School of Electrical and Electronics

Department of Electronics and Communication

SECA1403- Digital Communication

UNIT I – SAMPLING AND QUANTIZATION - SECA1403

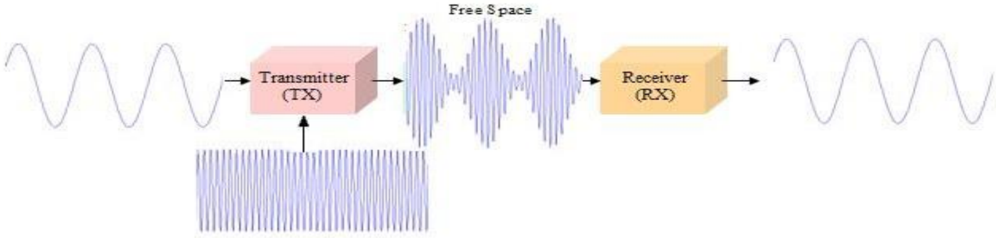
SAMPLING AND QUANTIZATION

Review of Sampling process -Natural Sampling-Flat Sampling - Aliasing - Signal Reconstruction-Quantization - Uniform & non-uniform quantization - quantization noise Bandwidth -Noise trade off-PCM- Noise considerations in PCM - differential pulse code modulation - Delta modulation -Linear prediction –Adaptive Delta Modulation.

SAMPLING AND QUANTIZATION

Introduction:

In Communication systems we studied the term Modulation, which is helpful in transmitting the signals at higher frequencies rather than at baseband frequencies. The following are the outputs of Amplitude modulation (AM), Frequency modulation (FM) and Phase modulation (PM),



Amplitude Modulation (AM)

Fig.1 Types of Modulation

Analog systems use electrical signals to represent these natural patterns. The electrical signals are referred to as analog signals.

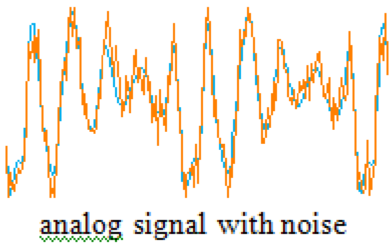
Fig. 1.1 Analog Signal

Digital systems use electrical signals that represent discrete, binary values. The electrical signals are referred to as **digital signals**. Specifically, binary digital signals use two discrete voltage levels to represent binary 1 or 0. Combining multiple bits into words permits us to represent larger values. Digital circuits operate on digital signals performing logic and arithmetic functions like we saw with our digital logic gates.

Fig. 1.2 Digital Signal

Digital advantages

1. <u>Noise immunity</u>: Noise immunity is the **most important advantage** of digital communications. Receiver circuitry can distinguish between a binary 0 and 1 with a significant amount of noise.



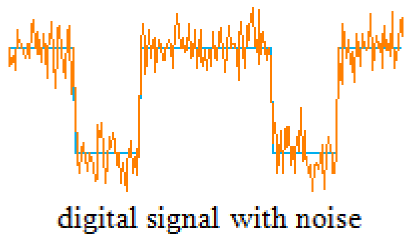


Fig. 1.3 Noise Signals

2. <u>Error detection/corrections:</u> Even if a digital signal does contain bit errors, many of these errors can be fixed at the receiver through the use of *error correcting codes*. Error correcting codes allows CDs with minor scratches to be played without errors.

3. <u>Easier multiplexing:</u> Multiplexing is the process of allowing multiple signals to share the same transmission channel, which we will talk about in our next lesson.

4. <u>Easier to process and store:</u> Since digital is the how computers work, digital signals can be easily processed (aka digital signal processing (DSP)) by computers. Similarly the digital format for easier storage of communication signals.

There are some disadvantages to digital communication; the most important is that the bandwidth size required is larger by almost twice that of analog communication. The digital circuitry is also more complex, but generally smaller and less expensive.

Conversion from analog to digital

Before we can use digital transmission, we must convert the signal of interest into a digital format. The natural pattern (i.e. speech) that we want to transmit will be acquired using an analog device. The analog signal will be translated into a digital signal using a method called **analog-to-digital (A/D) conversion**, digitizing a signal, or encoding. The device used to perform this translation is known as an analog-to-digital converter or ADC. Through A/D conversion continuously variable analog signals are changed into a series of binary numbers.

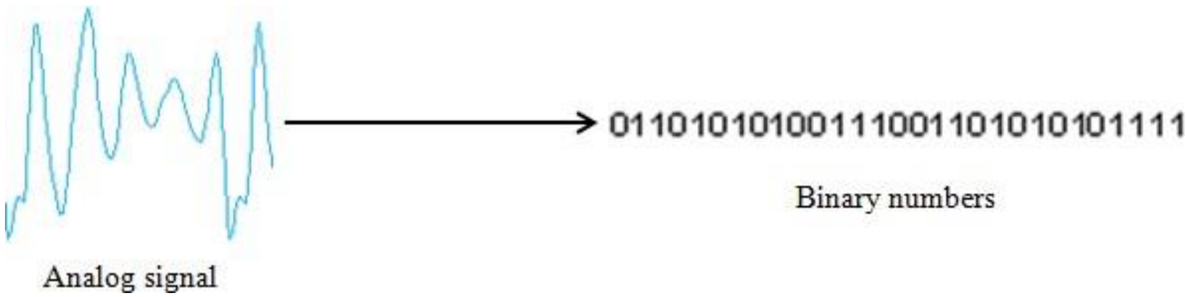


Fig. 1.4 A/D conversion

There are two steps we will use to convert an analog signal to a digital format as represented by binary numbers.

1. Sampling
2. Quantization

Sampling Process:

Sampling Theorem

Consider a band-limited signal with no frequency components above a certain frequency f_m. The sampling theorem states that this signal can be recovered completely from a set of samples of its amplitude, if the samples are taken at the rate of $\mathbf{f_s > 2f_m}$ samples per second.This is often called the uniform sampling theorem for baseband or low-pass signals.

The minimum sampling rate, $2f_m$ samples per second, is called the Nyquist sampling rate (or Nyquist frequency); its reciprocal $l/(2f_m)$ (measured in seconds) is called the Nyquist interval.

$f_s = 2 * f_m$ is called the Nyquist sampling rate.

For telephone speech the standard sampling rate is 8 kHz (or one sample every 125 □s).

Sampling Methods

Suppose we have an arbitrary signal **(the baseband signal m(t)) which has a spectrum M(f).** Take infinitesimally short samples of the signal m(t) at a uniform rate once every t_s seconds i.e. at a frequency f_s. This is the ideal form of sampling, it is called instantaneous (or impulse) sampling.

In effect the signal m(t) is multiplied by a train of impulses giving rise to a train of pulses as in the lower line of the diagram. The train of sampling impulses has a frequency spectrum consisting of all harmonics or multiples of f_s and all are at the same amplitude.

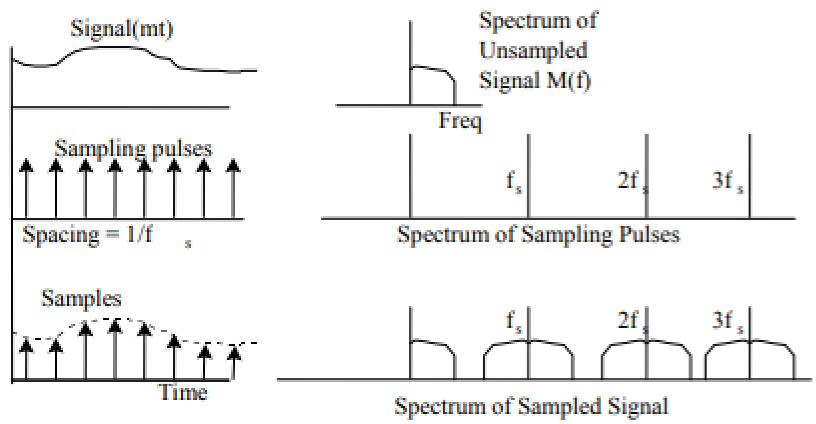


Fig. 1.5 Sampling

This **sampled signal has a spectrum as shown where M(f) is repeated unattenuated periodically and appears around all multiples of the sampling frequency ($f_s = 1/t_s$)**.

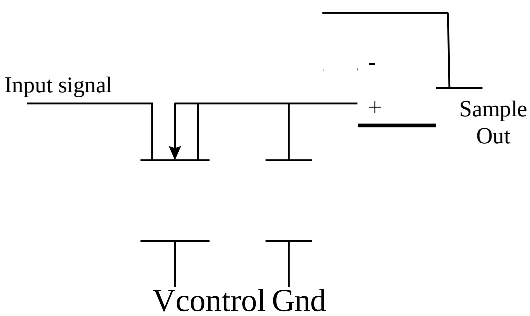
To recover m(t) from the sampled signal we need only pass the sampled signal through a low pass filter with a stop frequency of $f_s/2$. All of the higher frequency components will be dropped. In the diagram, if f_s is greater than twice the highest frequency in m(t) the repetitions of the sampled spectra around the harmonics of the f_s do not overlap.

Flat - top Sampling

An Analogue to Digital Converter requires that the sample value be held constant for a fixed time until the conversion is completed. This requires a flat-top sampled signal. This has approximately the same repeated frequency spectrum as with the instantaneous sampling above, but with each repetition slightly spread out.

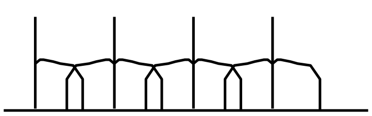
The simplest and most common sampling method is performed by a functional block termed a Sample and Hold (S/H) circuit.

Fig 1.6 Flat top sampling



The output from the circuit must be held at a constant level for the sampling duration. Vcontrol switches the MOSFET ON until the charge on C is equal to the amplitude of the sampled voltage. Vcontrol then goes LOW, the MOSFET is OFF and the charge is held by the capacitor. The charge held on the capacitor puts a voltage across the capacitor, and it is held at that value until the next time that Vcontrol switches the MOSFET ON. This is called a sample and hold circuit and is usually used as the input to an ADC.

Aliasing Error



Spectrum of Sampled Signal

If a signal is under sampled (sampled at a rate below the Nyquist rate), the spectrum consists of overlapping repetitions of the sampled spectrum. Because of the overlapping tails a single repetition of the spectrum no longer has the complete information about the unsampled signal, and it is no longer possible to recover it from the sampled signal. To recover the original signal at the receiving end the sampled signal is passed through a lowpass filter with a cut off of $f_s/2$, we get a spectrum that is not the sampled signal but is a different version due to:

- Loss of the tail of the sampled signal spectrum beyond $f_s/2$
- This same tail appears inverted, or folded, onto the spectrum at the cut-off frequency. This tail inversion is known as aliasing, (or spectral folding or fold over distortion).

The aliasing distortion can be eliminated by cutting the tail (i.e. filtering) of the sampled signal beyond $f > f_s/2$ before the signal is sampled. By so doing, the overlap of successive cycles in the sampled signal is avoided. The only error in the recovery of the unsampled signal is that caused by the missing tail above $f_s/2$.

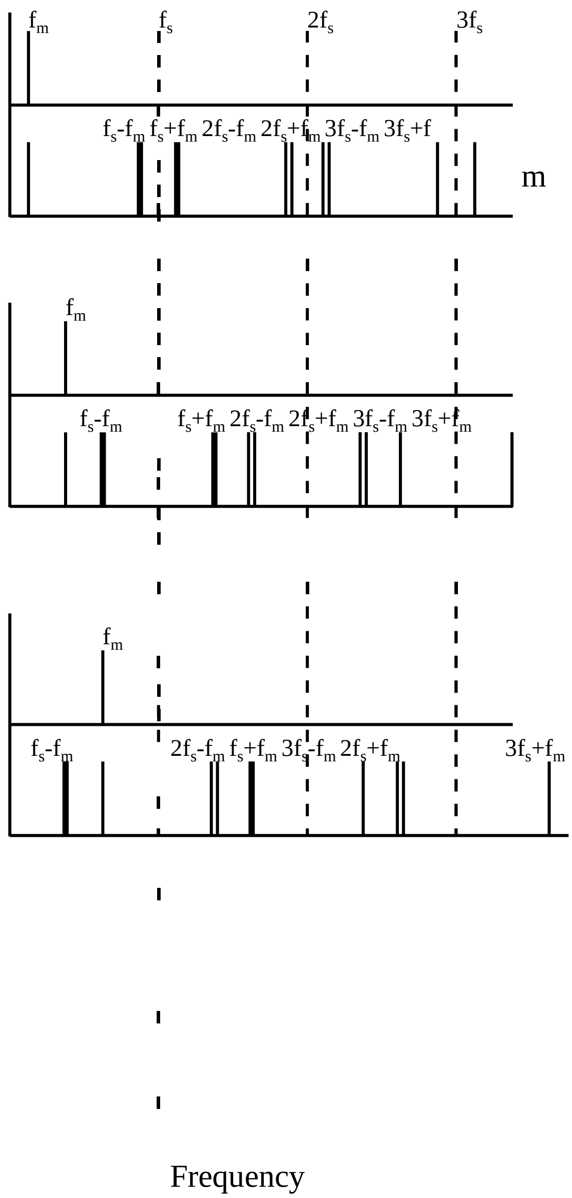


Fig. 1.7 Aliasing

It is simpler to consider aliasing by considering a single frequency component of m(t). We will look at the frequency f_m and it is sampled at a rate f_s. The diagrams show the frequencies which will be present in the sampled signal. There will be frequency components at f_m, $f_s - f_m$, $f_s + f_m$, $2 f_s - f_m$, $2 f_s + f_m$, $3 f_s - f_m$, $3 f_s + f_m$, etc. etc.

In the first case f_m is very much less than f_s, so that $f_s - f_m$ is much higher than the cut off of the filter ($f_s/2$).

In the second case f_m is below, but close to $f_s/2$, so that a sharp cut off filter is required to ensure that f_m is passed but $f_s - f_m$ is stopped.

In the third case f_m is higher than $f_s/2$, so that $f_s - f_m$ is less than $f_s/2$. The low pass filter with a cutoff of $f_s/2$ will therefore block f_m (the actual signal frequency) but will pass a signal with frequency $f_s - f_m$.

This is **aliasing**

Strictly speaking, a band limited signal does not exist in reality. It can be shown that if a signal is time limited it cannot be band limited. All physical signals are necessarily time limited because they begin at some finite instant and must terminate at some other finite instant. Hence, all practical signals are theoretically non band limited.

A real signal contains a finite amount of energy, therefore its frequency spectrum must decay at higher frequencies. Most of the signal energy resides in a finite band, and the spectrum at higher frequencies contributes little. The error introduced by cutting off the tail beyond a certain frequency B can be made negligible by making B sufficiently large.

Thus, for all practical purposes a signal can be considered to be essentially band limited at some value B, the choice of which depends upon the accuracy desired. A practical example of this is a speech signal. Theoretically, a speech signal, being a finite time signal, has an infinite bandwidth. But frequency components beyond 3400 Hz contribute a small fraction of the total energy. When speech signals are transmitted by PCM they are first passed through a lowpass filter of bandwidth of 3500 Hz. (This filter is called an **anti aliasing filter**). Higher sampling rates (i.e. 8000 samples/sec) permits recovery of the signal from its samples using relatively simple filters i.e. it allows for guard bands between the repetitions of the spectrum (otherwise recovering signals sampled at the Nyquist rate would require very sharp cut-off (ideal) filters).

