originating and Answering modems. A Space is transmitted by a change in carrier frequency of +100 Hz. A Mark is represented by a change in carrier frequency of -100 Hz. For the carrier frequency of 1080 Hz, a Space is represented by a 1180 Hz signal and a Mark is represented by a 980 Hz signal. For the carrier frequency of 1750 Hz, a Space is represented by a 1850 Hz signal and a Mark is represented by a 1650 Hz signal.

This modem modulation scheme is specified in CCITT (now ITU-T) Recommendation V.21, naturally.

Bell 202 Modem This Bell System modem was designed to support asynchronous data at rates of up to 1200 baud on 2-wire dial-up circuits, and up to 1800 baud on conditioned leased lines. Operation is half-duplex and the modem modulation scheme used is Frequency Shift Keying (FSK).

A Mark is represented by a frequency of 1200 Hz, while a Space is represented by a frequency of 2200 Hz.

This modem specification also describes an optional 'Reverse' channel, for slow, general purpose use. While the specification calls for this 'Reverse' channel to operate at 5 BPS, many modem manufacturers implement 75 to 150 baud versions.

Phase Shift Keying (PSK)

201 Modem These Bell System modems support synchronous data rates up to 2400 BPS. Full-duplex is available, but the modem operates in a Half-Duplex mode when using a 2-Wire switched voice circuit. This modem uses a modulation scheme that encodes data by using four specific 'phase shifts' of the transmitted carrier. This type of modulation is known as DPSK (Differential Phase Shift Keying), but is sometimes called QPSK (Quad Phase Shift Keying).

In this modulation scheme, two bits (called a 'dibit') are represented with a single phase change:

00 = 45 degrees 10 = 135 degrees 11 = 225 degrees 01 = 315 degrees

The actual modem modulation rate is 1200 BAUDS, with each BAUD capable of supporting two data bits.

The CCITT (now 'ITU-T') has specified a compatible modulation scheme in Recommendation V.26, Alternative 'B'.

V.26 Modem CCITT Recommendation V.26, Alternative 'A' describes a modulation type that is similar to the Bell 201 series of modems, except that different phase shift values are specified to encode the 'dibit':

00 = 0 degrees 10 = 90 degrees 11 = 180 degrees 01 = 270 degrees

The actual modem modulation rate is 1200 baud, with each baud capable of supporting two data bits.

In the second release of the V.26 Recommendation (called 'V.26bis'), fallback operation to 1200 BPS is defined. V.26bis also recommends that the modulation type used be 'Alternative B', as described above (Bell 212). The third iteration of this Recommendation (called 'V.26ter') incorporates echo cancellation techniques within the modem to delineate the Originating and Answering modem signals. V.26ter allows full-duplex operation on a 2-Wire, switched, voice circuit.

Bell 212 Modem This Bell System modem supports either asynchronous or synchronous data rates up to 1200 BPS on switched 2-wire dial-up circuits. Operation is full-duplex with different carrier frequencies used between the Originating and Answering modems. The modulation method employed is DPSK, using 'dibits' to represent up to four phase changes.

The actual modem modulation rate is 600 baud, with each baud consisting of two data bits.

The 212 modem series also incorporates a 300 baud, Type 103, modem for compatibility with preexisting Bell 103 modems.

This modem type was widely used in the early to mid 1980s, and is still commonly found in use today.

V.22 Modem This modulation is described in CCITT Recommendation V.22 and FED-STD-1008! It utilizes Differential Phase Shift Keying

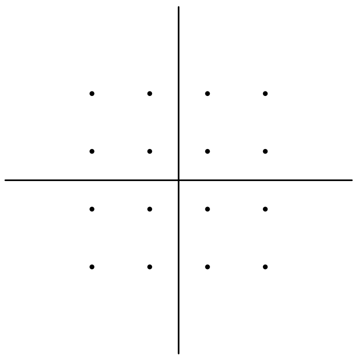

Figure 164: V22 modem signal space

(DPSK) and is designed for synchronous or asynchronous operation at 1200 BPS. Operation is full-duplex with different carrier frequencies used between the Originating and Answering modems.

The modem modulation rate is 600 bauds, with each baud representing two data bits.

This modem modulation scheme supports a 'fall-back' rate of 600 BPS.

00 = 90 degrees 10 = 0 degrees 11 = 270 degrees 01 = 180 degrees

V.22bis Modem The 'bis' qualifier is a French (also, Latin) term for 'duo' or 'twice'. Thus, as the name would suggest, this modulation scheme is described in the second release of CCITT's V.22 Recommendation.

This modulation scheme supports transmission of full-duplex 2400 BPS synchronous or asynchronous data over a switched, 2-Wire, voice circuit. Alternatively, these modems may be employed on leased-lines as well. The modulation scheme used is QAM (Quadrature Amplitude Modulation). In this modulation scheme, the data stream is divided up into groups of four bits, known as 'quadbits'. The first two bits of each 'quadbit' are encoded as a phase change, changing the 'quadrant' (except in cases where the first two bits are '01'; in this case, the 'quadrant' is not changed from its previous state). The second two bits of each 'quadbit' define one of four signal states in the new 'quadrant'.

The modulation rate is 600 bauds, with each baud representing four data bits.

These modems support fallbacks to V.22 modulation schemes also. Most of the popular V.22bis PC modems support fallback operation to Bell 212 modulation also, depending upon the capabilities of both Originating and Answering modems.

Although the V.22bis specification was defined in 1984, practical deployment of these modems did not occur until the late 1980s.

V.27 Modem CCITT Recommendation V.27 describes a modulation scheme that is capable of supporting 4800 BPS, full-duplex, synchronous data. Operation may be full-duplex on a 4-Wire leased line or half-duplex on a 2-Wire, switched, voice circuit. The modulation scheme is known as D8PSK (Differential 8 Phase Shift Keying) and operates by breaking the incoming data stream into groups of three bits ('tribit'). These 'tribits' are represented by one of eight possible phase shifts:

001 = 0 degrees 000 = 45 degrees 010 = 90 degrees 011 = 135 degrees 111 = 180 degrees 110 = 225 degrees 100 = 270 degrees 101 = 315 degrees

The modulation rate is 1600 bauds, with each baud representing three data bits.

The second and third releases of the V.27 Recommendation (V.27bis and V.27ter, respectively) added the ability to fallback to a 2400 BPS rate using V.26, Alternative 'A' modulation. Also, the start-up/training times are reduced in the V.27bis Recommendation.

V.29 Modem This modulation scheme was first standardised by the CCITT in 1976. It uses a form of Quadrature Amplitude Modulation (QAM), which transports data in groups of four bits ('quadbits'). The first bit determines the amplitude of the signal while the next three bits represent one of eight phase changes. The phase shifts are similar to the 'tribit' modulation scheme described in Recommendation V.27:

001 = 0 degrees 000 = 45 degrees 010 = 90 degrees 011 = 135 degrees 111 = 180 degrees 110 = 225 degrees 100 = 270 degrees 101 = 315 degrees

V.29 modulation is capable of transporting synchronous data at rates up to 9600 BPS. It operates full-duplex on a 4-Wire leased line or half-duplex on a 2-Wire, switched, voice line.

The V.29 modulation rate is 2400 bauds, with each baud representing four data bits.

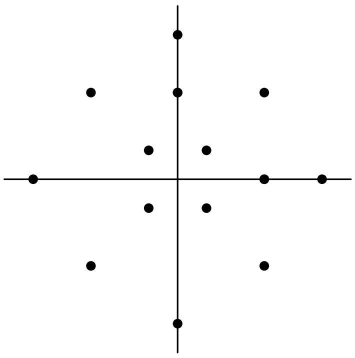

Figure 165: V29 modem signal space

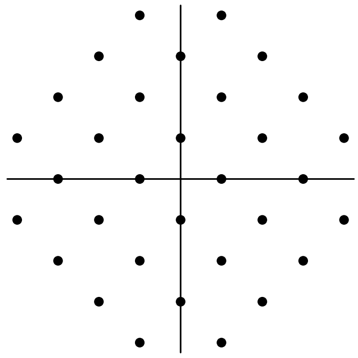

Figure 166: V32 modem signal space

V.29 modulation also incorporates fallback to 7200 BPS. In this mode, three bits ('tribit') are combined to represent one of eight possible phase changes. In this mode of operation, the modem's modulation rate is still 2400 baud, but each baud now represents three data bits.

V.29 modulation also incorporates fallback to 4800 BPS. In this mode, two bits ('dibit') are combined to represent up to four phase changes (0, 90, 180, and 270 degrees). The modem's modulation rate remains at 2400 baud, but each baud now represents only two data bits:

00 = 0 degrees 10 = 90 degrees 11 = 180 degrees 01 = 270 degrees

V.29 modems were highly popular in the 1970s and 1980s for use on 4-Wire leased lines and are still found in use today. The Group 3 FAX machines that are popular today operate in a half-duplex fashion using V.29 modulation.

V.32 Modem First defined in 1984 by the CCITT, V.32 defines a modem that can support 9600 BPS asynchronous or synchronous data. Operation is full-duplex over a 2-Wire, switched, voice circuit. The modulation used may be Quadrature Amplitude Modulation (QAM), or QAM with Trellis coding. Trellis coding is actually a Forward Error Correcting (FEC) scheme.

The modulation rate is 2400 baud, in both 'Nonredundant' and 'Trellis' modes of operation. In the 'Nonredundant' mode, each baud represents four bits. In the 'Trellis' mode, each baud represents five bits; the four data bits, plus a coded, redundant bit that is the result of convolutional coding of the first two bits in the previous 'quadbit' modulation process.

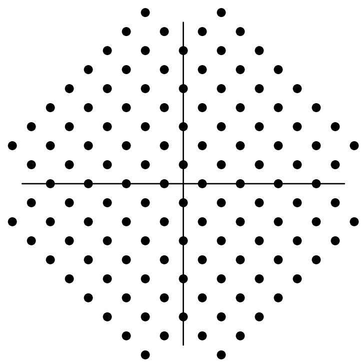

Figure 167: V33 modem signal space

The use of echo cancellation techniques allows the same carrier frequency (1800 Hz) to be used at each end of a modem system.

Delays in the development of cost-effective echo cancellation techniques resulted in practical deployment of V.32 modems in the early 1990s.

V.32bis Modem This standard modulation scheme was developed by the CCITT at the end of the 1980s (1988), although practical deployment of such systems did not occur until the early 1990s. This modulation method allows the transport of asynchronous or synchronous data at line rates up to 14400 BPS (14.4 KBPS). Operation is full-duplex over 2-Wire, switched, voice circuits, using echo cancellation techniques to differentiate between the Originating and Answering modem's signals.

The modulation rate is 2400 bauds, and use the Forward Error Correcting (FEC) capabilities of Trellis coding. Quadrature Amplitude Modulation (QAM) is employed, using groupings of seven bits. Only six of these bits contain actual user data, the remaining bit is the convolutional coded, redundant bit generated from the previous bits.

V.FC Modem V.FC (or 'V.Fast Class') is a non-standard 28.8 KBPS modulation scheme developed by, and proprietary to, Rockwell. It was first released in 1993 and has enjoyed some success. As a mature product, most of the 'bugs' have been worked out, thus improving reliability. However, the majority of the industry has been awaiting the recent release of the new CCITT V.34 Recommendation.

It is capable of supporting full-duplex synchronous or asynchronous data and supports either 4-Wire leased lines or 2-Wire switched voice circuits.

V.34 Modem Approved in the summer of 1994 was the new 28.8 KBPS modulation scheme described in CCITT (ITU-T) Recommendation V.34. During the development of this modulation method, this scheme was known as 'V.FAST'.

It is capable of supporting full-duplex synchronous or asynchronous data over 4-Wire leased lines or 2-Wire circuits.

The modulation rate (baud or 'symbol' rate) can vary. The carrier frequency can vary. The V.34 Recommendation also describes a 'line probing' process that allows the modem to automatically setup optimally for any type of line connection. The training time has been reduced, but the modem will recover automatically from most line disturbances.

Multi-dimensional Trellis coding is employed for robust Forward Error Correcting. A 'Reverse Channel' option is also described in V.34!

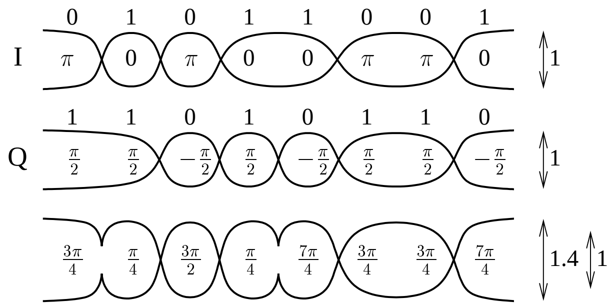

Figure 168: QPSK in a band-limited channel

Advanced Modulation Formats And Applications

In the tutorials on Digital Baseband Transmission (Section) and Digital Passband Transmission (Section) we have analysed the basic techniques for digital signal transmission. We have looked at their bit rate to bandwidth efficiency, the effects of noise and inter-symbol interference in the ideal case. Many modern applications require larger bit rates much closer to the theoretical maximum, and in this case we do have to be concerned by the non-ideal characteristics such as non-linear effects and multiple-path reception. To reduce these problems more advanced modulation schemes, based on those considered are often used. The detailed analysis of these schemes is considered beyond the scope of this tutorial, and we will be taking a descriptive approach using application examples.

Continuous Phase Modulation

Previously when looking at Coherent M-ary Phase Shift Keying (MPSK) (Section) we considered it as two indepandlty tnasmitting binary PSK channels each of which is modulated with ideal bipolar non-return-to-zero symbols. In such the phase transition between symbols is instantaneous, and the amplitude, and thus power transmitted is a constant. Real channels are however bandlimited with the result shown in figure 168).

The change in amplitude from +ve to -ve now not instantaneous but will take some time due to

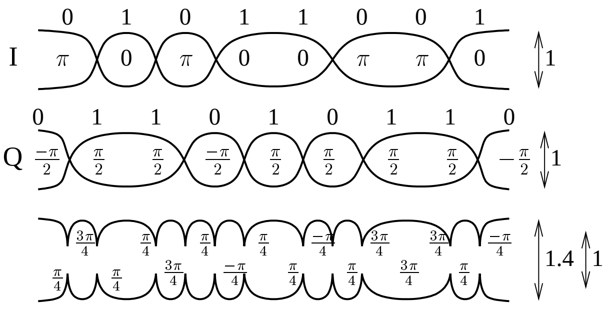

Figure 169: OQPSK in a band-limited channel

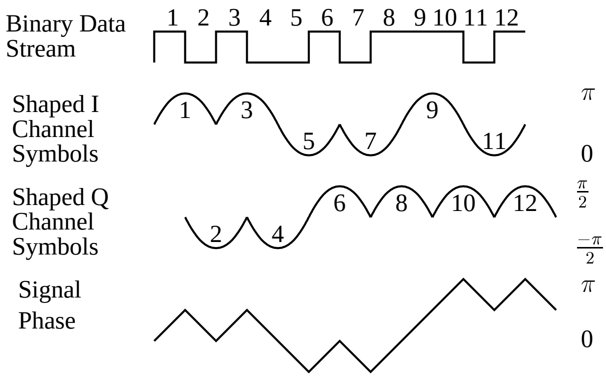

Figure 171: MSK Waveforms

the filtering of the channel. The result is that during the transition the amplitude of the channel will decrease to zero. If both channels change phase (sign) at the same time (an overall phase change of π)the total power will, for a moment, go to 0. If only one channel changes then the power will be momentarily halved (the signal amplitude reduced by $1/\sqrt{2}$). If the system is nonlinear i.e. has a response depending on power then this fluctuation in power will lead to spectral spreading and signal will occupy a larger bandwidth than would be needed in the ideal linear case.

Offset Quadrature Phase Shift Keying (OQPSK)

As mentioned previously the power in a bandlimited QPSK falls to zero if the two channels change phase simultaneously but only falls to a half if only one channel changes. Preventing both channels changing simultaneously would therefore reduce the power fluctuations and the consequent impairments. Offset Quadrature Phase Shift Keying (OQPSK) acheives this by simply delaying one of the channels by half a symbol time (the bit time). Thus the maximum phase change at any one time in the signal is $\pi/2$ and we have the result is in figure 169).

Clearly we have reduced the amplitude of the power fluctuations in the signal and we will this reduce the power spreading in a nonlinear system.

Minimum Shift Keying

We see that although QPSK reduces the power fluctuations it doesn't eliminate them. With minimum shift keying we can acheive this by effectively allowing the phase to change slowely over the entire symbol time. We acheive this by applying sinusoidal pulse shaping to the I and Q channels prior to multiplication by the carrier i.e. we transmit sinusoidal pulses. The transmitted signal is then given by

$$\begin{aligned} f(t) =& a_n \sin\left(\frac{2\pi t}{4T_b}\right)\cos 2\pi f_c t \\ &+ b_n \cos\left(\frac{2\pi t}{4T_b}\right)\sin 2\pi f_c t \end{aligned}$$

and the transmitter would be as shown in figure 170).

We note that the power of this signal will be constant and the phase changes continuously and linearly over the symbol time. A linear change in phase over time is course just a frequency shift, and this is modulation technique is exactly equivalent to differential binary Coherent Binary Frequency Shift Keying (FSK) (Section) but with the frequencies seperated by a half cycle in the symbol period - half as much as what we had when we looked at it previously. This is illustrated in figure 171).

If we look at the power spectrum for this signal (figure 172)) we find that it falls off at -12 dB per octave compared to -6 dB per octave for QPSK or OQPSK.