In summary, aliasing distortion produces frequency components in the desired frequency band that did not exist in the original waveform. Aliasing problems are not confined to speech digitization processes. The potential for aliasing is present in any sample data system.

Motion picture taking, for example, is another sampling system that can produce aliasing. A common example occurs when filming a rotating wheel. Often the sampling process (the picture refresh rate) is too slow to keep up with the wheel movements and spurious rotational rates are produced. If the wheel rotates 355^0 between frames, it looks to the eye as if it has moved backwards 5^0.

Pulse Modulation of Signals:

In many cases, bandwidth of communication link is much greater than signal bandwidth. The signal can be transmitted using short pulses with low duty cycle:

- Pulse amplitude modulation: width fixed, amplitude varies
- Pulse width modulation: position fixed, width varies
- Pulse position modulation: width fixed, position varies

All three methods can be used with time-division multiplexing to carry multiple signals over a single channel

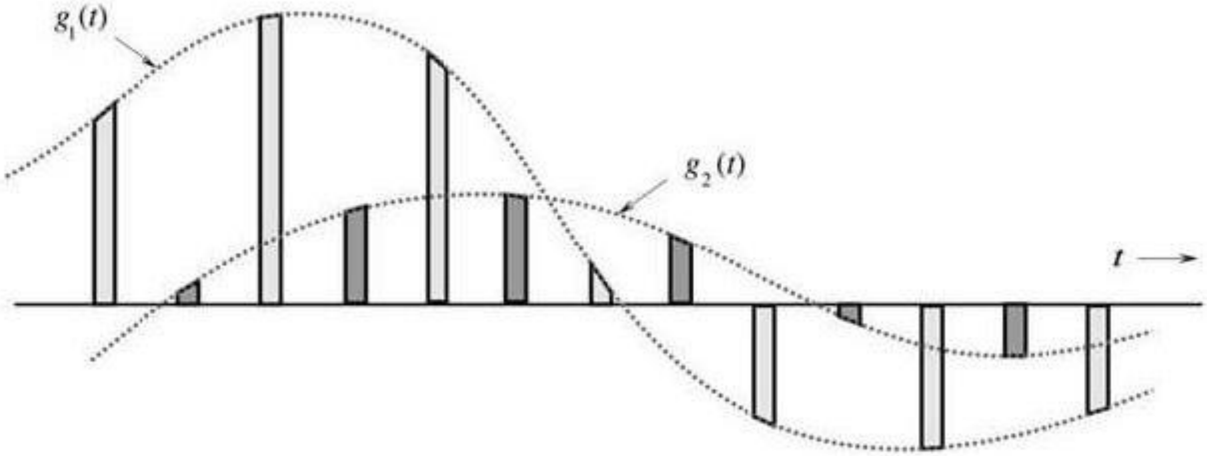
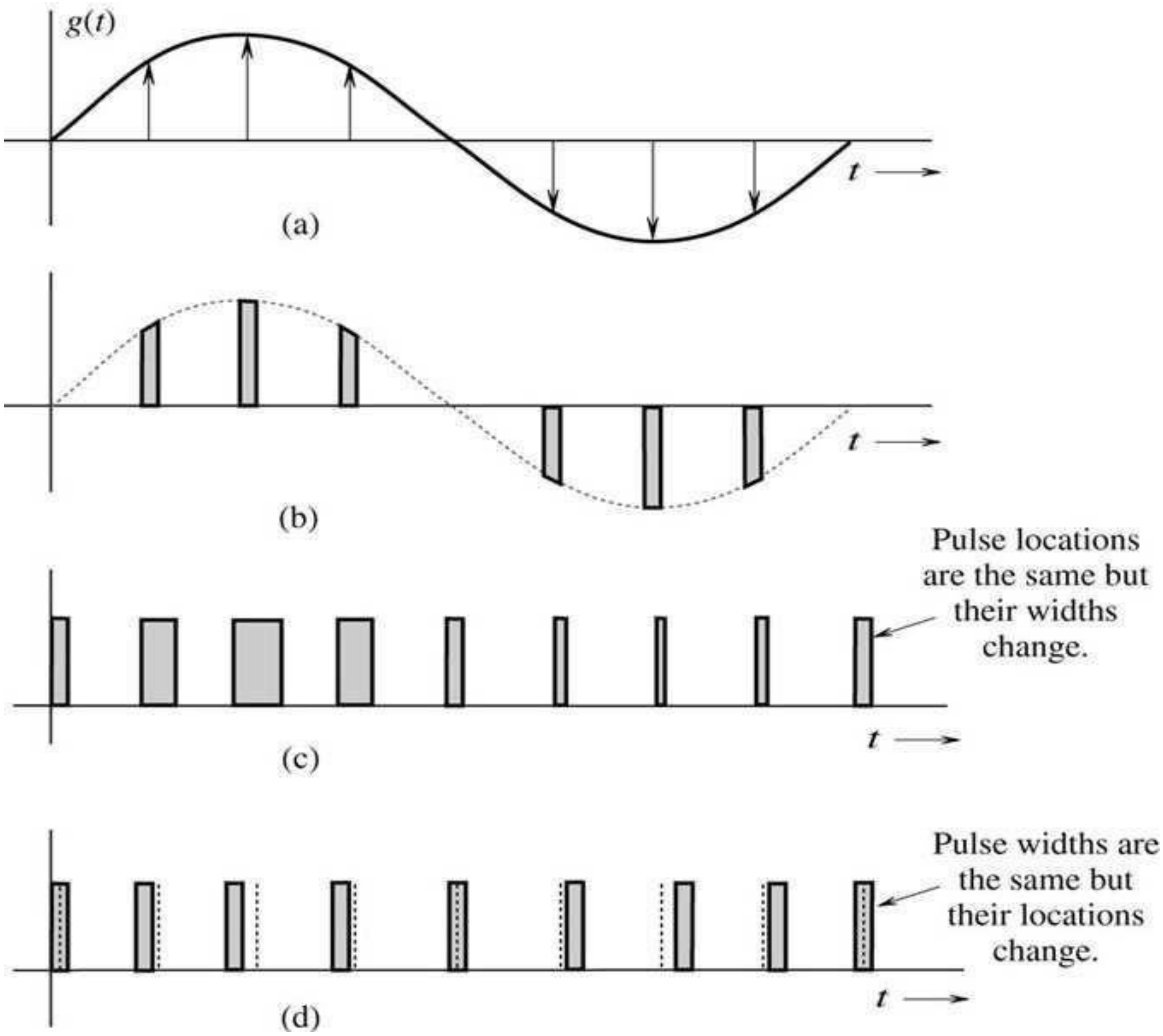


Fig. 1.9 PAM,PWM,PPM: Amplitude, Width and Position



Pulse Amplitude Modulation

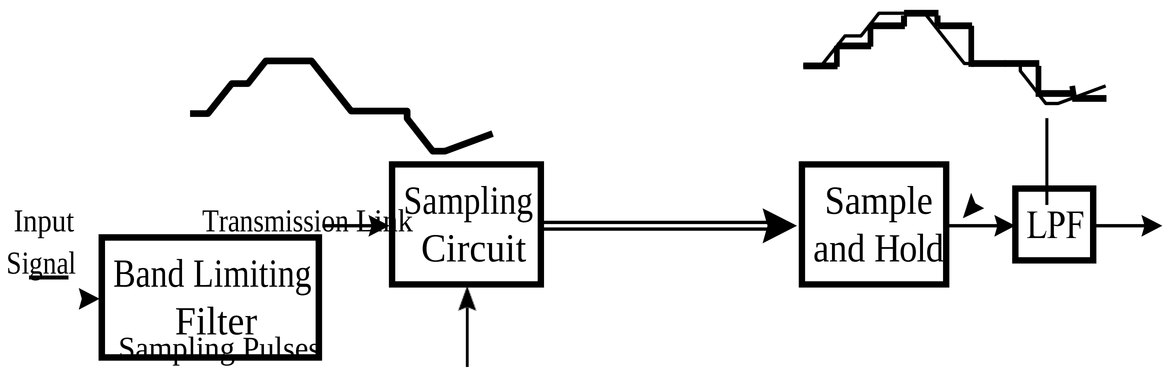


Fig. 1.10 PAM

A sampled signal consists of a train of pulses, where each pulse corresponds to the amplitude of the signal at the corresponding sampling time. The signal sent to line is modulated in amplitude and hence the name **Pulse Amplitude Modulation** (PAM).

A complete PAM system must include a band limiting (or **anti aliasing**) filter before sampling to ensure that no spurious or source-related signals get folded back into the desired signal bandwidth - no aliasing. The input filter may also be designed to cut off very low frequencies to removed 50 Hz hum from power lines.

Several PAM signals can be multiplexed together as long as they are kept distinct and are recoverable at the receiving end. This system is one example of Time Division Multiplex (TDM) transmission (although it has never been widely used fo speech, it has applications in remote monitoring and telemetry).

The sample-and-hold circuit takes in each pulse and holds the amplitude of that pulse until the arrival of the next pulse. It produces a staircase approximation to the sampled wave form. With use of the staircase approximation, the power level of the signal coming out of the reconstructive filter (LPF) is nearly the same as the level of the sampled input signal.

The filters are assumed to have ideal characteristics - not like real filters. Filters with real attenuation slopes at the band edge can be used if the input signal is slightly over sampled. If sampling frequency is greater than twice the bandwidth, the spectral bands are sufficiently separated from each other that filters with gradual roll-off characteristics can be used.

As an example, sampled voice systems typically use band limiting filters with a 3 dB cut- off around 3.4 kHz and a sampling rate of 8 kHz. Thus the sampled signal is sufficiently attenuated at of 4 kHz to adequately reduce the energy level of the foldover spectrum.

Pulse Width Modulation (PWM)

Pulse width modulation is also called pulse duration modulation (PDM).PWM is more often used for control than for communication. A signal can be recovered exactly from its PWM samples at rate 2B, provided the bandwidth is ≤ 0.637B.PWM output can be generated by a sawtooth signal gating the input. Below the pulse width varies from nearly 0 to 1/2 the pulse period.

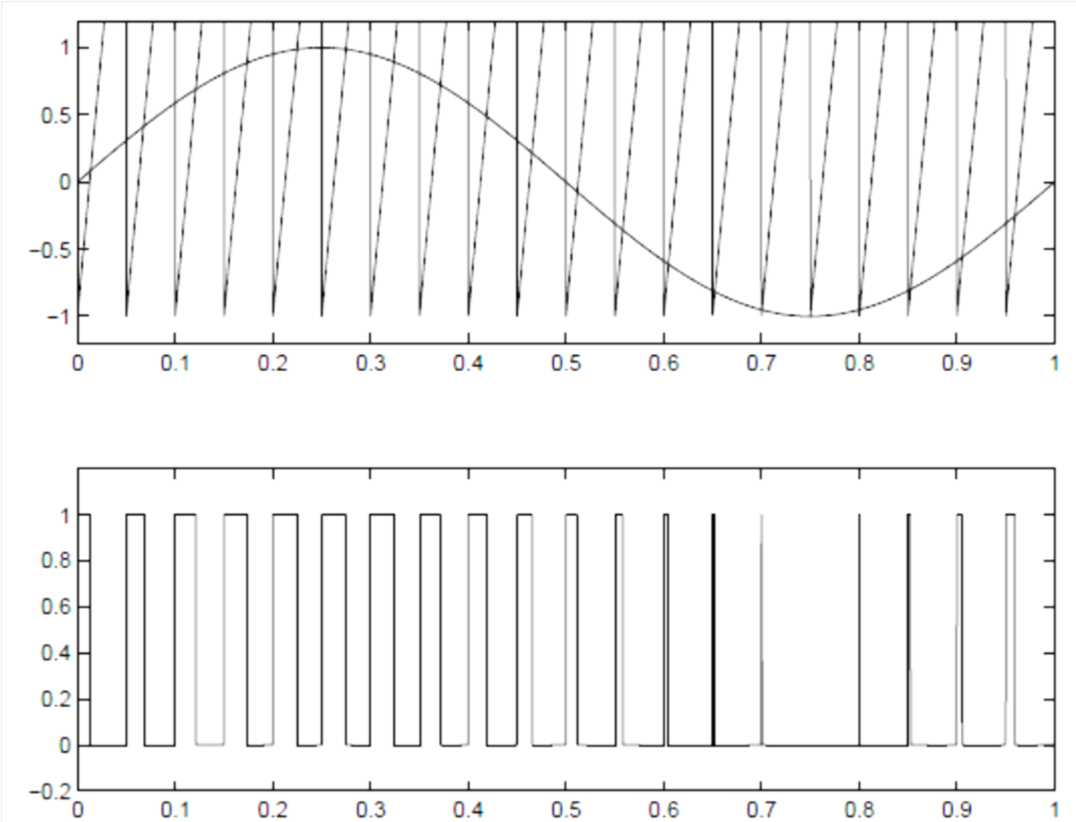


Fig.1.11 PWM

Pulse Position Modulation (PPM)

The value of the signal determines the delay of the pulse from the clock. Electrical circuits use timers such as 555. Microcontrollers can generate PPM (and PWM) in software.

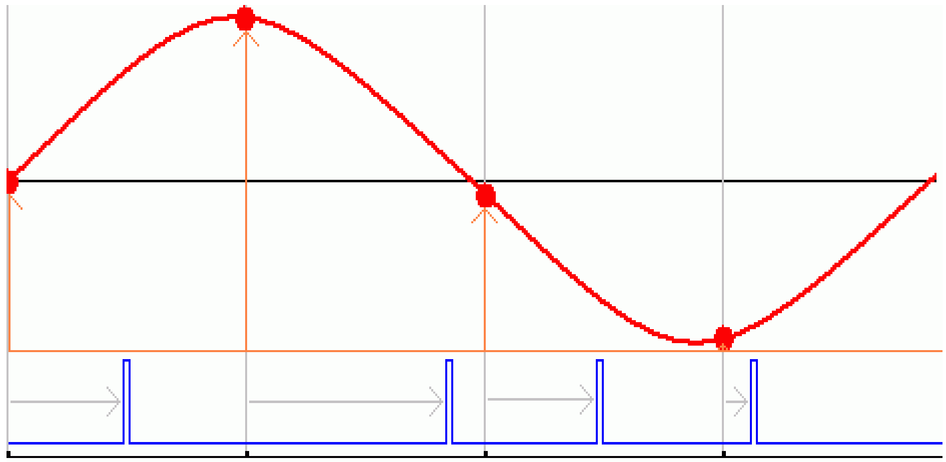


Fig. 1.12 PPM

Bandwidth Noise Tradeoff:

The noise analysis of PPM are as follows:

1. The figure of merit is proportional to square of the ratio [B_T/W].
2. As the signal to noise ratio is reduced, the systems exhibit threshold effect.
3. With digital pulse modulation, the better noise performance than square law can be obtained.
4. The digital pulse modulation such as PCM gives negligible noise effect by increasing the average power in binary PCM signal.
5. With PCM, the bandwidth noise tradeoff can be related by exponential law.