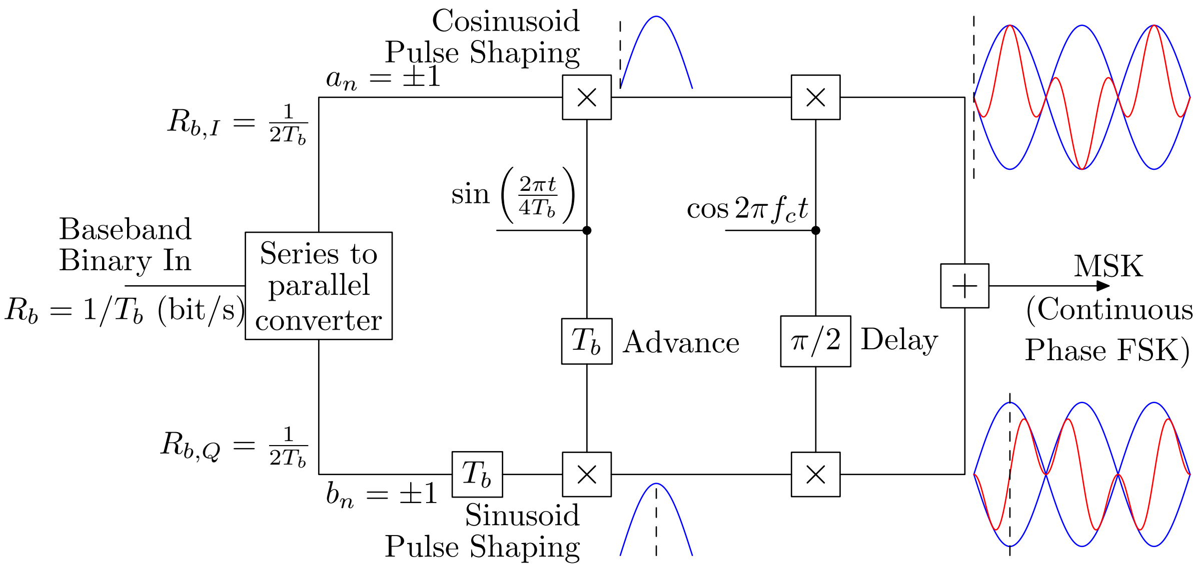

Figure 170: An MSK Transmitter

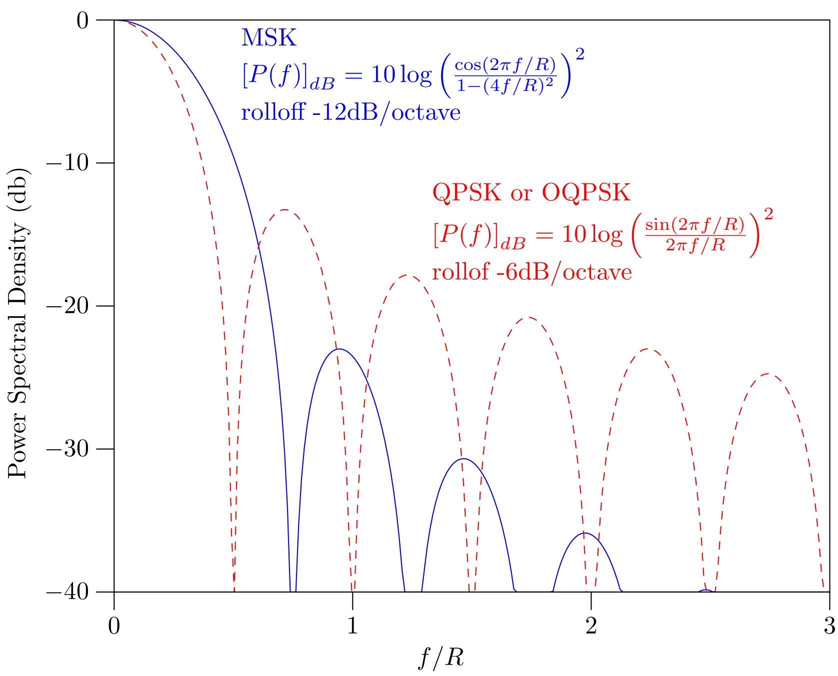

Figure 172: Power Spectrum Comparison between OQPSK and MSK

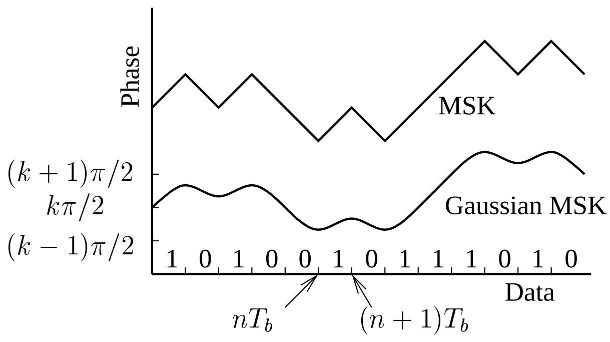

Figure 173: GMSK Phase Change

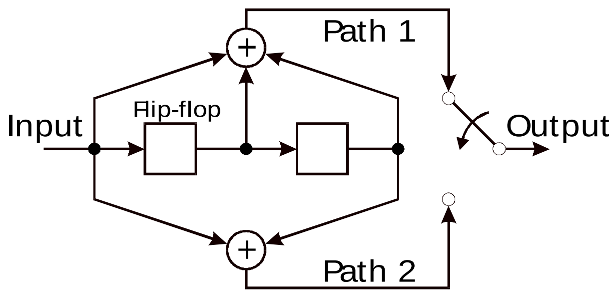

Figure 176: A constraint length 3 rate 1/2 convolutional code

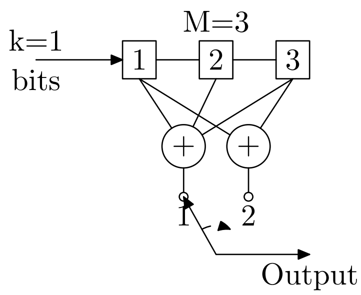

Figure 177: Alternative state machine representation of the coder in 176)

Gaussian Minimum Shift Keying

We can take this whole idea of slowely changing the phase further by using other functions to spread out the change in phase. A good function to use is the Gaussian hence we have Gaussian Minimum Shift Keying (GMSK). Now the width of the Gaussian spreading function can be varied and the change in phase can be spread over a time period wider than the symbol period. This is not the case with sinusoidal MSK. We now also acheive not only continuous phase but a continuous derivative in the phase change i.e. the frequency sweeps slowy between extremes rather than switching suddenly. See figure 173).

The result is a further decrease in the sidelobes in the power spectrum as shown in figure 174). Indeed we can optimise the Gaussian spreading width for different applications. In radio frequency applications this reduction in the sidelobes of the power spectrum will reduce co-channel interference. This is of particular importance in cellular mobile systems and it is for this reason than GMSK is used in the GSM digital cellular radio systems.

Convolutional Coding

Figure 175) shows a non-recursive finite state machine representation for a convolutional code used to introduce a dependency between successive signal points such that only certain patterns or sequences of signal points are permitted. The code is made up of Mk stages of shift registers. At every clock tick k information bits are shifted into the registers. The n output bits from the code are each determined by a linear summation of the register bits. The logic determining which bits of the register are added for which output bits determines the specific code. Such a code depends on $(M-1)k$ previous information bits (this is called the *constraint length*). The code rate is the ratio of input to output bits k/n.

Figures 176) and 178) show specific examples of two convolutional codes.

In order to determine the output we need simply trace the transition in stored state for each input bit and determine the outputs using binary addition. If we take the finite state machine illustrated in figure 176) as an example for an input sequence of `101100` we get the following transitions (note we normally assume an initial starting state of all zeros).

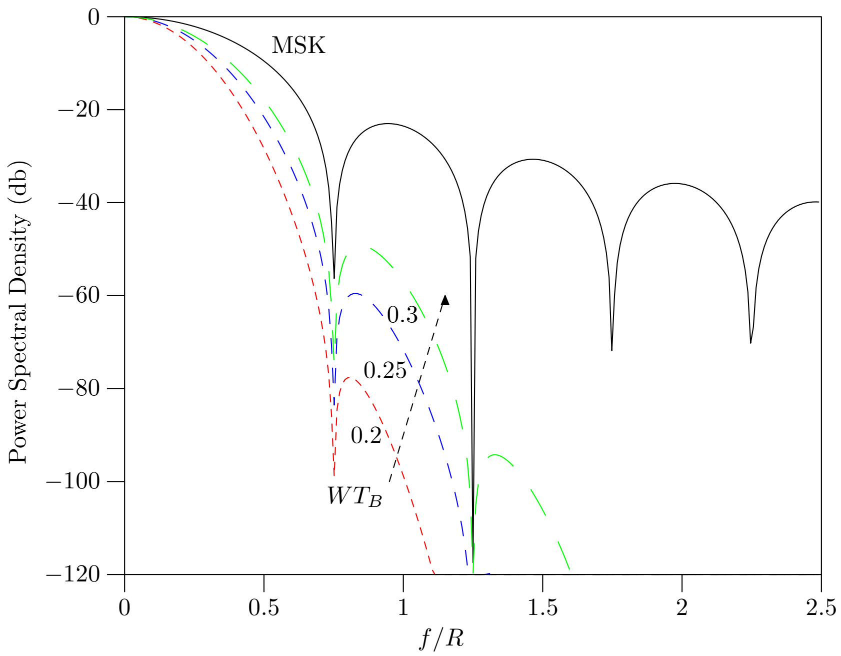

Figure 174: Power Spectrum Comparison between GMSK and MSK

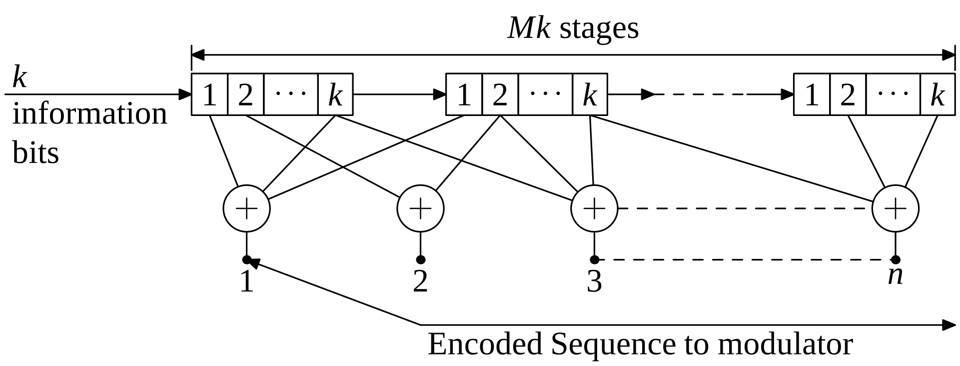

Figure 175: Finite state machine representation of a generic convolutional code

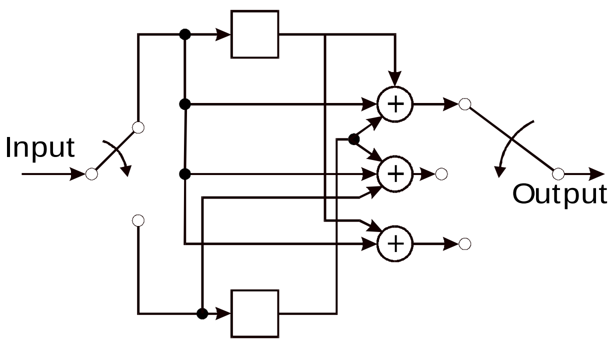

Figure 178: A constraint length 2 rate 2/3 convolutional code

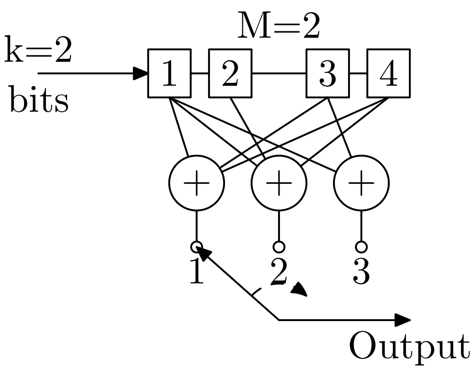

Figure 179: Alternative state machine representation of the coder in 178)

Table 8: Encoding the binary sequence `101100` using the convolutional code of figure 176)

input	**initial state**	**output**	**new state**
1	00	11	10
0	10	10	01
1	01	00	10
1	10	01	11
0	11	01	01
0	01	11	00

Table 9: Encoding the binary sequence `101100` using the convolutional code of figure 178)

input	**initial state**	**output**	**new state**
10	00	110	10
11	10	000	11
00	11	011	00

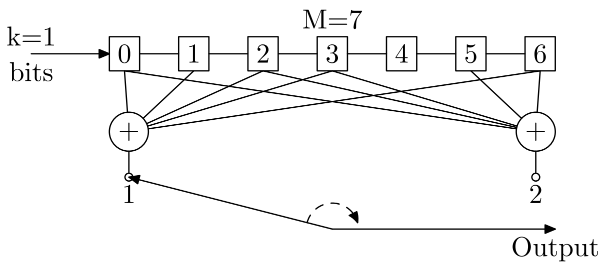

Figure 180: NASA rate 1/2 constraint length 7 convolutional code

With the resulting output of `111000010111`. Notice we input 1 bit at a time and output 2 bits making it this a rate 1/2 convolutional code and that the output depends on 3 inputs giving it a constraint length of 3.

And if we take the same input sequence for the finite state machine illustrated in figure 178) as an example for an input sequence of `101100` we get the following

Giving an output of `111000010111`. Note that this convolutional encoder inputs 2 bits and outputs 3 bits at a time making it rate 2/3, and that the output depends on the current and one previous input sequence in the state machine giving it a constraint length of 2.

An example of a convolutional code in use is the rate 1/2 constraint length 7 convolutional code shown in figure 180). This code was implemented in hardware in several space missions including the Voyager missions. More recent missions have used a constraint length 15 code implemented partly in software.

All of the convolutional codes illustrated so far have as their input the unencoded sequence - they are *non-recursive*. An alternative is to feedback

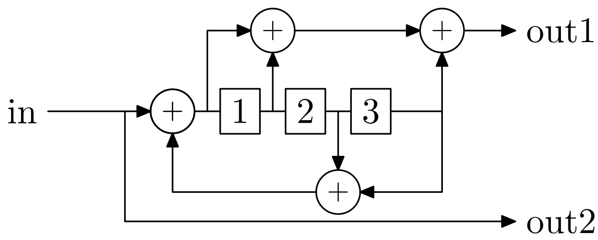

Figure 181: A rate 1/2 recursive convolutional code

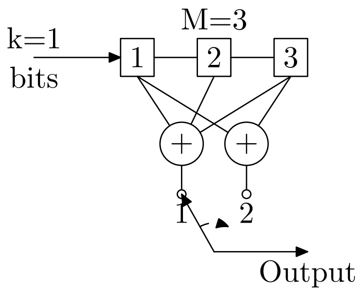

Figure 182: Finite State Machine for a constraint length 3, rate 1/2 convolutional code

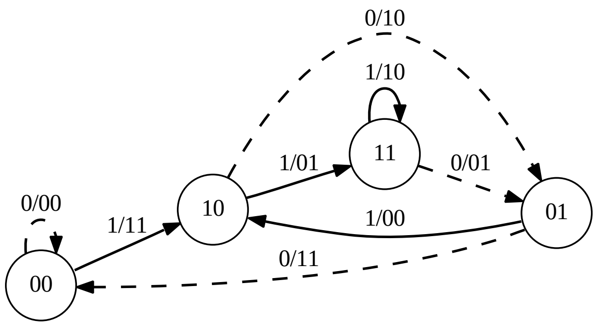

Figure 183: State diagram for the code in figure 182).

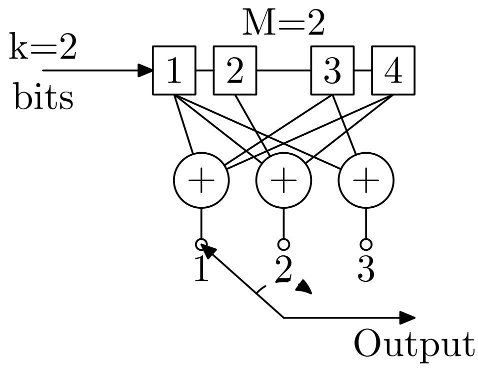

Figure 184: Finite State Machine for a constraint length 2, rate 2/3 convolutional code

some of the registers cells into the input in what is called a *recursive* code as shown in figure 181). This figure also shows a *systematic* code - one in which the encoded sequence is part of the output code. In practice most systematic codes are recursive and most non-systematic codes are non-recursive.

Assessment

Encoding binary sequences using finite state machines.

Deadline: 2009-12-12 23:59:00
No Questions: 2
Time Allowed: 10 min

State Diagram

Convolutional codes consist of a finite state machine and may therefore be represented using a state diagram. Figure 183) shows a state diagram corresponding to the convolutional coder state machine in figure 182). Using the terminology from section Convolutional Coding (Section) the code has a state represented by $(M-1)k$ bits of previous state information and has k bits governing the transition to a new state and the combination of the two determines the output. Thus in the state diagram (figure 182)) we have four states given by two stored bits represented by the circles and from each state we have two transitions corresponding to the 0 or 1 of the input bit which determines the two output bits and the new state. In this diagram the *0* bit transitions are represented using dashed lines and the *1* transitions using solid lines, with each transition labelled with the input bit followed by the output bits. The state diagram provides a complete static view of a convolutional code.

Similarly figure 185) shows a state diagram corresponding to the convolutional coder state machine

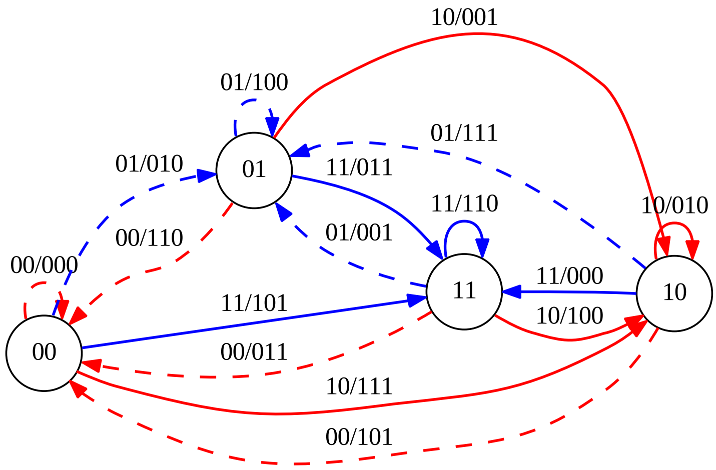

Figure 185: State diagram for the code in figure 184).

in figure 184). In this example, because there are two input bits at a time there are four possible transitions from each state each of which produces 3 bits of output. These are coded using colors and dashed lines. Clearly the constraint length 7 NASA code would have a state diagram with $2^6 = 64$ states and would be too big to show in its entirety here and the constraint length 15 code is 256 times more complex again. The complexity of a convolutional code grows exponentially with its constraint length. While there are easy to implement in terms of encoding as we shall see this complexity places limitations on practical constraint lengths that can be optimally decoded.

Assessment
Encoding binary sequences using state diagrams.
Deadline: 2009-12-12 23:59:00
No Questions: 2
Time Allowed: 10 min

Trellis

The state diagram for convolutional codes discussed in section State Diagram (Section) is a static picture. For the purposes of decoding convolutional codes a more dynamic view is required. This is done by unravelling the state diagram and along a time axis to form a trellis diagram. For our exemplar constraint length 3, rate 1/2 code shown in figure 183 the corresponding trellis is shown in figure 186). The set of four states are represented by vertical positions on the graph and along the horizontal axis we have "levels" corresponding to incoming bits. The transitions from each state at one point in time are joined to the appropriate transitions at the subsequence level. Dashed (red) transitions correspond to an incoming *0* and solid blue transitions to a *1*.

Clearly encoding of an incoming sequence simply corresponds to tracing a path, starting from the initial state (all zeros usually) and following the transitions corresponding to the incoming bits. Figure 186) shows the path traced when encoding the bit sequence *1001101*. From this we see the encoded output *11101111010100* and that the final state reached is *01*.

The real advantage of the trellis diagram is however that we can use it for decoding as demonstrated in section *viterbi-algorithm*.

The Viterbi Algorithm

Viterbi decoding is a maximum likelihood decoding algorithm for convolutional codes. It uses the code trellis. The difficulty in decoding trellis codes is that the number of possible routes grows exponentially. The Viterbi algorithm reduces the number

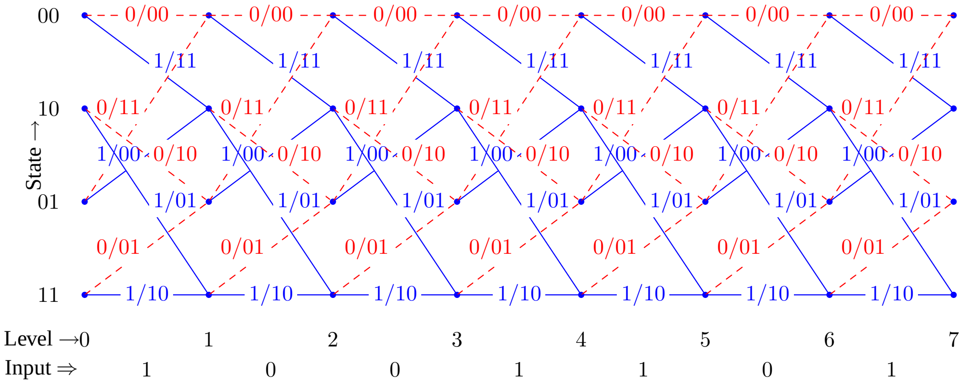

Figure 186: Trellis for constraint length 3, rate 1/2 convolutional code.