Pulse Code Modulation

Pulse Code Modulation (PCM) is an extension of PAM wherein each analogue sample value is quantized into a discrete value for representation as a digital code word.

Thus, as shown below, a PAM system can be converted into a PCM system by adding a suitable analogue-to-digital (A/D) converter at the source and a digital-to-analogue (D/A) converter at the destination.

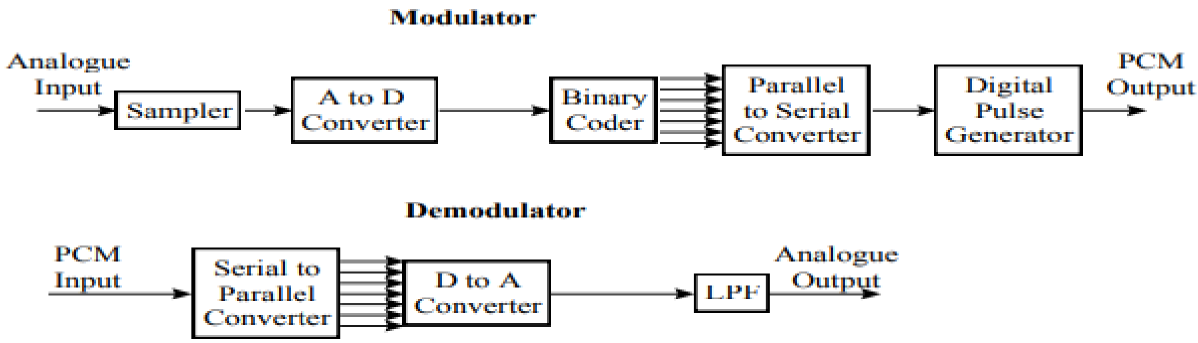


Fig 1.13 PCM

PCM is a true digital process as compared to PAM. In PCM the speech signal is converted from analogue to digital form.

PCM is standardized for telephony by the ITU-T (International Telecommunications Union - Telecoms, a branch of the UN), in a series of recommendations called the G series. For example the ITU-T recommendations for out-of-band signal rejection in PCM voice coders require that 14 dB of attenuation is provided at 4 kHz. Also, the ITU-T transmission quality specification for telephony terminals require that the frequency response of the handset microphone has a sharp roll-off from 3.4 kHz.

In quantization the levels are assigned a binary codeword. All sample values falling between two quantization levels are considered to be located at the centre of the quantization interval. In this manner the quantization process introduces a certain amount of error or distortion into the signal samples. This error known as quantization noise, is minimised by establishing a large number of small quantization intervals. Of course, as the number of quantization intervals increase, so must the number or bits increase to uniquely identify the quantization intervals. For example, if an analogue voltage level is to be converted to a digital system with 8 discrete levels or quantization steps three bits are required. In the ITU-T version there are 256 quantization steps, 128 positive and 128 negative, requiring 8 bits. A positive level is represented by having bit 8 (MSB) at 0, and for a negative level the MSB is 1.

Quantization

This is the process of setting the sample amplitude, which can be continuously variable to a discrete value. Look at Uniform Quantization first, where the discrete values are evenly spaced.

Uniform Quantization

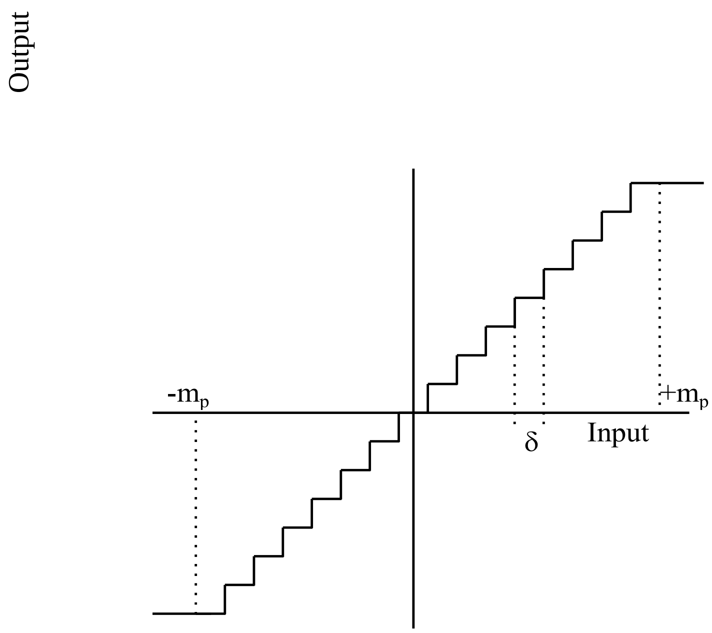


Fig.1.14 Quantization

We assume that the amplitude of the signal m(t) is confined to the range (-m_p, +m_p). This range ($2m_p$) is divided into L levels, each of step size δ, given by

$$\delta = 2\, m_p / L$$

A sample amplitude value is approximated by the midpoint of the interval in which it lies. The input/output characteristics of a uniform quantizer are shown.

Nonuniform quantizers

They have unequally spaced levels.The spacing can be chosen to optimize the Signal- to-Noise Ratio for a particular type of signal.It is characterized by:

Variable step size and Quantizer size depend on signal size

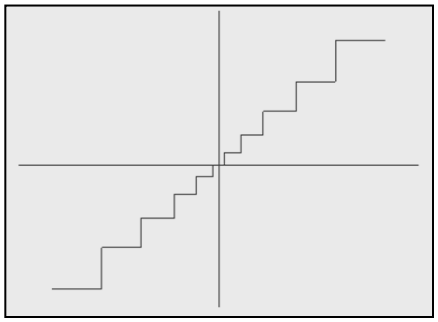


Fig. 1.15 Non uniform quantizer

Companding

In a uniform or linear PCM system the size of every quantization interval is determined by the SQR requirement of the lowest signal to be encoded. This interval is also for the largest signal - which therefore has a much better SQR.

> **Example:** A 26 dB SQR for small signals and a 30 dB dynamic range produces a 56 dB SQR for the maximum amplitude signal.

In this way a uniform PCM system provides unneeded quality for large signals. In speech the max amplitude signals are the least likely to occur. The code space in a uniform PCM system is very inefficiently utilised.

A more efficient coding is achieved if the quantization intervals increase with the sample value. When the quantization interval is directly proportional to the sample value (assign small quantization intervals to small signals and large intervals to large signals) the SQR is constant for all signal levels. With this technique fewer bits per sample are required to provide a specified SQR for small signals and an adequate dynamic range for large signals (but still with the SQR as for the small signals). The quantization intervals are not constant and there will be a non linear relationship between the code words and the values they represent.

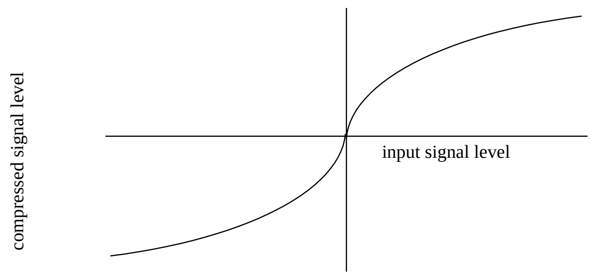


Fig. 1.16 Companding

Originally to produce the non linear quantization the baseband signal was passed through a non-linear amplifier with input/output characteristics as shown before the samples were taken. Low level signals were amplified and high level signals were attenuated. The larger the sample value the more it is **comp**ressed before encoding. The PCM decoder ex**pands** the compressed value using an inverse compression characteristic to recover the original sample value. The two processes are called **companding**.

There are 2 companding schemes to describe the curve above:

1. **μ-Law Companding** (also called log-PCM)
   This is used in North America and Japan. It uses a logarithmic compression curve which is ideal in the sense that quantization intervals and hence quantization noise is directly proportional to signal level (and so a constant SQR).

2. **A- Law Companding**
   This is the ITU-T standard. It is used in Europe and most of the rest of the world. It is very similar to the μ-Law coding. It is represented by straight line segments to facilitate digital companding. Originally the non linear function was obtained using non linear devices such as special diodes. These days in a PCM system the A to D and D to A converters (ADC and DAC) include a companding function.

Differential pulse coding schemes

PCM transmits the absolute value of the signal for each frame. Instead we can transmit information about the difference between each sample. The two main differential coding schemes are:

- Delta Modulation
- Differential PCM and Adaptive Differential PCM (ADPCM)

Delta Modulation

Delta modulation converts an analogue signal, normally voice, into a digital signal.

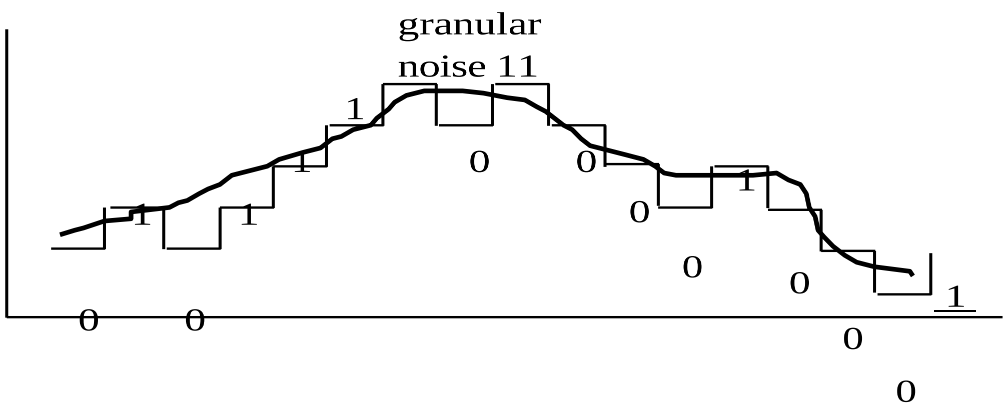


Fig. 1.17 Delta Modulaion

The analogue signal is sampled as in the PCM process. Then the sample is compared with the previous sample. The result of the comparison is quantified using a one bit coder. If the sample is greater than the previous sample a 1 is generated. Otherwise a 0 is generated. The advantage of delta modulation over PCM is its simplicity and lower cost. But the noise performance is not as good as PCM.

To reconstruct the original from the quantization, if a 1 is received the signal is increased by a step of size q, if a 0 is received the output is reduced by the same size step. Slope overload occurs when the encoded waveform is more than a step size away from the input signal. This condition happens when the rate of change of the input exceeds the maximum change that can be generated by the output.

Overload will occur if:

$$\max\left|\frac{d}{dt}x(t)\right| > \frac{\delta}{T_s}$$

$$\max\left|\frac{d}{dt}A_m \sin(2\pi f_m t)\right| > \frac{\delta}{T_s}$$

$$\max\left|A_m\, 2\pi f_m \cos(2\pi f_m t)\right| > \frac{\delta}{T_s}$$

$$A_m\, 2\pi f_m > \frac{\delta}{T_s}$$

or

$$\boxed{A_m > \frac{\delta}{2\pi f_m T_s}}$$

where: x(t) = input signal, δ = step size, T = period between samples, f_s = sampling frequency

Example A = 2 V, F = 3.4 kHz, and the signal is sampled 1,000,000 times per second, requires q > 2 * 3.14 * 3,400 * 2 /1,000,000 V > 42.7 mV

Granular noise occurs if the slope changes more slowly than the step size. The reconstructed signal oscillates by 1 step size in every sample. It can be reduced by decreasing the step size. This requires that the sample rate be increased. Delta Modulation requires a sampling rate much higher than twice the bandwidth. It requires oversampling in order to obtain an accurate prediction of the next input, since each encoded sample contains a relatively small amount of information. Delta Modulation requires higher sampling rates than PCM.

Differential PCM (DPCM) and ADPCM

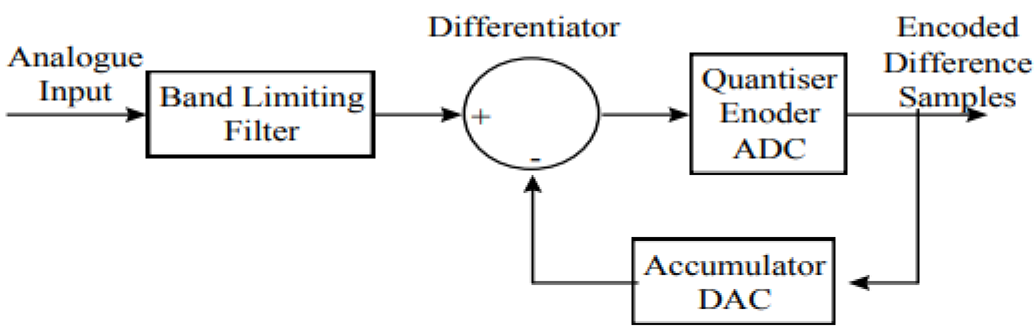


Fig. 1.18 DPCM

DPCM is also designed to take advantage of the redundancies in a typical speech waveform. In DPCM the differences between samples are quantized with fewer bits that would be used for quantizing an individual amplitude sample. The sampling rate is often the same as for a comparable PCM system, unlike Delta Modulation.

Adaptive Differential Pulse Code Modulation ADPCM is standardised by ITU-T recommendations G.721 and G.726. The method uses 32,000 bits/s per voice channel, as compared to standard PCM's 64,000 bits/s. Four bits are used to describe each sample, which represents the difference between two adjacent samples. Sampling is 8,000 times a second. It makes it possible to reduce the bit flow by half while maintaining an acceptable quality. While the use of ADPCM (rather than PCM) is imperceptible to humans, it can significantly reduce the throughput of high speed modems and fax transmissions.

The principle of ADPCM is to use our knowledge of the signal in the past time to predict the signal one sample period later, in the future. The predicted signal is then compared with the actual signal. The difference between these is the signal which is sent to line - it is the error in the prediction. However this is not done by making comparisons on the incoming audio signal - the comparisons are done after PCM coding.

To implement ADPCM the original (audio) signal is sampled as for PCM to produce a code word. This code word is manipulated to produce the predicted code word for the next sample. The new predicted code word is compared with the code word of the second sample. The result of this comparison is sent to line. Therefore we need to perform PCM before ADPCM.

The ADPCM word represents the prediction error of the signal, and has no significance itself. Instead the decoder must be able to predict the voltage of the recovered signal from the previous samples received, and then determine the actual value of the recovered signal from this prediction and the error signal, and then to reconstruct the original waveform.

ADPCM is sometimes used by telecom operators to fit two speech channels onto a single 64 kbit/s link. This was very common for transatlantic phone calls via satellite up until a few years ago. Now, nearly all calls use fibre optic channels at 64 kbit/s.

QUESTION BANK
UNIT I

SL.NO	**Part A**	**CO**	**level**
1.	Summarize the advantages of digital communication.	**1**	**2**
2.	Outline the demerit of digital communication.	**1**	**2**
3.	Illustrate sampling theorem for band limited signals and the filters to avoid aliasing.	**1**	**2**
4.	Demonstrate companding? sketch the input-output characteristic of a compressor and expander.	**1**	**3**
5.	Make use of sampling theorem the condition for aliasing?	**1**	**3**
6.	Distinguish natural and flat top sampling?	**1**	**2**
7.	Illustrate the difference between uniform and non-uniform quantization?	**1**	**3**
8.	How would you show your understanding of the components required for signal reconstruction	**1**	**3**
9.	Describe non uniform quantization.	**1**	**4**
10.	Interpret the principle of ADM.	**1**	**3**
11.	Illustrate the difference between DM and ADM.	**1**	**3**
12.	A band pass signal has the spectral range that extends from 20 kHz to 82 kHz. Find the acceptance range of sampling frequency fs	**1**	**5**

13.	A certain low pass bandlimited signal x(t) is sampled and the spectrum of the sampled version has the first guard band from 1500Hz to 1900Hz.How will you determine the sampling frequency and the maximum frequency of the signal?	**1**	**5**
14.	Find the Nyquist sampling rate for a band limited signal given by x(t)=cos1000t.	**1**	**5**
15.	Give the interpolation formula for the reconstruction of the original signal	**1**	**2**
16.	Find the Nyquist sampling rate for a band limited signal given by x(t)=cos1000t.	**1**	**3**
17.	Suggest the minimum sampling frequency for a band pass signal with frequency range from 200kHz to 300kHz.	**1**	**3**
18.	Specify the Nyquist sampling rate for the following signals: x(t)=sin(200t) and b) $x(t)sin^2(200t)$	**1**	**3**

SL.NO	**Part B**	**CO**	**LEVEL**
1.	(i)Develope your understanding on PCM transmitter and receiver with neat sketch(10) (ii) Narrate the merits and demerits of digital communication system(6)	**1**	**3**
2.	Elucidate the working of each functional blocks of a digital communication system with suitable illustration.	**1**	**2**
3.	A band limited analog signal $\sin 2\pi f_m t$ is naturally sampled using an impulse train with frequency f_s. Derive the expression for the sampled signal s(t) and plot the spectrum of s(t) for the three conditions: i) $f_s > 2f_m$ ii) $f_s = 2f_m$ and iii) $f_s < 2f_m$	**1**	**3**
4.	Make use of suitable derivation explain in detail about sampling theorem and discuss the reconstruction of the signal from its samples.	**1**	**3**
5.	Outline the process of natural sampling and how the message can be reconstructed from its samples. Also illustrate the effect of aliasing with neat sketch.	**1**	**2**
6.	(i) Describe delta modulation system in detail with a neat block diagram. (10) (ii) Derive the derivation for slope overload error(6).	**1**	**3**

7.	Explain with neat diagram the adaptive delta modulation and demodulation system in detail.	**1**	**5**
8.	(i) Illustrate the functioning of DPCM system with block diagram(10) **(ii)** Explain the structure of linear predictor?(6)	**1**	**3**
9.	Explain quantization error and derive an expression for maximum signal to noise ratio in PCM system that uses linear quantization.	**1**	**5**

UNIT II – BASEBAND PULSE TRANSMISSION - SECA1403

Base band transmission - Wave form representation of binary digits -Matched Filter- Error Rate due to noise- Nyquist's criterion for Distortionless Base band Binary Transmission- Inter symbol Interference - Ideal Nyquist channel- Raised cosine channels- Correlative level coding –Baseband M-ary PAM transmission- Equalization - Eye patterns- Companding - A law and μ law- correlation receiver.

INTRODUCTION

Naturally, noise and ISI arise in the system simultaneously. However, to understand how they affect the performance of the system, we first consider them separately channel noise. Naturally, noise and ISI arise in the system simultaneously. However, to understand how they affect the performance of the system, we first consider them separately. The device for the optimum detection of such a pulse immersed in additive white noise involves the use of a linear-time-invariant filter known as a matched filter, which is so called because its impulse response is matched to the pulse signal.

2.1 Matched Filter

A basic problem that often arises in the study of communication systems is that of detecting a pulse transmitted over a channel that is corrupted by channel noise (i.e., additive noise at the front end of the receiver). For the purpose of the discussion presented in this section, we assume that the major source of system limitation is the channel noise.

Consider then the receiver model shown in Figure 2.1, involving a linear time-invariant filter of impulse response h(t). The filter input x(t) consists of a pulse signal g(t) corrupted by additive channel noise w(t), as shown by

$$x(t) = g(t) + w(t),\ 0<t<T \qquad (2.1)$$

where T is an arbitrary observation interval. The pulse signal g(t) may represent a binary symbol

1 or in a digital communication system. The w(t) is the sample function of a white noise process of zero mean and power spectral density N /2. It is assumed that the receiver has knowledge of the waveform of the pulse signal g(t). The source of uncertainty lies in the noise w(t). The function of the receiver is to detect the pulse signal g(t) in an optimum manner, given the received signal x(t). To satisfy this requirement, we have to optimize the design of the filter so as to minimize the effects of noise at the filter output in some statistical sense, and thereby enhance the detection of the pulse signal g(t). Since the filter is linear, the resulting output y(t) may be expressed as

y(t) = g (t) + n(t) (2.2)

Where g (t) and n(t) are produced by the signal and noise components of the input x(t), respectively. A simple way of describing the requirement that the output signal component g (t) be considerably greater than the output noise component n(t) is to have the filter make the instantaneous power in the output signal g (t), measured at time t = T, as large as possible compared with the average power of the output noise n(t). This is equivalent to maximizing the peak pulse signal-to-noise ratio, defined as

$$\eta = \frac{|g_o(T)|^2}{E[n^2(t)]} \qquad (2.3)$$

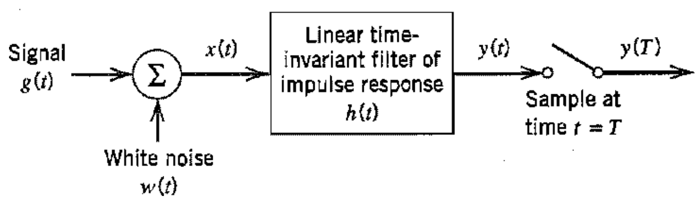


Figure 2.1 Linear receiver.

where $|g_o(T)|^2$ is the instantaneous power in the output signal, E is the statistical expectation operator, and E[n 2 (t)] is a measure of the average output noise power. The requirement is to specify the impulse response h(t) of the filter such that the output signal-to-noise ratio in Equation (2.3) is maximized.

Let G(/) denote the Fourier transform of the known signal g(t), and H(f) denote the frequency response of the filter. Then the Fourier transform of the output signal g (t)is equal to H(f)G(f), and g (t) is itself given by the inverse Fourier transform

$$g_o(t) = \int_{-\infty}^{\infty} H(f)G(f) \exp(j2\pi ft)\, df$$

(2.4)

Hence, when the filter output is sampled at time t = T, we have (in the absence of channel noise) Consider next the effect on the filter output due to the noise w(t) acting alone. The power spectral density S N {f) of the output noise n(t) is equal to the power spectral density of the input noise w(t) times the squared magnitude response \H(f)\ 2

Since w{t) is white with constant power spectral density N /2, it follows that

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

(2.6)

The average power of the output noise n(t) is therefore

E[n 2 (t)] = |__S N (f) df

Thus substituting Equations (2.5) and (2.7) into (2.3), we may rewrite the expression
for the peak pulse signal-to-noise ratio as

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

(2.7)

Our problem is to find, for a given G(f), the particular form of the frequency response H(/) of the filter that makes 17 a maximum. To find the solution to this optimization problem, we apply a mathematical result known as Schwarz's inequality to the numerator of Equation (2.8).

if we have two complex functions 4>i(x) and <j> 2 (x) in the real variable x, satisfying the conditions

where k is an arbitrary constant, and the asterisk denotes complex conjugation.

$$\eta = \frac{\left| \int_{-\infty}^{\infty} H(f)G(f) \exp(j2\pi fT)\, df \right|^2}{\frac{N_0}{2} \int_{-\infty}^{\infty} |H(f)|^2\, df} \quad (4.8)$$

2.2 Error Rate due to noise

To derive a formula for the error rate in a matched filter due to noise.

To proceed with the analysis, consider a binary PCM system based on polar non- return-to-zero (NRZ) signaling. In this form of signaling, symbols 1 and 0 are represented by positive and negative rectangular pulses of equal amplitude and equal duration. The channel noise is modeled as additive white Gaussian noise w(t) of zero mean and power spectral density N /2; the Gaussian assumption is needed for later calculations. In the signaling interval $0 \leq t \leq T_b$, , the received signal is thus written as follows:

$$x(t) = \begin{cases} +A + w(t), & \text{symbol 1 was sent} \\ -A + w(t), & \text{symbol 0 was sent} \end{cases}$$

where Tj is the bit duration, and A is the transmitted pulse amplitude. It is assumed that the receiver has acquired knowledge of the starting and ending times of each transmitted pulse; in other words, the receiver has prior knowledge of the pulse shape, but not its polarity. Given the noisy signal x(t), the receiver is required to make a decision in each signaling interval as to whether the transmitted symbol is a 1 or a 0.

The structure of the receiver used to perform this decision-making process is shown in Figure 2.4. It consists of a matched filter followed by a sampler, and then finally a

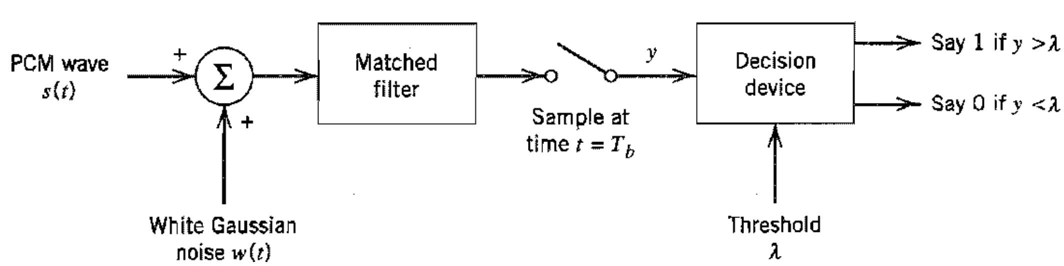


Figure 2.4 Receiver for baseband transmission of binary-encoded PCM wave using polar NRZ signaling.

decision device. The filter is matched to a rectangular pulse of amplitude A and duration T exploiting the bit-timing information available to the receiver. The resulting matched filter output is sampled at the end of each signaling interval. The presence of channel noi Se w(t) adds randomness to the matched filter output.

Let y denote the sample value obtained at the end of a signaling interval. The sample value y is compared to a preset threshold A in the decision device. If the threshold jj exceeded, the receiver makes a decision in favor of symbol 1; if not, a decision is made in favor of symbol 0. We adopt the convention that when the sample value y is exactly equal to the threshold A, the receiver just makes a guess as to which symbol was transmitted such a decision is the same as that obtained by flipping a fair coin, the outcome of which will not alter the average probability of error.

There are two possible kinds of error to be considered:

1. Symbol 1 is chosen when a was actually transmitted; we refer to this error as an error of the first kind.

2. Symbol 0 is chosen when a 1 was actually transmitted; we refer to this error as anerror of the second kind.

To determine the average probability of error, we consider these two situations separately.

Suppose that symbol was sent. Then, according to Equation (2.21), the received signal is

$$x(t) = -A + w(t), \qquad 0 \leq t \leq T_b \qquad (2.22)$$

Correspondingly, the matched filter output, sampled at time t = T b , is given by (in light of Example 2.1 with kAT b set equal to unity for convenience of presentation)

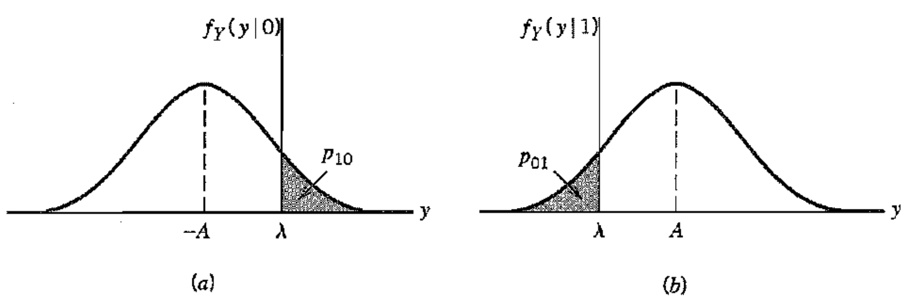


Figure 4.5 Noise analysis of PCM system (a) Probability density function of random variable y at matched filter output when 0 is transmitted (b) pdf of y when 1 is transmitted

2.3 Intersymbol Interference

The next source of bit errors in a baseband-pulse transmission system that we wish to study is intersymbol interference (ISI), which arises when the communication channel is dispersive. First of all, however, we need to address a key question: Given a pulse shape of interest, how do we use it to transmit data in M-ary form?

The answer lies in the use of discrete pulse modulation, in which the amplitude, duration, or position of the transmitted pulses is varied in a discrete manner in accordance with the given data stream. However, for the baseband transmission of digital data, the use of discrete pulse- amplitude modulation (PAM) is one of the most efficient schemes in terms of power and bandwidth utilization. Accordingly, we confine our attention to discrete PAM systems. We begin

the study by first considering the case of binary data; later in the chapter, we consider the more general case of M-ary data.

Consider then a baseband binary PAM system, a generic form of which is shown in Figure 2.7. The incoming binary sequence {b k } consists of symbols 1 and 0, each of duration T b . The pulse-amplitude modulator modifies this binary sequence into a new sequence of short pulses (approximating a unit impulse), whose amplitude a k is represented in the polar form

$$a_k = \begin{cases} +1 & \text{if symbol } b_k \text{ is } 1 \\ -1 & \text{if symbol } b_k \text{ is } 0 \end{cases} \tag{2.42}$$

The sequence of short pulses so produced is applied to a transmit filter of impulse response g(t), producing the transmitted signal

$$s(t) = \sum_k a_k g(t - kT_b) \tag{2.43}$$

The signal s(t) is modified as a result of transmission through the channel of impulse response h(t). In addition, the channel adds random noise to the signal at the receiver input. The noisy signal x(t) is then passed through a receive filter of impulse response c(t). The resulting filter output y(t) is sampled synchronously with the transmitter, with the sampling instants being determined by a clock or timing signal that is usually extracted from the receive filter output. Finally, the sequence of samples thus obtained is used to reconstruct the original data sequence by means of a decision device. Specifically, the amplitude of each sample is compared to a threshold A. If the threshold A is exceeded, a decision is made in favor of symbol

1. If the threshold A is not exceeded, a decision is made in favor of symbol 0. If the sample amplitude equals the threshold exactly, the flip of a

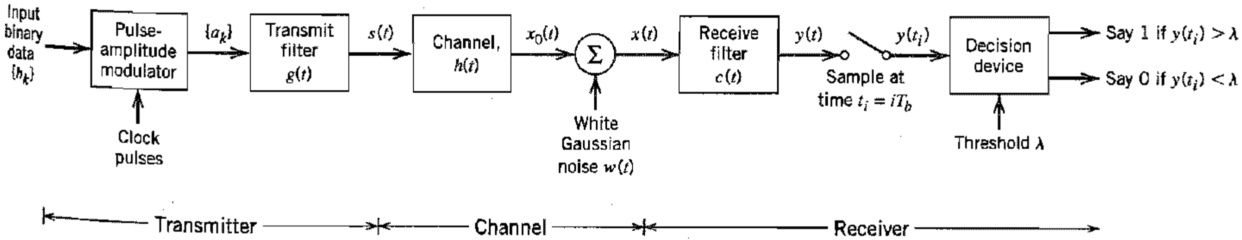


FIGURE 2.7 Baseband binary data transmission system.

fair coin will determine which symbol was transmitted (i.e., the receiver simply makes a random guess).