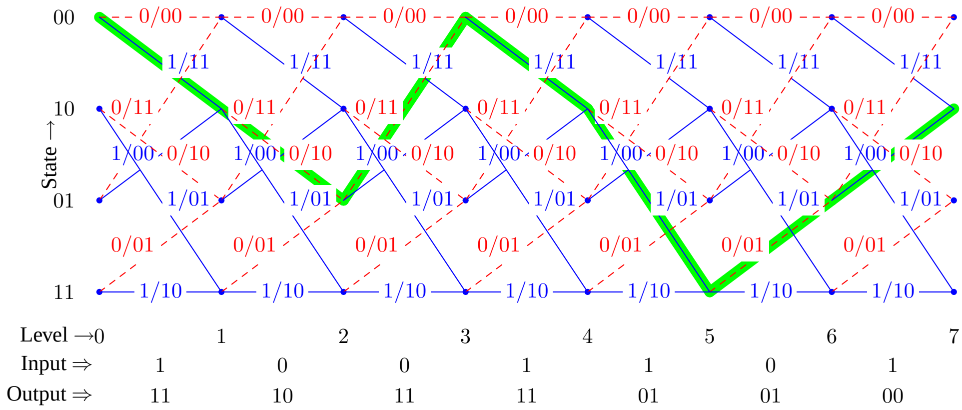

Figure 187: Data encoding of sequence *1001101* using the trellis

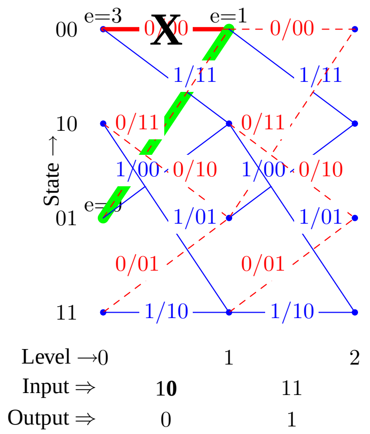

Figure 191: Viterbi decoding - merging paths

of nodes that have to retained to a maximums size of 2^S where \$S\$ is the number of encoder states by merging nodes. It is best explained by illustration.

Figure 188) shows the trellis for our usual constrain length 3 rate 1/2 code. The input in this case is an encoded sequence which has one error (marked in bold) and we are start with an initial state of *00*.

The first part of decoding is to trace the possible paths through the trellis and keeping track of the accumulated likelihood of each path. In this case we are going to use the Hamming distance however soft decision decoding using accumulated probabilities may also be used. Starting from the initial state there are two possible transitions, *00* or *11* leading to the states *00* and *10*. With the incoming sequence of *11* the hamming distances corresponding to the two transitions are 2 and 0 respectively.

We can the continue keeping track of the possible routes with the next input sequence of *10* where we now have four possible paths leading to the four states accumulated Hamming distance ranging from 0 to 3.

With the next two bits of input we find that we may have several paths leading to the same state. Figure 191) shows one example. The Viterbi algorithm is that we retain the maximum likelihood path i.e. that with the least number or probability of errors. In the example we have two transitions to state 0 - one which would have 4 errors and one which would have 1 error. We retain the one with one error. We do this at all points.

Figure 192) shows the full set of routes through the trellis. The most probable one has a Hamming distance of 1 (since there was one error in out encoded sequence) and is marked in Green. We can therefore trace its path and determine at each stage what the decoded sequence would be from the transitions. That is Viterbi decoding. In our simple example we only need to keep track of the four paths to the four states. It can also be shown that you do not to wait for the entire transmitted sequence to decode the sequence - there is a *truncation length*. With the example above we see that once we by the time we reached the end the all the surviving routes had converged for the first two transitions. However it is still the case that the complexity of Viterbi decoding is prohibitive for long constraint lengths - the constraint length 15 code now used by NASA requires specialised hardware called the BVD (Big Viterbi Decoder).

Print off the following Trellis diagrams to use for completing the assessment.

Assessment

Decoding convolutional encoded sequences using the Viterbi algorithm. You may wish to print copies of the trellis before starting this questionnaire.

Deadline: 2009-12-12 23:59:00

No Questions: 3

Time Allowed: 100 min

Trellis Coded Modulation

So far we have considered the channel coding process as being performed separately from the modulation, and likewise for detection and decoding with error control being provided by additional redundant bits in the code thus lowering the information bit rate per channel bandwidth. By treating coding and modulation as a single process we can achieve a more effective utilisation of available bandwidth and power. The coding then involves imposing certain patterns on the transmitted signal. One way of combing modulation and error control is called *Trellis-coded modulation* (TCM) which has three features

1. The number of signal points in the constellation used is large than what is re-

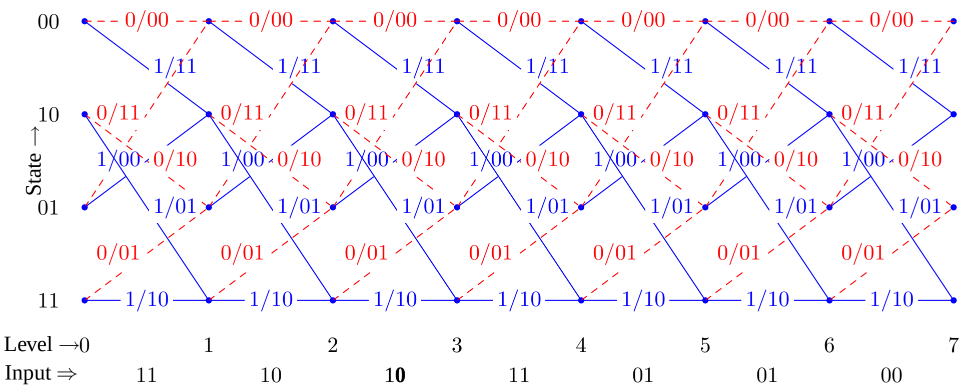

Figure 188: Trellis for Viterbi Decoding an encoded sequence with one error.

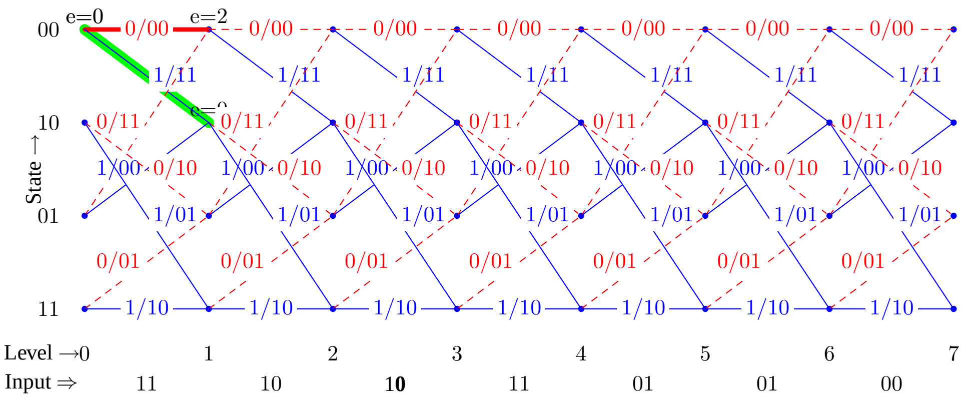

Figure 189: First stage of decoding the sequence

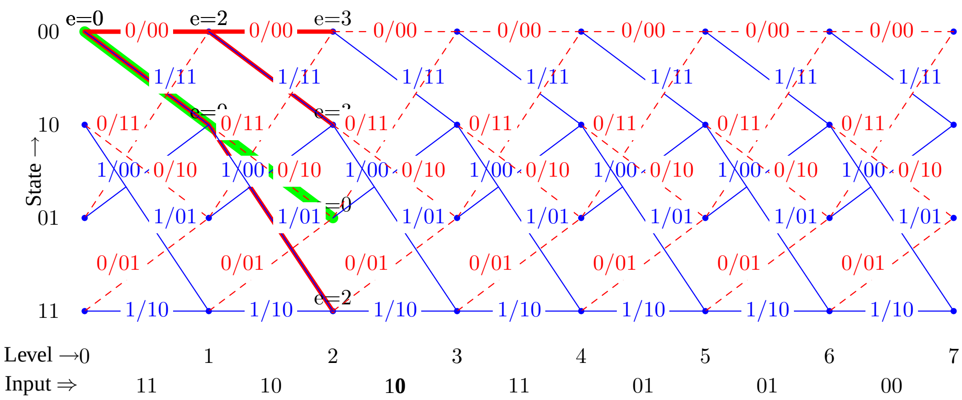

Figure 190: Several stages of Viterbi decoding.

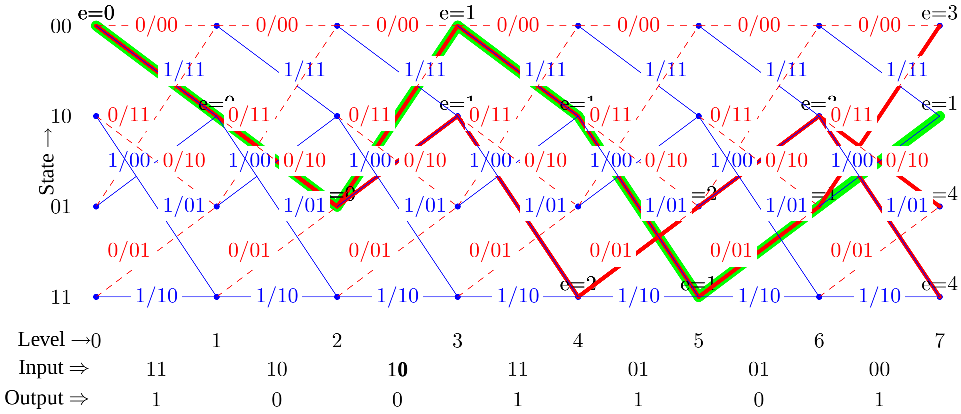

Figure 192: The fully Viterbi decoding.

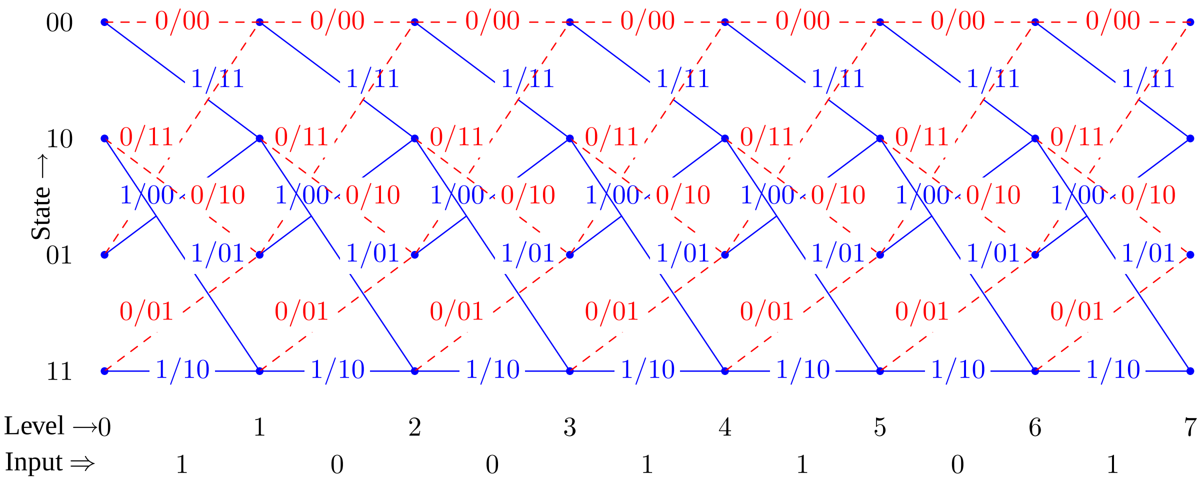

Figure 193: Trellis for decoding a constraint length 3 rate 1/2 convolutional code

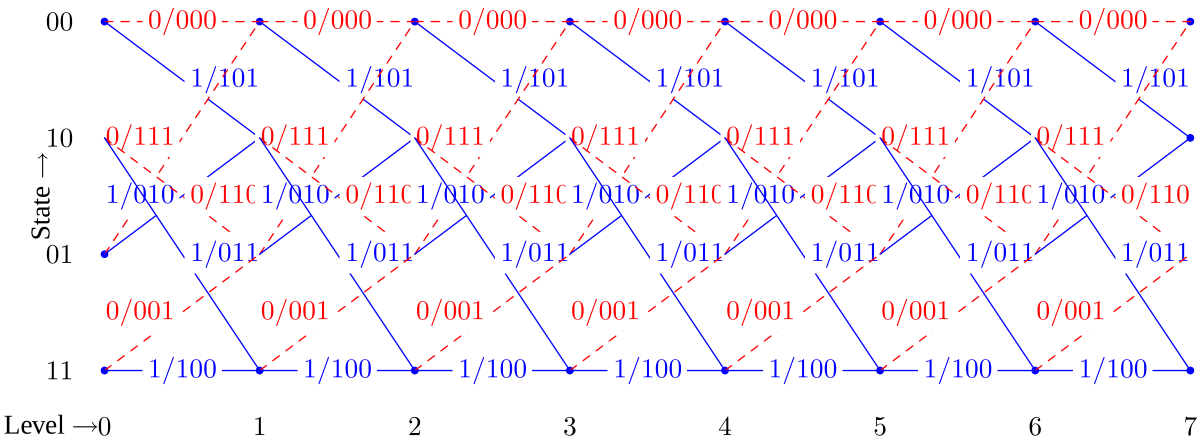

Figure 194: Trellis for decoding a constraint length 3 rate 1/3 convolutional code

quired for the modulation format of interest with the same data rate; the additional points allow redundancy for forward error-control coding without sacrificing bandwidth.

2. Convolutional coding is used to introduce a dependency between successive signal points such that only certain patterns or sequences of signal points are permitted.

3. Soft-decision decoding is performed in the receiver in which the permitted sequence of signals is modelled as a trellis structure.

By increasing the size of the constellation the probability of symbol error increases for a fixed signal-to-noise ratio and so with hard decision decoding we would gain nothing. Soft-decision decoding on the combined code and modulation ameliorates this problem. In soft decision decoding we maximise the Euclidean signal distance between code vectors as compared to maximising the hamming vectors between code words.

The design of this type of code involves partitioning an M-ary constellation successively into 2,4,8 subsets with size $M/2, M/4, M/8$ having progressively larger increasing Euclidean distance between their respective signal points. Shown here is the partitioning of an 8-PSK constellation into successive subsets having increase within subset Euclidean distances $d_0 < d_1 < d_2$. Similar partitioning can be done for QAM constellations.

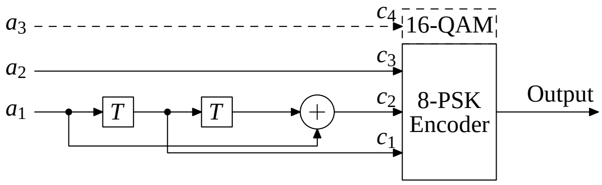

Figure 195: An Ungerboeck encoder

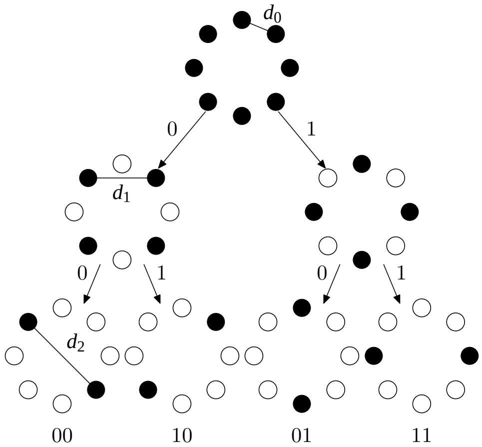

Figure 196: Partitioning of an 8-PSK constellation

Based on the subsets from the partitioning of the two-dimensional constellation we can devise simple efficient coding schemes. To send n bits/symbol with quadrature modulation we start with a two-dimensional constellation of 2^{n+1} signal points in the appropriate format. These are partitioned into 4 or 8 subsets and 1 or 2 incoming bits per symbol enter a rate-1/2 or rate-2/3 binary convolutional encoder respectively. The resulting 2 or 3 coded bits per symbol determine which subset is selected and the remaining uncoded data bits determine which particular point from the selected subset is to be signalled. This kind of Trellis code is called a *Ungerboeck code.* The Viterbi algorithm can be used to perform maximum likelihood sequence detection at the receiver with each branch in the trellis corresponding to a subset rather than a single signal point. The first step is to determine the signal point within each subset nearest to the received signal point. This signal point and its metric (distance from the received signal) are used thereafter for the branch in question and the Viterbi algorithm may proceed as normal.

Trellis coded modulation is used in the current generation of Vfast (V34) modems.

Assessment
Deadline: 2009-12-12 23:59:00
No Questions: 1
Time Allowed: 3 min

Turbo Coding

Turbo codes, first presented at the International Conference on Communications in 1993, have become very important in recent times in that they have demonstrated that 9it is possible to achieve transmission to withing 0.5 dB of the Shannon Limit for transmission in a practical way. They involve a combination of parallel concatenated codes and iterative decoding.

Classic Serial Concatenated Codes Figure 197) shows the classical idea of serial concatenated codes which typically uses a soft-decision trellis code and Viterbi decoder as the inner code which produces a hard decision output and a algebraic type hard-decision code (such as Reed-Solomon Code) as the outer coder to "clean up" residual errors of the first decoder. An example use of this is a Consultative Committee for Space Data Systems (CCSDS) standard which used a maximum-free-distance (MFD) (2,1,6) convolutional code as the inner code and various Reed-Solomon codes as outer codes.

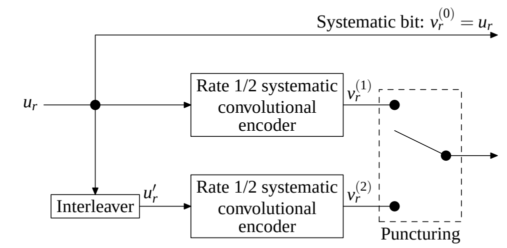

Figure 198: Parallel Concatenated Encoders

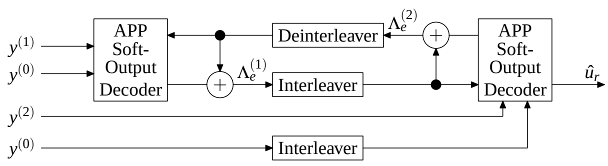

Figure 199: Iterative Decoding

Turbo Coding Turbo-Codes typically use parallel encoders and a iterative decoder as shown in figures 198) and 199).

The encoder illustrated uses two parallel systematic convolutional encoders. The fed into one of the encoders is interleaved compared with the input to the other. The outputs of the two encoders are transmitted in parallel with the systematic bits. The encoded output may be punctured (that is some of the bits may be removed - this technique is often used to increase the rate of trellis codes with little reduction in error correcting ability if the puncturing code is appropriately chosen).

The decoder makes use of two parallel "a posteriori probability" (APP) decoders. APP decoders use algorithms to compute the probabilities of transmission of either the information bits or encoded symbols $P\left[x_r|\vec{y}\right]$ where $\vec{y}$ is the received sequence at the output of the channel whose input is the transmitted sequence $\vec{x}$. Normally APP performs

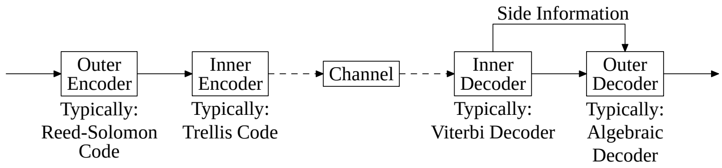

Figure 197: Classic Serial Concatenated Codes

only slightly better than the Viterbi algorithm for decoding however for Turbo decoding APP is used for soft-in soft-out (SISO) algorithms (i.e. both the inputs and outputs have probability information relation to the symbols of a Trellis code rather than hard decisions). The two SISO decoders are used to iteratively decode the data - that is the output of one is fed into the other which is then fed back into the first one (with appropriate interleaving and deinterleaving). After a small number of iterations the output is taken.