The receive filter output is written as

$$y(t) = \mu \sum_{k} a_k p(t - kT_b) + n(t)$$

where yu. is a scaling factor, and the pulse p(t) is to be defined.

The scaled pulse /j,p(t) is obtained by a double convolution involving the impulse response g(t) of the transmit filter, the impulse response h(t) of the channel, and the impu|se response c(t) of the receive filter, as shown by

ixp(t) = g(t) • h(t) • c(t) (2.45)

where the star denotes convolution. We assume that the pulse p(t) is normalized by setting

Since convolution in the time domain is transformed into multiplication in the frequency domain, we may use the Fourier transform to change Equation (2.45) into the equivalent form

Ptf) = G(f)H(f)C{f) (2.47)

where P{f), G(f), H[f), and C(/) are the Fourier transforms of p(t), g{t), h(t), and c{t), respectively.

Finally, the term n(t) in Equation (2.44) is the noise produced at the output of the receive filter due to the channel noise w(t). It is customary to model w(t) as a white Gaussian noise of zero mean.

The receive filter output y(t) is sampled at time t, = iT h (with i taking on integer values), yielding [in light of Equation (2.46)]

$$\begin{aligned} y(t_i) &= \mu \sum_{k=-\infty}^{\infty} a_k p[(i-k)T_b] + n(t_i) \\ &= \mu a_i + \mu \sum_{\substack{k=-\infty \\ k \neq i}}^{\infty} a_k p[(i-k)T_b] + n(t_i) \end{aligned} \qquad (2.48)$$

In Equation (2.48), the first term fia t represents the contribution of the *'th transmitted bit. The second term represents the residual effect of all other transmitted bits on the decoding of the rth bit; this residual effect due to the occurrence of pulses before and after the sampling instant t t is called intersymbol interference (EI). The last term n(t,) represents the noise sample at time tj.

2.4 Nyquist’s criterion for Distortionless Base band Binary Transmission

Typically, the frequency response of the channel and the transmitted pulse shape are specified, and the problem is to determine the frequency responses of the transmit and receive filters so as to reconstruct the original binary data sequence {b k }. The receiver does this by extracting and then decoding the corresponding sequence of coefficients, {a^}, from the output y(t). Theextraction involves sampling the output y(t) at time t = il b . The decoding requires that the weighted pulse contribution af. for be free from ISI due to the overlapping tails of all other weighted pulse contributions represented by k ^ I

$$a_k p(iT_b - kT_b) \qquad\qquad k \neq i.$$

This, in turn, requires that we control the overall pulse p(t), as shown by

$$p(iT_b - kT_b) = \begin{cases} 1, & i = k \\ 0, & i \neq k \end{cases} \tag{2.49}$$

where p(0) = 1, by normalization. If p(t) satisfies the conditions of Equation (2.49), the receiver output y{t,) given in Equation (2.48) simplifies to (ignoring the noise term)

$$y(t_i) = \mu a_i \qquad \text{for all } i$$

which implies zero intersymbol interference. Hence, the two conditions of Equation (2.49) ensure perfect reception in the absence of noise.

Ideal Nyquist Channel

The simplest way of satisfying Equation (2.53) is to specify the frequency function P(f) t0 be in the form of a rectangular function, as shown by

$$\begin{aligned} P(f) &= \begin{cases} \dfrac{1}{2W}, & -W < f < W \\ 0, & |f| > W \end{cases} \\ &= \frac{1}{2W}\,\mathrm{rect}\left(\frac{f}{2W}\right) \end{aligned} \tag{2.54}$$

where rect(/) stands for a rectangular function of unit amplitude and unit support centered on / = 0, and the overall system bandwidth W is defined by

$$W = \frac{R_b}{2} = \frac{1}{2T_b} \tag{2.55}$$

According to the solution described by Equations (2.54) and (2.55), no frequencies of absolute value exceeding half the bit rate are needed. Hence, from Fourier-transform pair 2 of Table A6.3 we find that a signal waveform that produces zero inter symbol interference is defined by the sine function:

$$\begin{aligned} p(t) &= \frac{\sin(2\pi W t)}{2\pi W t} \\ &= \mathrm{sinc}(2Wt) \end{aligned} \tag{2.56}$$

The special value of the bit rate R h = 2 W is called the Nyquist rate, and W is itself called the Nyquist bandwidth. Correspondingly, the ideal baseband pulse transmission system described by Equation (2.54) in the frequency domain or, equivalently, Equation (2.56) in the time domain, is called the ideal Nyquist channel.

Figures 2.8a and 2.8b show plots of P(f) and p(t), respectively. In Figure 2.8a, the normalized form of the frequency function P{f) is plotted for positive and negative frequencies. In Figure 2.8b, we have also included the signaling intervals and the corresponding centered sampling instants. The function p(t) can be regarded as the impulse response of an ideal low-pass filter with passband magnitude response 1/2 W and bandwidth W.

The function p{t) has its peak value at the origin and goes through zero at integer multiply of the bit duration T b . It is apparent that if the received waveform y{t) is sampled at the instants of time t = 0, ±T b , ±2T b , • • • , then the pulses defined by fip(t - iT b) with arbitrary amplitude p and index i = 0, ±1, ±2, • • • , will not interfere with each other. This condition is illustrated in Figure 2.9 for the binary sequence 1011010.

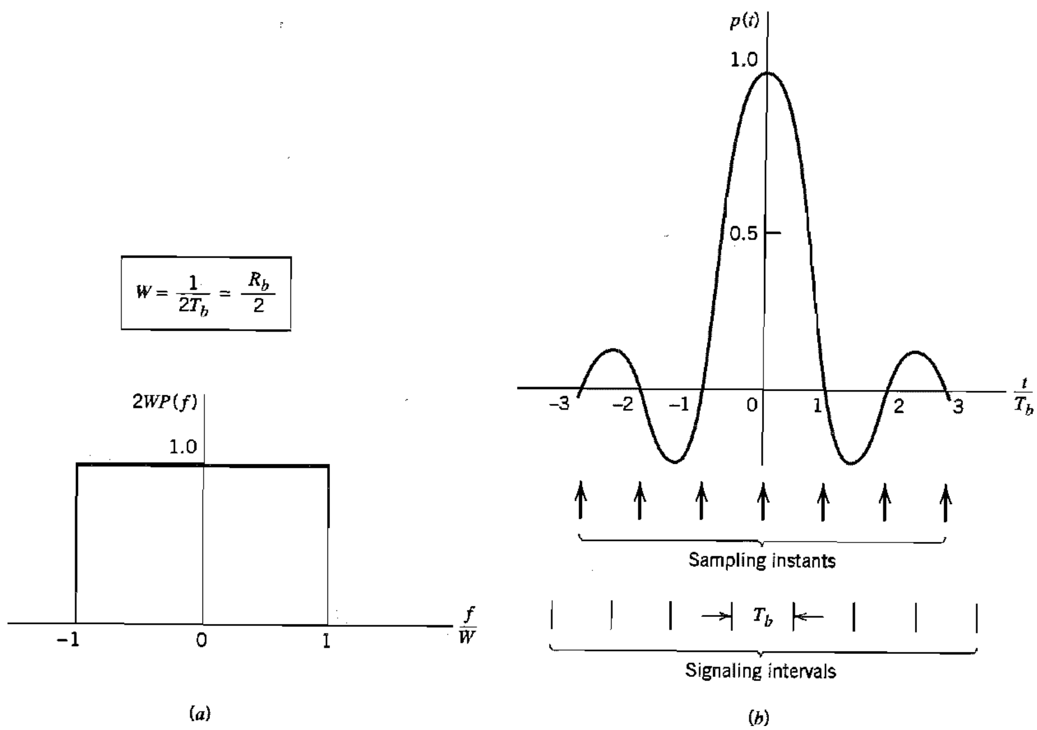


Figure 2.8 (a) Ideal magnitude response (b) Ideal basic pulse shape

There are two practical difficulties that make it an undesirable objective for system design:

1. It requires that the magnitude characteristic of P(f) be flat from -WtoW, and zero elsewhere. This is physically unrealizable because of the abrupt transitions at flu band edges ± W.

2. The function p(t) decreases as l/|f | for large \t\, resulting in a slow rate of decay. This is also caused by the discontinuity of P(f) at ±W. Accordingly, there is p rac . tically no margin of error in sampling times in the receiver.

Raised Cosine Spectrum

We may overcome the practical difficulties encountered with the ideal Nyquist channel by extending the bandwidth from the minimum value W = R b /2 to an adjustable value between W and 2 W. We now specify the overall frequency response P(/) to satisfy a condition more elaborate than that for the ideal Nyquist channel

This frequency response consists of a flat portion and a ro//o/yportion that has a sinusoidal form, as follows:

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

The frequency parameter f 1 and bandwidth W are related by

$$\alpha = 1 - \frac{f_1}{W}$$

The parameter a is called the rolloff factor; it indicates the excess bandwidth over the ideal solution, W. Specifically, the transmission bandwidth B T is defined by

$$\begin{aligned} B_T &= 2W - f_1 \\ &= W(1 + \alpha) \end{aligned}$$

The frequency response P(f), normalized by multiplying it by 2 W, is plotted in Figure 2.10a for three values of a, namely, 0, 0.5, and 1. We see that for a = 0.5 or 1,

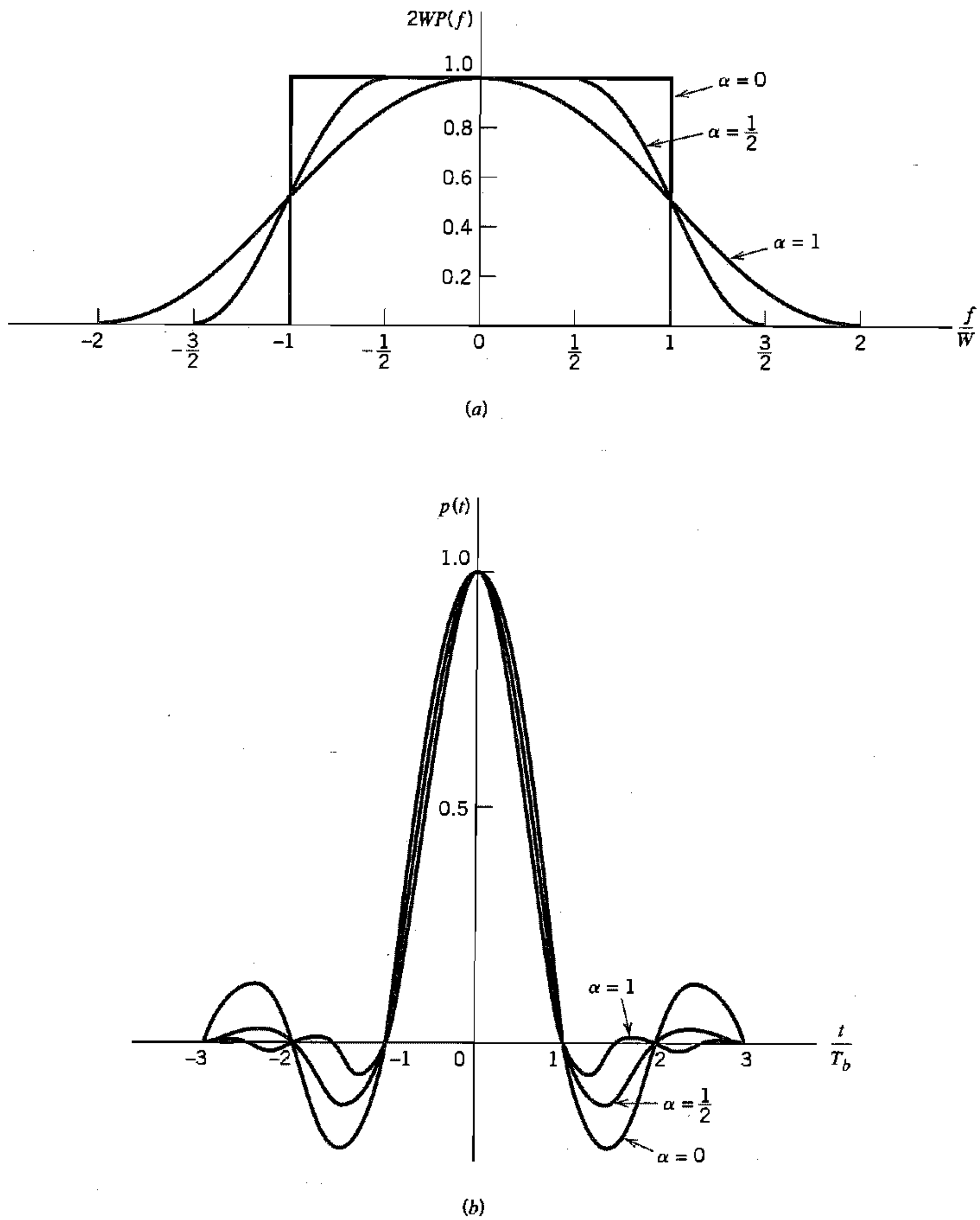


Figure 2.10 Response for different rolloff factors. (a) Frequency response (b) Time response

2.5 Correlative level coding

Thus far we have treated intersymbol interference as an undesirable phenomenon that

produces a degradation in system performance. Indeed, its very name connotes a nuisance effect Nevertheless, by adding intersymbol interference to the transmitted signal in a controlled manner, it is possible to achieve a signaling rate equal to the Nyquist rate of 2W symbols per second in a channel of bandwidth W Hertz. Such schemes are called correlatwe-level codmg or partial-response signaling schemes. 5 The design of these schemes is based on the fo lowing premise: Since intersymbol interference introduced into the transmitted signal is known, its effect can be interpreted at the receiver in a deterministic way. Thus correlative-level coding may be regarded as a practical method of achieving the theoretical maximum signaling rate of 2Wsymbols per second in a bandwidth of W Hertz as postulated by Nyquist, using realizable and perturbation-tolerant filters.

Duobinary Signaling

The basic idea of correlative-level coding will now be illustrated by considering the specific example of duobmary signaling, where "duo" implies doubling of the transmission capacity of a straight binary system. This particular form of correlative-level coding is also called class I partial response.