It is found that with this type of iterative decoding the error performance can converges quickly to close to the Shannon limit. To achieve a similar performance with Viterbi decoding would require very large convolutional codes and the consequent decoding complexity. The Trellis codes used in Turbo-coding are much shorter and therefore the complexity is much reduced. Note however that iterative decoding does add additional latency which may make it unsuitable in certain applications. Turbo-coding is used in Deep-space communication links and in recent DVB and 3G Wireless Standards.

See [TTC2004] for in depth analysis of Trellis and Turbo codes..

Efficiency and Complexity of Convolutional and Turbo Decoding

Figure 200) shows a comparison of various modulation schemes and the Shannon limit. For and efficiency of 2 bits/Hz Shannon's theory allows zero error if SNR > 1.5 (1.76dB). QPSK error rate improves only very gradually with SNR, Trellis coded modulation (TCM) provides significant further improvements but only at the cost of very large decoding complexity. The Turbo-coded regime however achieves further improvements using only two small concatenated trellis codes.

Table 10: Coding Gains for Trellis-coded 16-PSK modulation

Number of States	k_1	**Code Rate** $\frac{k_1}{k_1+1}$	$m = 3$ **coding gain (dB) of 16-PSK versus uncoded 8-PSK**	$m \to \infty$ N_{fed}
4	1	1/2	3.54	4
8	1	1/2	4.01	4
16	1	1/2	4.44	8
32	1	1/2	5.13	8
64	1	1/2	5.33	2
128	1	1/2	5.33	2
256	2	2/3	5.51	8

Digital Subscriber Lines (xDSL)

Many of us are familiar with using standard voice-band telecom modems (e.g. 56K, 38.8K etc) to connect to the internet. With these modems we connect end to end over the standard voice telephony system to a receiver system - usually a modem run by your service provider of choice. The transmission rate is limited to about 48Kbps by

[turbo-coding::ttc2004] "Trellis and Turbo Coding", Christian B. Schlegel and Lance C. PÃl'rez, IEE Press 2004. ISBN 0-471-22755-2.

[convolutional-coding-efficiency::ungerboeck87] "Channel coding with multilevel/phase signals", G. Ungerboeck, *IEEE Transactions on Information THeory*, IT-28, pp. 55-67, 1982

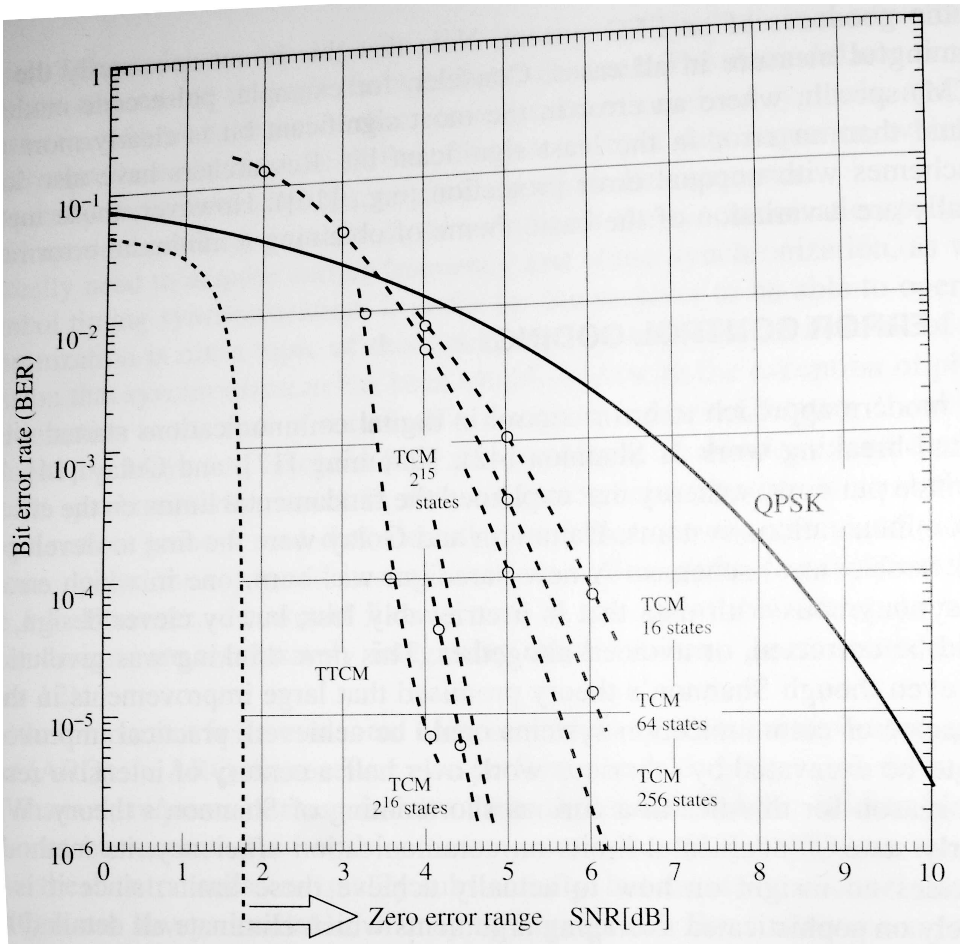

Figure 200: Bit error probability of quadrature phase-shift keying (QPSK) and selected 8-PSK trellis coded modulation (TCM) and trellis-turbo-coded modulation (TTCM) systems as a function of the normalised signal-to-noise ratio.

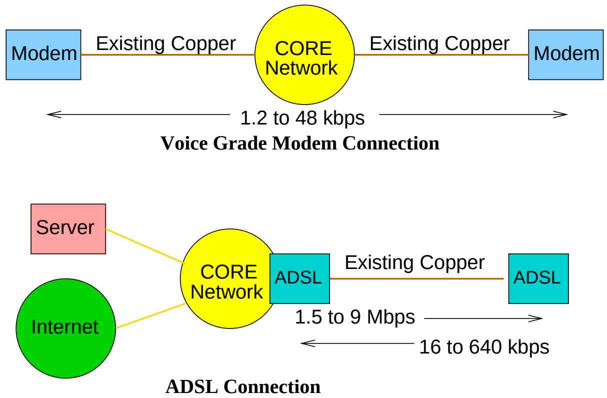

Figure 201: Comparison between Voiceband and xDSL modems

Table 11: Coding Gains for Trellis-coded QAM modulation

Number of States	k_1	**Code Rate** $\frac{k_1}{k_1+1}$	$m = 3$ **coding gain (dB) of 16-QAM versus uncoded 8-QAM**	$m = 4$ **coding gain (dB) of 32-QAM versus uncoded 16-QAM**	$m = 5$ **coding gain (dB) of 64-QAM versus uncoded 32-QAM**	**asymptotic coding gain (dB)**	N_{fed}
4	1	1/2	3.01	3.01	2.80	3.01	4
8	2	2/3	3.98	3.98	3.77	3.98	16
16	2	2/3	4.77	4.77	4.56	4.77	56
32	2	2/3	4.77	4.77	4.56	4.77	16
64	2	2/3	5.44	5.44	4.23	5.44	56
128	2	2/3	6.02	6.02	5.81	6.02	344
256	2	2/3	6.02	6.02	5.81	6.02	44

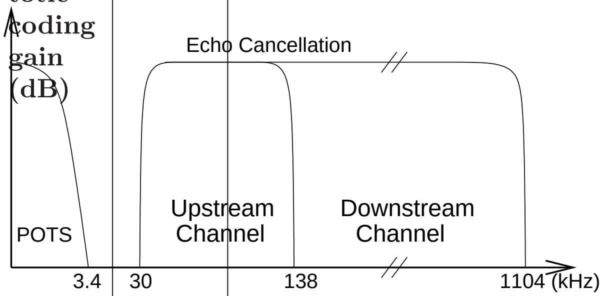

Figure 202: Spectrum utilisation for ADSL

the 4kHz bandwidth available over a standard voice band connection.

Clearly there is a need to support higher data rates to the home . xDSL are a family of technologies for sending data over the coppr wire. With Digital Subscriber Line (DSL) modems the signal does not pass into the telephone switching system. You are communicating over the copper wire in you local loop (the so called "last mile") to a modem at your local exchange (the DSLAM) which is connected to the data network. Consequently DSL modems are not limited to use just voiceband but can use much higher frequencies. The typical spectrum use is as shown in figure 202).

The range of DSL technologies which are summarised inthe table below. This is not an exhaustive table but shows those technologies which

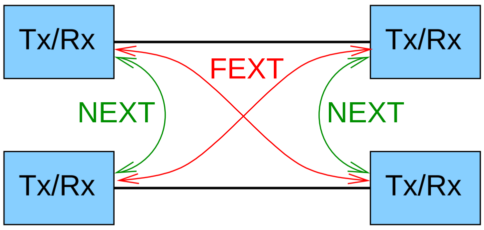

Figure 203: Near end and far end crosstalk (NEXT and FEXT)

seem to be gaining ground. Assymetric DSL (ADSL) is the main technology being used to roll-out broadband for home use. ITU-T Recommendation G.995.1 provides a comprehensive overview of these standardized recommendations.

Impairments in xDSL Systems

Typically a DSL modem will use high frequencies and as consequence the voiceband frequencies an be used for telephony as normal. Splitters are used at either end to seperate the voice from the data signals. The DSL signals uses freuncies typically at several hundered kHz. Sending such signals over the twisted pair copper wires and sharing them with voice signals is of course not without its problems. THe properites of the copper wires are quite variable in that the loop lengths vary, they are typically carried in differential mode, may be transformer coupled (no DC path) and may have bridge taps and loading coils (a variable impedance). One of the most significant issues is the crosstalk between twisted pairs (FEXT and NEXT).