Consider a binary input sequence $\{b_k\}$ consisting of uncorrelated binary symbols 1 and 0, each having duration T_b. As before, this sequence is applied to a pulse-amplitude modulator producing a two-level sequence of short pulses (approximating a unit impulse), whose amplitude a_k is defined by

$$a_k = \begin{cases} +1 & \text{if symbol } b_k \text{ is } 1 \\ -1 & \text{if symbol } b_k \text{ is } 0 \end{cases} \quad (2.65)$$

When this sequence is applied to a duobinary encoder, it is converted into a three- level output, namely, -2, 0, and +2. To produce this transformation, we may use the scheme shown in Figure 2.11. The two-level sequence {a k \ is first passed through a simple filter involving a single delay element and summer. For every unit impulse applied to the

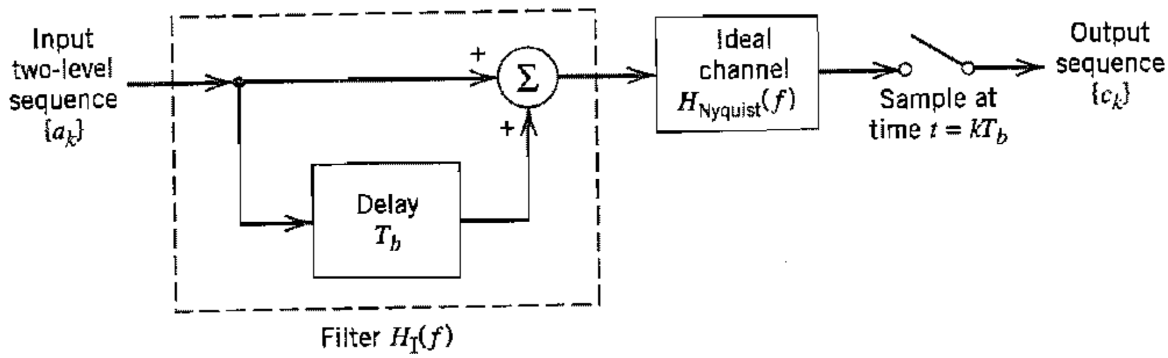


Figure 2.11 Duo binary signaling scheme.

input of this filter, we get two unit impulses spaced T b seconds apart at the filter output. We may therefore express the duobinary coder output c k as the sum of the present input pulse a_k and its previous value a_{k-u} u as shown by

$$c_k = a_k + a_{k-1} \tag{2.66}$$

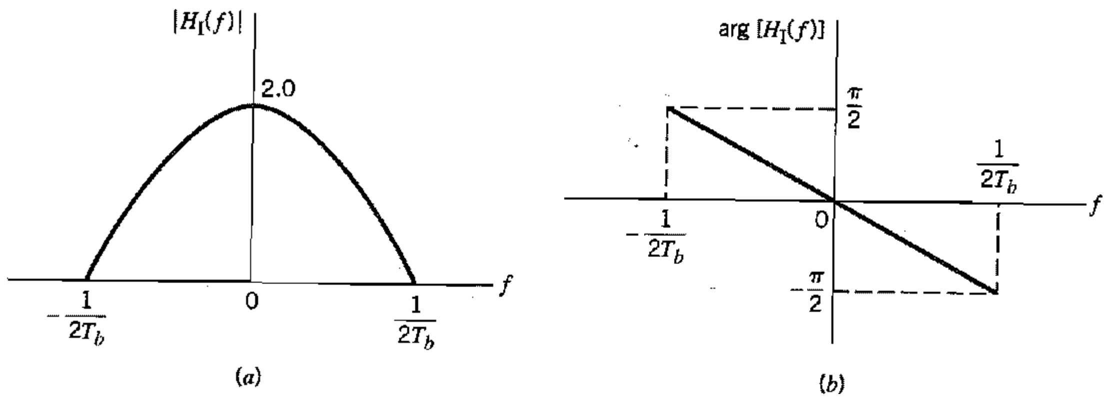


Figure 2.12 Frequency response of the duobinary conversion filter. (a) Magnitude response. (b) Phase response.

Note that unlike the linear operation of duobinary coding, the precoding described by eqn 2.72 is a nonlinear operation.

$$c_k = \begin{cases} 0 & \text{if data symbol } b_k \text{ is } 1 \\ \pm 2 & \text{if data symbol } b_k \text{ is } 0 \end{cases} \tag{2.75}$$

Which is illustrated in example 2.3 . From eqn 2.75 we deduce the following decision rule for detecting the original binary sequence $\{b_k\}$ from $\{c_k\}$

$$\text{If } |c_k| < 1, \quad \text{say symbol } b_k \text{ is } 1$$
$$\text{If } |c_k| > 1, \quad \text{say symbol } b_k \text{ is } 0$$

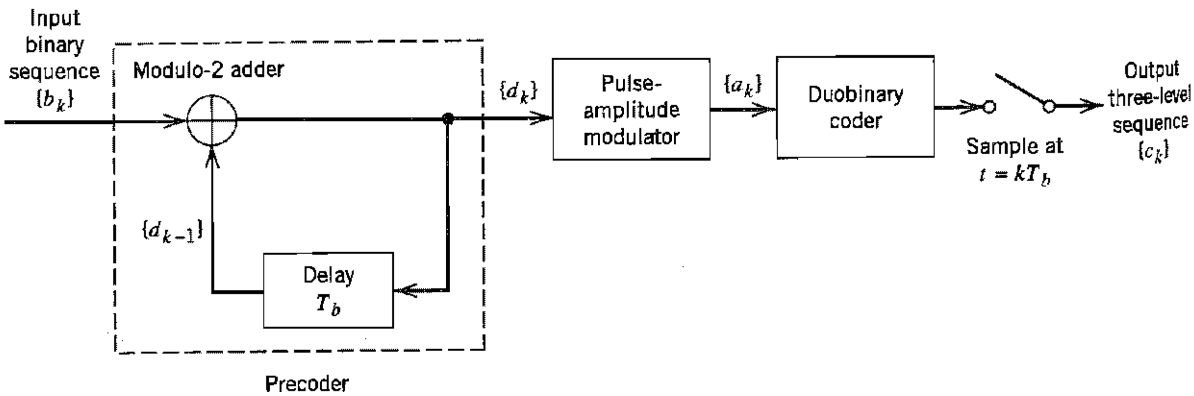


Figure 2.14 A precoded duobinary scheme

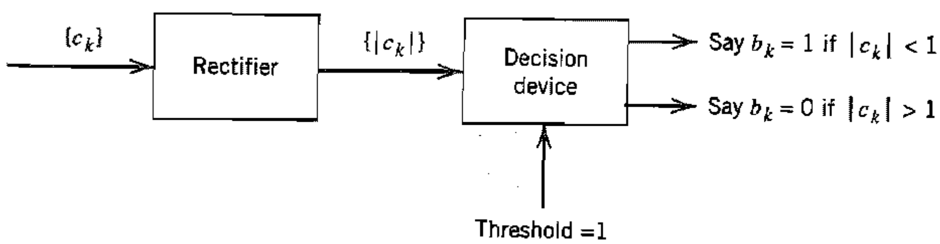


Figure 2.15 Detector for recovering original binary sequence from the precoded duobinary coder output.

When | c t | = 1, the receiver simply makes a random guess in favor of symbol 1 or 0. According to this decision rule, the detector consists of a rectifier, the output of which is compared in a decision device to a threshold of 1. A block diagram of the detector is shown m Figure 2.15. A useful feature of this detector is that no knowledge of any input sample other than the present one is required. Hence, error propagation cannot occur in the detector of Figure 2.15.

Modified Duobinary Signaling

In the duobinary signaling technique the frequency response H(f), and consequently the power spectral density of the transmitted pulse, is nonzero at the origin. This is considered to be an undesirable feature in some applications, since many communications channels cannot transmit a DC component. We may correct for this deficiency by using the class IV partial response or modified duobinary technique, which involves a correlation span of two binary digits. This special form of correlation is achieved by subtracting amplitude modulated pulses spaced lT b seconds apart, as indicated in the block diagram of Figure 2.16

TABLE 2. 1 Illustrating Example 4.3 on 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
Binary sequence obtained by applying decision rule of Eq. (4.76)		0	0	1	0	1	1	0

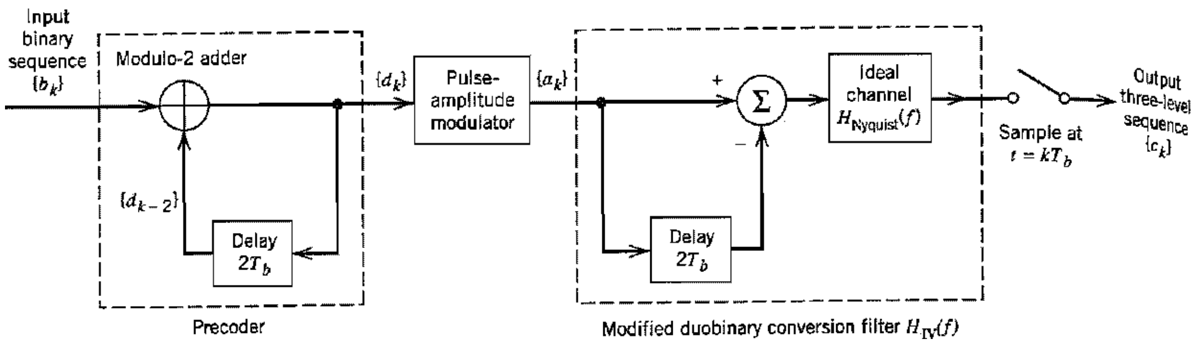


Figure 2.16 Modified duobinary signaling scheme.

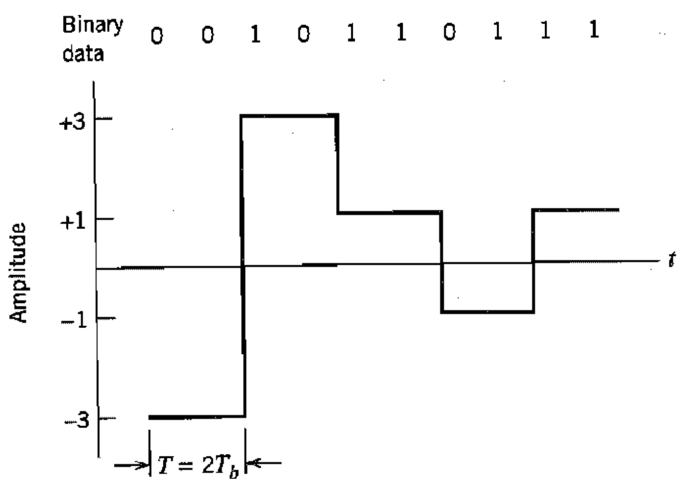
2.6 Baseband M-ary PAM transmission

In the baseband binary PAM system of Figure 2.7, the pulse-amplitude modulator produces binary pulses, that is, pulses with one of two possible amplitude levels. On the other hand, in a baseband M-ary PAM system, the pulse-amplitude modulator produces one of M possible amplitude levels with M > 2. This form of pulse modulation is illustrated in Figure 2.20(a) for the case of a quaternary (M = 4) system and the binary data sequence 0010110111. The waveform shown in Figure 2.20a is based on the electrical representation for each of the four possible dibits (pairs of bits) given in Figure 4 .20(b). Note that this representation is Gray encoded, which means that any dibit in the quaternary alphabet differs from an adjacent dibit in a single bit position.

In an M-ary system, the information source emits a sequence of symbols from an alphabet that consists of M symbols. Each amplitude level at the pulse-amplitude modulator output corresponds to a distinct symbol, so that there are M distinct amplitude levels to be transmitted. Consider then an M-ary PAM system with a signal alphabet that contains M equally likely and statistically independent symbols, with the symbol duration denoted by T seconds. We refer to 1/T as the signaling rate of the system, which is expressed in symbols per second, or bauds. It is informative to relate the signaling rate of this system to that of an equivalent binary PAM system for which the value of M is 2 and the successive binary symbols 1 and are equally likely and statistically independent, with the duration of either symbol denoted by T b seconds. Under the conditions described here, the binary PAM system produces information at the rate of 1/T 6 bits per seconds. We also observe that in the case of a quaternary PAM system, for example, the four possible symbols may be identified with the dibits 00, 01, 10, and 11. We thus see that each symbol represents 2 bits of information, and 1 baud is equal to 2 bits per second. We may generalize this result by stating that in an M-ary PAM system, 1 baud is equal to log! M bits per second, and the symbol duration T of the M-ary PAM system is related to the bit duration T b of the equivalent binary PAM system as

$$T = T_b \log_2 M \qquad (2.84)$$

Therefore, in a given channel bandwidth, we find that by using an M-ary PAM system, we are able to transmit information at a rate that is log 2 M faster than the corresponding binary PAM system. However, to realize the same average probability of symbol error, an M-ary PAM system requires more transmitted power. Specifically, we find that for M much larger than 2 and an average probability of symbol error small compared to 1, the transmitted power must be increased by the factor M 2 /log 2 M, compared to a binary PAM system.



(a)

Dibit	Amplitude
00	−3
01	−1
11	+1
10	+3

(b)

Figure 2.20 Output of a quaternary system. (a)Waveform.
(b) Representation of the 4 possible dibits, based on gray encoding.

In a baseband M-ary system, first of all, the sequence of symbols emitted by the information source is converted into an M-level PAM pulse train by a pulse-amplitude modulator at the transmitter input. Next, as with the binary PAM system, this pulse train is shaped by a transmit filter and then transmitted over the communication channel, which corrupts the signal waveform with both noise and distortion. The received signal is passed through a receive filter and then sampled at an appropriate rate in synchronism with the transmitter. Each sample is compared with preset threshold values (also called slicing levels), and a decision is made as to which symbol was transmitted. We therefore find that the designs of the pulse-amplitude modulator and the decision-making device in an M-ary PAM are more complex than those in a binary PAM system. Intersymbol interference, noise, and imperfect synchronization cause errors to appear at the receiver output. The transmit and receive filters are designed to minimize these errors. Procedures used for the design of these filters are similar to those discussed in Sections 4.5 and 4.6 for baseband binary PAM systems.

2.7 Adaptive Equalization

In this section we develop a simple and yet effective algorithm for the adaptive equalization of a linear channel of unknown characteristics. Figure 2.28 shows the structure of an adaptive synchronous equalizer, which incorporates the matched filtering action. The algorithm used to adjust the equalizer coefficients assumes the availability of a desired response. One's first reaction to the availability of a replica of the transmitted signal is: If such a signal is available at the receiver, why do we need adaptive equalization? To answer this question, we first note that a typical telephone channel changes little during an average data call. Accordingly, prior to data transmission, the equalizer is adjusted under the guidance

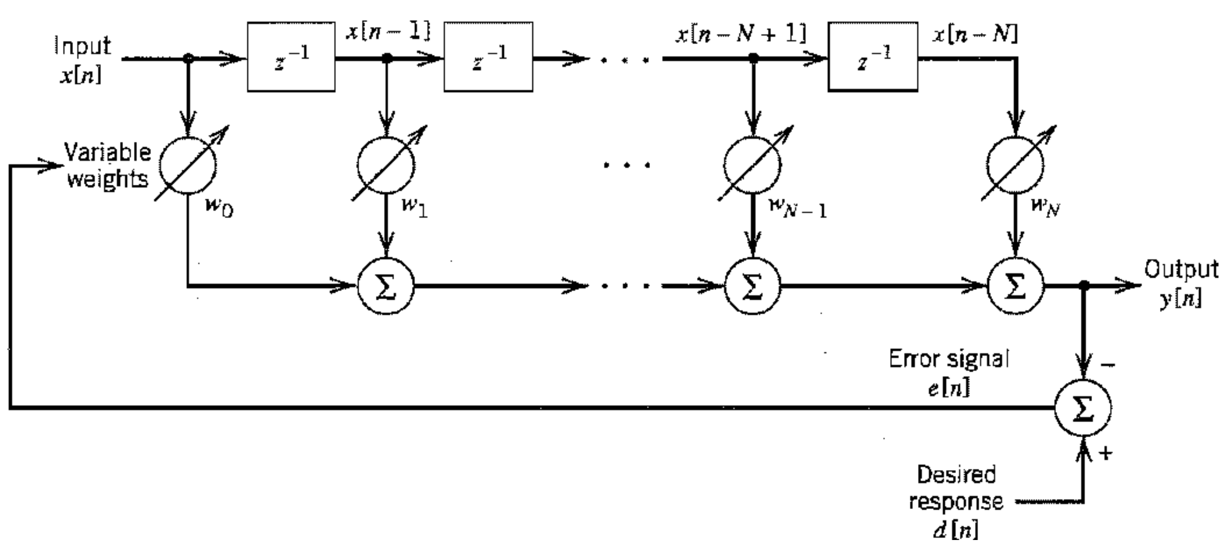


Figure 2.28 Block diagram of adaptive equalizer.

of a training sequence transmitted through the channel. A synchronized version of this training sequence is generated at the receiver, where (after a time shift equal to the transmission delay through the channel) it is applied to the equalizer as the desired response. A training sequence commonly used in practice is the pseudonotse (PN) sequence, which consists of a deterministic periodic sequence with noise-like characteristics. Two identical PN sequence generators are used, one at the transmitter and the other at the receiver. When the training process is completed, the PN sequence generator is switched off, and the adaptive equalizer is ready for normal data transmission.

Least-Mean-Square Algorithm

To simplify notational matters, we let

$$x[n] = x(nT)$$
$$y[n] = y(nT)$$

Then, the output $y[n]$ of the tapped-delay-line equalizer in response to the input sequence $\{x[n]\}$ is defined by the discrete convolution sum (see Figure 4.28)

$$y[n] = \sum_{k=0}^{N} w_k x[n-k] \qquad (4.112)$$

where w_k is the weight at the kth tap, and $N+1$ is the total number of taps. The tap-weights constitute the adaptive filter coefficients. We assume that the input sequence $\{x[n]\}$ has finite energy. We have used a notation for the equalizer weights in Figure 4.28 that is different from the corresponding notation in Figure 4.27 to emphasize the fact that the equalizer in Figure 4.28 also incorporates matched filtering.

Let a[n] denote the desired response defined as the polar representation or the m» transmitted binary symbol. Let e[n] denote the error signal denned as the difference between the desired response a[n] and the actual response y [»] of the equalizer, as shown dj

$$e[n] = a[n] - y[n]$$

In the least-mean-square (LMS) algorithm for adaptive equalization, the error signal* actuates the adjustments applied to the individual tap weights of the equalizer as algorithm proceeds from one iteration to the next. A derivation of the LMS algorithm adaptive prediction was presented in Section 3.13. Recasting Equation (3.72) into its most general form, we may state the formula for the LMS algorithm in words as follows:

$$\begin{pmatrix}\text{Updated value}\\ \text{of } k\text{th tap-}\\ \text{weight}\end{pmatrix} = \begin{pmatrix}\text{Old value}\\ \text{of } k\text{th tap-}\\ \text{weight}\end{pmatrix} + \begin{pmatrix}\text{Step-size}\\ \text{parameter}\end{pmatrix} \cdot \begin{pmatrix}\text{Input signal}\\ \text{applied to}\\ k\text{th tap-}\\ \text{weight}\end{pmatrix} \begin{pmatrix}\text{Error}\\ \text{signal}\end{pmatrix} \quad (2.114)$$

now summarize the LMS algorithm for adaptive equalization as follows:

1. Initialize the algorithm by setting $\hat{\mathbf{w}}[1] = 0$ (i.e., set all the tap-weights of the equalizer to zero at $n = 1$, which corresponds to time $t = T$).
2. For $n = 1, 2, \ldots$, compute

$$y[n] = \mathbf{x}^T[n]\hat{\mathbf{w}}[n]$$
$$e[n] = a[n] - y[n]$$
$$\hat{\mathbf{w}}[n+1] = \hat{\mathbf{w}}[n] + \mu e[n]\mathbf{x}[n]$$

   where μ is the step-size parameter.
3. Continue the iterative computation until the equalizer reaches a "steady state," by which we mean that the actual mean-square error of the equalizer essentially reaches a constant value.

The LMS algorithm is an example of a feedback system, as illustrated in the block diagram of Figure 4.29, which pertains to the kth filter coefficient. It is therefore possible

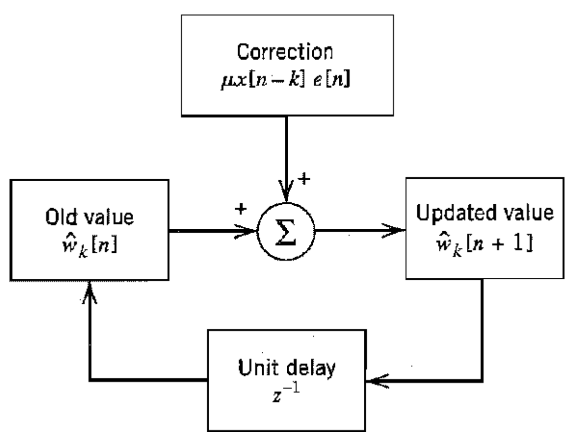


Figure 2.29 Signal flow graph representation of the LMS algorithm involving the kth tap weight.

Operation of the Equalizer

There are two modes of operation for an adaptive equalizer, namely, the training mode and decision-directed mode, as shown in Figure 4.30. During the *training mode*, as explained previously, a known PN sequence is transmitted and a synchronized version of it is generated in the receiver, where (after a time shift equal to the transmission delay) it is applied to the adaptive equalizer as the desired response; the tap-weights of the equalizer are thereby adjusted in accordance with the LMS algorithm.

When the training process is completed, the adaptive equalizer is switched to its second mode of operation: the *decision-directed mode*. In this mode of operation, the error signal is defined by

$$e[n] = \hat{a}[n] - y[n] \tag{4.120}$$

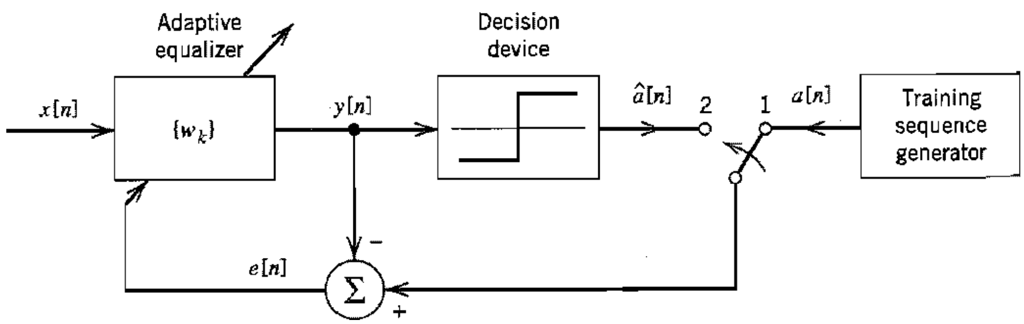


Figure 2.30 Illustrating the two operating modes of an adaptive equalizer, for the training mode, the switch is in position 1; and for the tracking mode, it is moved to position 2.

2.8 Eye patterns

In previous sections of this chapter we have discussed various techniques for dealing with the effects of channel noise and intersymbol interference on the performance of a baseband pulse-transmission system. In the final analysis, what really matters is how to evaluate the combined effect of these impairments on overall system performance in an operational environment. An experimental tool for such an evaluation in an insightful manner is the so-called eye pattern, which is defined as the synchronized superposition of all possible realizations of the signal of interest (e.g., received signal, receiver output) viewed within a particular signaling interval. The eye pattern derives its name from the fact that it resembles the human eye for binary waves. The interior region of the eye pattern is called the eye opening.

An eye pattern provides a great deal of useful information about the performance of a data transmission system, as described in Figure 2.33. Specifically, we make the following statements:

► The width of the eye opening defines the time interval over which the received signal can be sampled without error from intersymbol interference; it is apparent that the preferred time for sampling is the instant of time at which the eye is open the widest.

*■ The sensitivity of the system to timing errors is determined by the rate of closure of the eye as the sampling time is varied.

> The height of the eye opening, at a specified sampling time, defines the noise margin of the system.

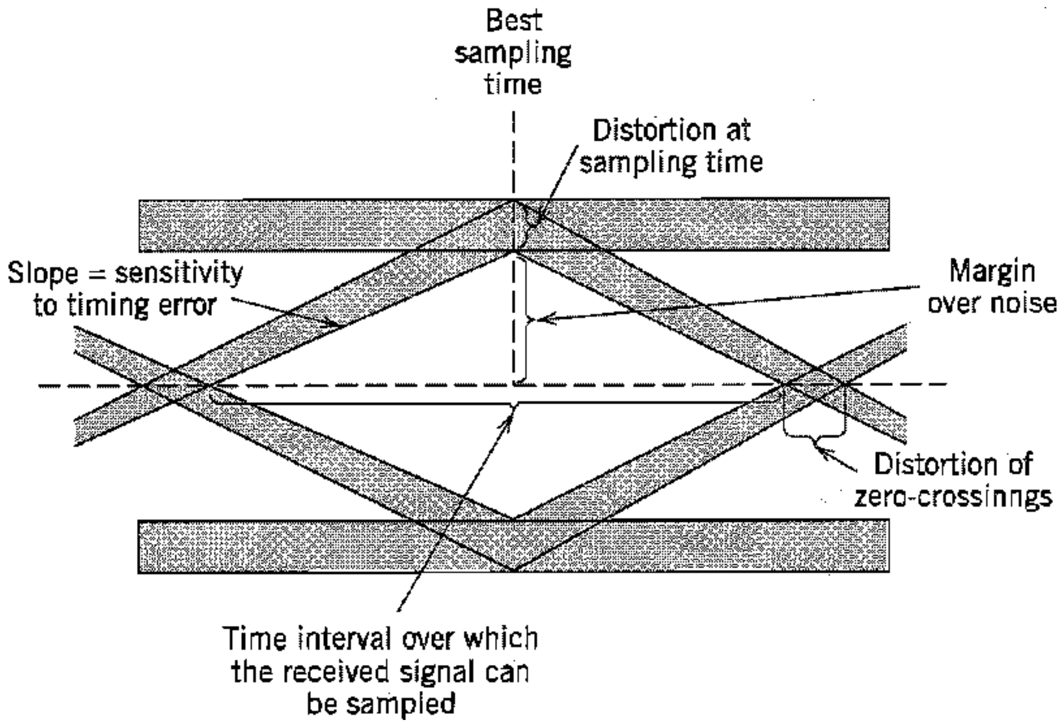
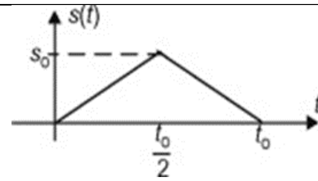


Figure 2.33 Interpretation of the eye pattern.

QUESTION BANK
UNIT II

PART-A
1. State Nyquist's criterion for distortion less transmission of baseband data.
2. Outline the causes for inter symbol interference.
3. Give any two properties of Matched filter.
4. Plot the impulse response of the matched filter for the input pulse given.



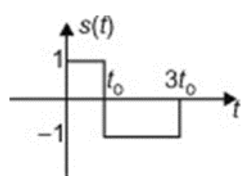
5. Distinguish matched filter receiver and correlation receiver.
6. Summarize the need of for channel Line Coding.
7. Draw the Manchester code format for the data 1101011.
8. For the binary data 10110001, show the unipolar-RZ format representation.
9. Outline the role of equalizers in digital communication receivers.
10. Narrate the conditions that should be satisfied for reducing inter symbol interference.

PART-B

1.Draw the time domain representation of the binary code '11100110' for following line-encoding schemes and outline the rules for each scheme. (16 marks)

i. Unipolar NRZ-L, NRZ-M,NRZ-S
ii. Bipolar NRZ-L, NRZ-M,NRZ-S
iii. Unipolar Return to zero, Bipolar Return to zero
iv. Return to Zero AMI
v. Biphase-L, Biphase-M and Biphase-S

2. a) Determine the impulse response of the filter matched to the input given below. Assume that the filter is designed to maximize the output SNR at time t = 3to. Also, plot the output of the matched filter. (8 marks)



b) Derive the expression for probability of error of a matched filter for a bipolar NRZ data. (8 marks)

3. a) Discuss in detail about inter symbol interference (ISI) and the Nyquist criterion for minimizing ISI. (8 marks)

b) Describe how correlative coding could eliminate ISI. (8 marks)

4. a) A binary PAM wave is to be transmitted over a low-pass channel with bandwidth of 75 kHz. The bit duration is 10µs. Find a raised-cosine pulse spectrum that satisfies these requirements. (8 marks)

b) Illustrate the principle of duo-binary encoding scheme with an example. (8 marks)

5. a) Describe the realisation of a non-uniform quantizer using a compressor and a uniform quantizer and plot its input and output characteristics. (10 marks)

b) Distinguish A-law and μ-law companding. (6 marks)

6. a) Elucidate the structure of eye pattern and explain how would you analyse the level of ISI degradation, dispersion and jitter from the eye diagram. (12 marks)

b. Explain how equalizers can be used to undo the distortions introduced in the channel.

(4 marks)

UNIT III – DIGITAL MODULATION TECHNIQUES - SECA1403

Introduction - ASK- FSK - PSK- coherent modulation techniques-BFSK-BPSK-signal space diagram-robability of error-Coherent Quadrature modulation techniques- QPSK-signal space diagram-probability of error- Non coherent modulation techniques-M-ary modulation techniques - Vectorial view of MPSK and MFSK

DIGITAL MODULATION TECHNIQUES

Given a binary source that emits symbols 0 and 1, the modulation process involves switching or keying the amplitude, phase, or frequency of a sinusoidal carrier wave between a pair of possible values in accordance with symbols 0 and 1. To be more specific, consider the sinusoidal carrier

$$c(t) = A_c \cos(2\pi f_c t + \phi_c)$$

---------(1)

Where A_c is the carrier amplitude, f_c is the carrier frequency, and ϕ_c is the carrier phase. Given these three parameters of the carrier(t), we may now identify three distinct forms of binary modulation:

1. Binary amplitude shift-keying (BASK), in which the carrier frequency and carrier phase are both maintained constant, while the carrier amplitude is keyed between the two possible values used to represent symbols 0 and 1.