NEXT is near end crosstalk, that is crosstalk with signals from the same end while FEXT is crosstalk from signals comming from the far end. In general, NEXT is much larger than FEXT because the interference source is closer to the receiver. With DSL NEXT is the more severe, and increases with frequencies and must be avoided. NEXT can in principle be eliminated by not transmitting in both directions in the same time, separating the two directions of transmission into either nonoverlapping intervals in time or nonoverlapping frequency bands. This is how xDSL systems attempt to avoid self-NEXT by using frequency or time-division duplexing. Finally Radio-frequency

Table 12: A summary of different DSL technologies

Technology	**Speed**	**Distance Limitation**	**Application**
56 Kbps	56 Kbps downstream, 33.6 Kbps upstream	None	Email, remote LAN access, Internet/intranet access
ISDN	Up to 128 kbps Full duplex	18,000 feet (additional equipment can extend the distance)	Video Conferencing, disaster recovery, leased line backup, transaction processing, call centre services, internet/intranet access
Cable Modem	10-30 Mbps downstream, 128 Kbps-10 Mbps upstream (shared)	30 miles over coaxial (additional equipment can extend the distance to 200 miles)	Internet Access
ADSL Lite	Up to 1 Mbps downstreamn Up to 512 Kbps upstream	18,000 feet	Internet/intranet access, IP telephony, video telephony
ADSL/R-ADSL	1.5-8 Mbps dowstream, up to 1.544 Mbps upstream	18,000 feet (12,000 feet for fastest speeds)	Internet/intranet access, video on demand, remote LAN access, VPNs, VoIP
HDSL	1.544 Mbps full duplex (T1) 2.048 Mbps full Duplex (E1) (uses 2-3 wire pairs)	12,000-15,000 feet	Local, repeatered T1/E1 trunk replacement, PBX interconnection, Frame Relay Traffic aggregator, LAN interconnect
SDSL	1.544 Mbps full duplex	10,000 feet	Local, repeatered

interference (ingress and egress) are significant issues. In particular the spectrum used by DSL is also shared by several amateur and industrial radio frequency bands. xDSL modems must therefore use advanced modulation tecnhnques to overcome these problems.

Unavoidable Impairments

- Alien Crosstalk
- kindred FEXT
- AWGN

Avoidable Impairments

- Kindred NEXT
- RFI (AM radio and amateur radio)
- POTS signalling
- Linear distortion
- Non-linear distortion
- Down/up interference leak through FDD filters and/or echo canceller
- Clipping
- Quantising noise in DAC and ADC
- DSP round-off noise
- Noise and/or distortion introduced by clock jitter

ADSL modulation formats

As mentioned in Digital Subscriber Lines (xDSL) (Section) advanced modulation formats are needed to enable high data rate transmission over the copper local loop. Two technologies used in ADSL are CAP and DMT. Both these technologies require digital signal processing.

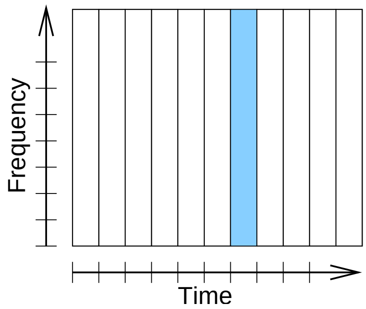

Figure 204: CAP use time division multiplexing

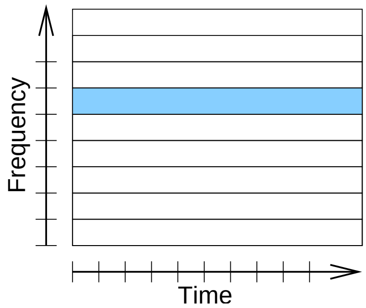

Figure 205: DMT use frequency division multiplexing

Carrierless AM/PM (CAP) Modulation This was the first modulation technology used to deliver ADSL. It is essentially suppressed carrier QAM. The othogonal signal modulation is executed digitally. Essentially the two channels are shaped using seperate digital transversal bandpass filters with equal amplitude characteristics and a $pi/2$ difference in phase (a Hilbert pair). The signals are then combined, fed to a digital to analogue converter (DAC) and then transmitted. The spectrum may be seperated into upstream and downstream components allowing duplex operation. This approach has now lost favour for DMT modulation.

Discrete Multitine (DMT) Modulation This is closely related to orthogonal frequency division multiplexing. The frequency band is split into many narrowband carriers, all transmitted in parallel and carrying a fraction of the total information. A tyical DMT transmitter will have 256 frequency bands (or channels) of 4.3125kHz each. In a typical assymetric configuration channels 6-31 (24kHz-136kHz) are used for upstream and channels 32-350 (136KHz-1.1MHz are used for downstreams. Other channels are needed for synchronisation putposes. The synthesis of this signal is carried out using digital signal processing and the inverse discrete fourier transform. In principle other transforms such as wavelet may perform better than FFT transforms - this is still in the research stage. Each of the freuquency channels will be modulated using QAM, typical constellations are shown in figure 206).

Adaptive Spectrum A major advantage of DMT is that the spectrum can be adaptively modified depending on conditions and the actual transmission channel. If there is interference in one part of the spectrum, then that part can be avoided.

Typically the modems will continuously monitor the signal to noise ratio on each channel and can swap bits assigned to each channel to maintain equal BER across all channels i.e. bit (s) may be swapped from the worst margin subchannel to the best margin channel(s). A synchronisation protocol is needed for the execution of bit swapping between two modems.

An ADSL Transmitter

Clearly an ADSL transmitter has a lot of processing to do. Some of the stages are listed below.

Transport of the Network Timing Reference Since the data is now send directly onto the network, it must be synchronised with the network clock

Input multiplexer and Latency (Interleave) Path Assignment The process of allocating bits to different channels

Scrambler Scrambling reduces the correlation between signals and aids elimintation of interference from echo signals

Reed-Solomon Forward Error Correction Hard decision error correction used to protect the data

Interleaving Jointly with the Reed-Solomon Error correction this greatly improves the error rate. It is similar to the error correction approach used on CDs

Tone Ordering

Trellis Code Modulation Soft decision decoding is used - this enables determination of the SNR on each channel which in turn feeds into the bit allocations

Pilot Tone To synchronise carrier frequencies

Inverse Discrete Fourier Transform To determine the actual transmitted waveform

PAR Reduction Power fluctuation, parameterised by the Peak-to-Average power is a significant issue with these techniques and so techniques are needed to try and maintain a nearly constant power.

Digital-to-Analogue Converter To generate the actual analogue signal

Line Drivers To match impedance with the transmission line

Coded Orthogonal Frequency Division Multiplexing (COFDM)

Coded Orthogonal Frequency Division Multiplexing (COFDM) is essentially another name for adsl-modulation-formats::Discrete Multitine (DMT) Modulation (Section) mentioned in the context of ADSL. It is used in DVB-T (terrestrial digital television broadcasting) and, in a slightly different form in DAB (digital audio broadcast). These broadcast media have similar interference issues to DSL technologies and additionally have to cope with multiple path fading (which can be time varying in the case of mobile reception). The use of an adaptive spectrum, as COFDM allows, enables the system to work around these problems. As with DTM COFDM uses soft decision decoding taking account of the channel state information (measured SNR in each channel). A guard interval is used between symbols to reduce multiple path problems.

Two good articles on COFDM have been written by members of the BBC research and development organisation

1. The how and why of COFDM
2. Explaining some of the magic of COFDM

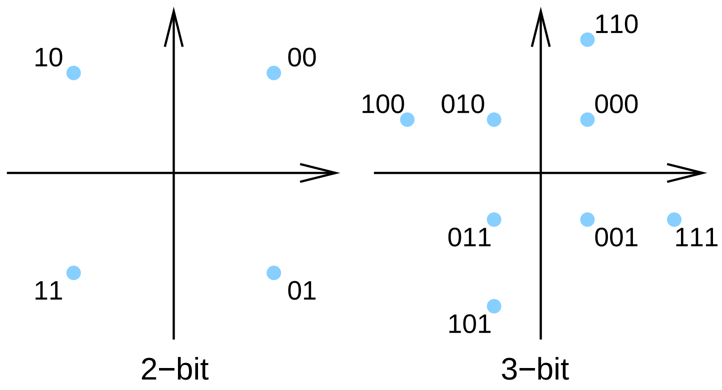

Figure 206: Typical subchannel signal constellations

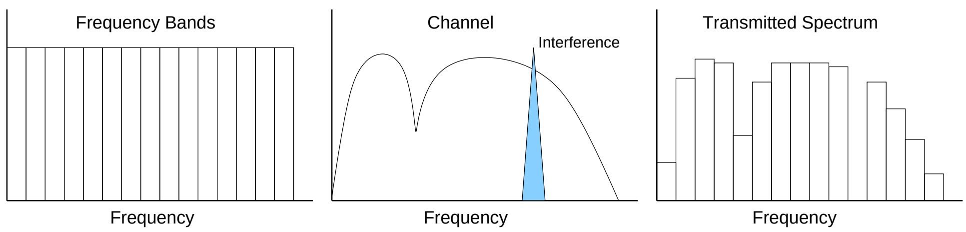

Figure 207: Adaptive Spectrum

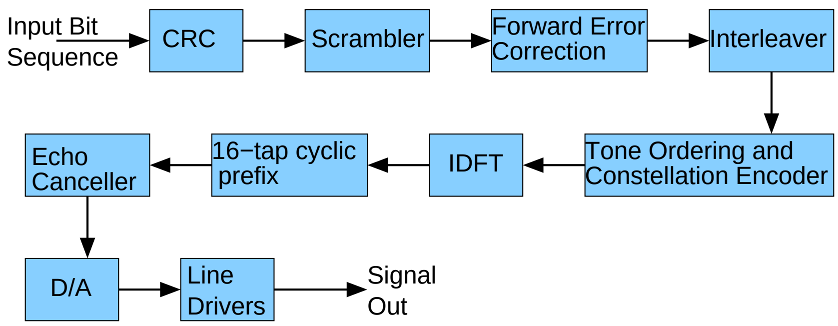

Figure 208: An ADSL Transmitter

Generated on : Tue, 07 Jul 2009 09:59:31 . Generated by cl-docutils 0.1.1 from restructured text source.