2. Binary phase-shift keying (BPSK), in which the carrier amplitude and carrier frequency are both maintained constant, while the carrier phase is keyed between the two possible values (e.g., 0° and 180°) used to represent symbols 0 and 1.

3. Binary frequency-shift keying (BFSK), in which the carrier amplitude and carrier phase are both maintained constant, while the carrier frequency is keyed between the two possible values used to represent symbols 0 and 1.

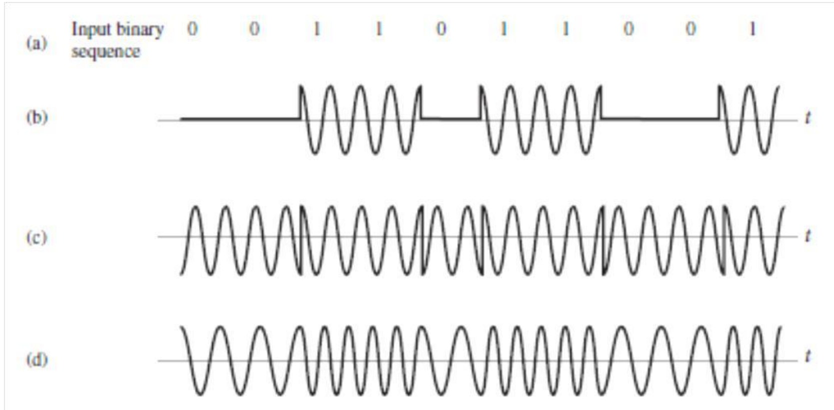


Fig. 3 Types of Binary signaling Information

The three basic forms of signaling binary information. (a) Binary data stream. (b) Amplitude-shift keying. (c) Phase-shift keying. (d) Frequency-shift keying with continuous phase. In the digital communications literature, the usual practice is to assume that the carrier c(t) has unit energy measured over one symbol (bit) duration.

$$A_c = \sqrt{\frac{2}{T_b}} \qquad \text{---------(2)}$$

Where T_b is the bit duration. Using the terminology of Eq. (2), we may thus express the Carrier c(t) in the equivalent form

$$c(t) = \sqrt{\frac{2}{T_b}}\cos(2\pi f_c t + \phi_c) \qquad \text{----------(3)}$$

The spectrum of a digitally modulated wave, exemplified by BASK, BPSK and BFSK, is centered on the carrier frequency f_c implicitly or explicitly. Moreover, as with analog modulation, it is normal practice to assume that the carrier frequency f_c is large compared with the "bandwidth" of the incoming binary data stream that acts as the modulating signal. This band-pass assumption has certain implications, as discussed next. To be specific, consider a linear modulation scheme for which the modulated wave is defined by

$$s(t) = b(t)c(t) \qquad \text{--------(4)}$$

where b(t) denotes an incoming binary wave. Then, setting the carrier phase ϕ_c fo convenience of presentation, we may use Eq. (3) to express the modulated wave as

$$s(t) = \sqrt{\frac{2}{T_b}}\, b(t) \cos(2\pi f_c t) \qquad \text{----------(5)}$$

Under the assumption where $f_c >> w$ is the bandwidth of the binary wave there will be no spectral overlap in the generation of s(t) (i.e., the spectral content of the modulated wave for positive frequencies is essentially separated from its spectral content for negative frequencies). Another implication of the band-pass assumption is that we may express the transmitted signal energy per bit as

$$\begin{aligned} E_b &= \int_0^{T_b} |s(t)|^2\, dt \\ &= \frac{2}{T_b} \int_0^{T_b} |b(t)|^2 \cos^2(2\pi f_c t)\, dt \end{aligned} \qquad \text{--------(6)}$$

Using the trigonometric identity

$$\cos^2(2\pi f_c t) = \frac{1}{2}[1 + \cos(4\pi f_c t)]$$

we may rewrite Eq. (6) as

$$E_b = \frac{1}{T_b} \int_0^{T_b} |b(t)|^2\, dt + \frac{1}{T_b} \int_0^{T_b} |b(t)|^2 \cos(4\pi f_c t)\, dt \qquad \text{---------(7)}$$

The band-pass assumption implies that $|b(t)|^2$ is essentially constant over one complete cycle of the sinusoidal wave $\cos(4\pi f_c t)$ which, in turn, means that

$$\int_0^{T_b} |b(t)|^2 \cos(4\pi f_c t)\, dt \approx 0$$

Accordingly, we may approximate Eq. (7) as

$$E_b \approx \frac{1}{T_b}\int_0^{T_b} |b(t)|^2\, dt \qquad \text{------------(8)}$$

In words, for linear digital modulation schemes governed by Eq. (5), the transmitted signal energy (on a per bit basis) is a scaled version of the energy in the incoming binary wave responsible for modulating the sinusoidal carrier.

BINARY AMPLITUDE-SHIFT KEYING

Binary amplitude-shift keying (BASK) is one of the earliest forms of digital modulation used in radio telegraphy at the beginning of the twentieth century. To formally describe BASK, consider a binary data stream b(t)

$$b(t) = \begin{cases} \sqrt{E_b}, & \text{for binary symbol 1} \\ 0, & \text{for binary symbol 0} \end{cases}$$

which is of the ON–OFF signaling variety. That is,b(t) is defined by

Then, multiplying b(t) by the sinusoidal carrier wave of Eq. (3) with the phase set ϕ_c equal to zero for convenience of presentation, we get the BASK wave

$$s(t) = \begin{cases} \sqrt{\frac{2E_b}{T_b}}\cos(2\pi f_c t), & \text{for symbol 1} \\ 0, & \text{for symbol 0} \end{cases} \qquad \text{---------(9)}$$

The carrier frequencyf_c may have an arbitrary value, consistent with transmitting the modulated signal anywhere in the electromagnetic radio spectrum, so long as it satisfies the bond-pass assumption.

When a bit duration is occupied by symbol 1, the transmitted signal energy E_b is When the bit duration is occupied by symbol 0, the transmitted signal energy is zero. On this basis, we may express the average transmitted signal energy as

$$E_{av} = \frac{E_b}{2} \qquad \text{---------(10)}$$

GENERATION AND DETECTION OF ASK SIGNALS

From Eqs. (8) and (9), we readily see that a BASK signal is readily generated by using a product modulator with two inputs. One input, the ON–OFF signal of Eq. (8), is the modulating signal. The sinusoidal carrier wave supplies the other input.

$$c(t) = \sqrt{\frac{2}{T_b}}\cos(2\pi f_c t)$$

------(11)

SPECTRAL ANALYSIS OF BASK

Consider a binary data stream that consists of a square wave, the amplitude of which alternates between the constant levels and zero every T_b seconds. The square wave is centered on the origin for convenience of the presentation. The objective of the experiment is twofold:

(i) To investigate the effect of varying the carrier frequency on the power spectrum of the BASK signal assuming that the square wave is fixed. Recall that the power spectrum of a signal (expressed in decibels) is defined as 10 times the logarithm (to base 10) of the squared magnitude (amplitude) spectrum of the signal.

(ii) To investigate the effect of varying the frequency of the square wave on the spectrum of the BASK signal, assuming that the sinusoidal carrier wave is fixed. For the purpose of computer evaluation, we set the carrier frequency $f_c=n/T_b$ where n is an integer. This choice of the carrier frequency f_c permits the simulation of a band-pass system on a digital computer without requiring $f_c>>1/T_b$ the only restriction on the choice is to make sure that spectral overlap is avoided. (We follow this practice when performing computer experiments as we go forward in the study of other digital modulation schemes.)

BINARY PHASE-SHIFT KEYING (BPSK)

In the simplest form of phase-shift keying known as binary phase-shift keying (BPSK), the pair of signals $s_1(t)$ and $s_2(t)$ used to represent symbols 1 and 0, respectively, are defined by

$$s_i(t) = \begin{cases} \sqrt{\frac{2E_b}{T_b}}\cos(2\pi f_c t), & \text{for symbol 1 corresponding to } i = 1 \\ \sqrt{\frac{2E_b}{T_b}}\cos(2\pi f_c t + \pi) = -\sqrt{\frac{2E_b}{T_b}}\cos(2\pi f_c t), & \text{for symbol 0 corresponding to } i = 2 \end{cases}$$ --------(11)

Where with T_b denoting the bit duration E_b and denoting the transmitted signal energy per bit; see the waveform of Fig. 7.1 (c) for a representation example of BPSK. A pair of sinusoidal waves, s1(t)ands2(t) which differ only in a relative phase-shift of π radians as defined in Eq. (7.12), are referred to as antipodal signals.BPSK differs from BASK in an important
respect: the envelope of the modulated signal s(t) is maintained constant at the value for all time t. This property, which follows directly from Eq. (7.12), has two important consequences:

1. The transmitted energy per bit,E_b is constant; equivalently, the average transmitted power is constant.

2. Demodulation of BPSK cannot be performed using envelope detection; rather, we have to look to coherent detection as described next.

GENERATION AND COHERENT DETECTION OF BPSK SIGNALS

(i) Generation

To generate the BPSK signal, we build on the fact that the BPSK signal is a special case of DSB-SC modulation. Specifically, we use a product modulator consisting of two components

- **(i)** Non-return-to-zero level encoder, whereby the input binary data sequence is encoded in polar form with symbols 1 and 0 represented by the constant-amplitude levels: and , respectively.

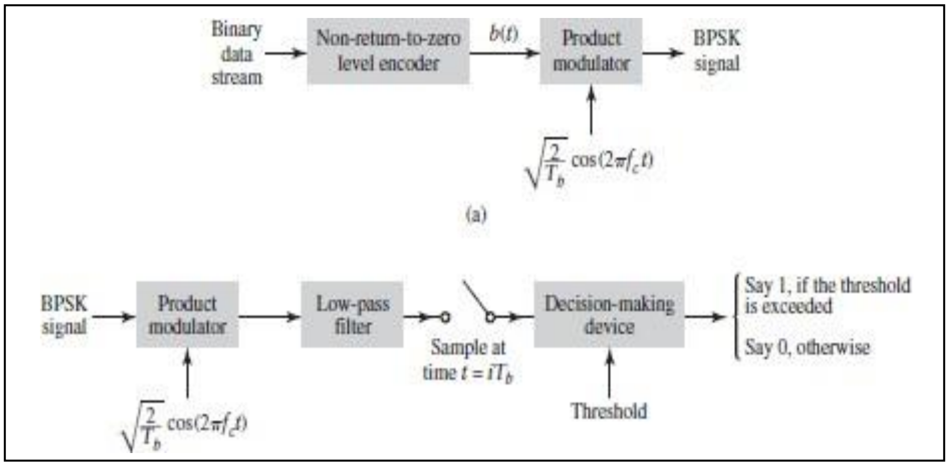


Fig 3.1 (a) BPSK modulator. (b) Coherent detector for BPSK; for the sampler, integer i _ 0, _1, _2, ...

(ii) Product modulator, which multiplies the level-encoded binary wave by the sinusoidal carrier c(t) of amplitude to produce the BPSK signal. The timing pulses used to generate the level-encoded binary wave and the sinusoidal carrier wave are usually, but not necessarily, extracted from a common master clock.

(ii) Detection

To detect the original binary sequence of 1s and 0s, the BPSK signal x(t) at the channel output is applied to a receiver that consists of four sections, as depicted in Fig. 3.1(b)

(i) Product modulator, which is also supplied with a locally generated reference signal that is a replica of the carrier wave c(t).

(ii) Low-pass filter, designed to remove the double-frequency components of the product modulator output and pass the zero-frequency components.

(iii) Sampler, which uniformly samples the output of the low-pass filter at $t=iT_b$ where ;i= the local clock governing the operation of the sampler is synchronized with the clock responsible for bit-timing in the transmitter.

(iv) Decision-making device, which compares the sampled value of the low-pass filter's output to an externally supplied threshold, every T_b seconds. If the threshold is exceeded, the device decides in favor of symbol 1; otherwise, it decides in favour of symbol 0.

The BPSK receiver described in Fig. 3.1 is said to be coherent in the sense that the sinusoidal reference signal applied to the product modulator in the demodulator is synchronous in phase (and, of course, frequency) with the carrier wave used in the modulator In addition to synchrony with respect to carrier phase, the receiver also has an accurate knowledge of the interval occupied by each binary symbol.The operation of the coherent BPSK receiver in Fig. 3.4(b) follows a procedure similar to that described for the demodulation of a double-sideband suppressed-carrier (DSBSC)modulated wave) with a couple of important additions:sampler and decision-making device. The rationale for this similarity builds on what we have already stated: BPSK is simply another form of DSB-SC modulation.However, an issue that needs particular attention is how to design the low-pass filter in Fig. 3.4(b)

SPECTRAL ANALYSIS OF BPSK

As with the experiment on BASK, consider a binary data stream that consists of a square wave, the amplitude of which alternates between every T_b seconds. The square wave is centered on the origin. The objectives of this second experiment are similar to those of
Computer Experiment I on BASK:

(i) To evaluate the effect of varying the carrier frequency on the power spectrum of the BPSK signal, for a fixed square modulating wave.

(ii) To evaluate the effect of varying modulation frequency on the power spectrum of the BPSK signal for a fixed carrier frequency.

BASK and BPSK signals occupy the same transmission bandwidth—namely,$2/T_b$ —which defines the width of the main lobe of the sinc-shaped power spectra. The BASK spectrum includes a carrier component, whereas this component is absent from the BPSK spectrum. With this observation we are merely restating the fact that BASK is an example of amplitude modulation, whereas BPSK is an example of double sideband-suppressed carrier modulation.

QUADRIPHASE-SHIFT KEYING

An important goal of digital communication is the efficient utilization of channel bandwidth.This goal is attained by a bandwidth-conserving modulation scheme known as quadriphase-shift keying.
In quadriphase-shift keying (QPSK), as with BPSK, information carried by the transmitted signal is contained in the phase of a sinusoidal carrier. In particular, the phase of the sinusoidal carrier takes on one of four equally spaced values, such as $\pi/4, 3\pi/4, 5\pi/4$, and $7\pi/4$. For this set of values, we define the transmitted signal as

$$s_i(t) = \begin{cases} \sqrt{\frac{2E}{T}} \cos\left[2\pi f_c t + (2i-1)\frac{\pi}{4}\right], & 0 \le t \le T \\ 0, & \text{elsewhere} \end{cases}$$ ---------(12)

Where i=1,2,3,4;E is the transmitted signal energy per symbol and T is the symbol duration. Each one of the four equally spaced phase values corresponds to a unique pair of bits called a dibit. For example, we may choose the foregoing set of phase values to represent the Gray encoded set of dibits: 10, 00, 01, and 11. In this form of encoding, we see that only a single bit is changed from one dibit to the next. Note that the symbol duration (i.e., the duration of each dibit) is twice the bit duration, as shown by

$$T = 2T_b$$ ---------(13)

Using a well-known trigonometric identity, we may recast the transmitted signal in the interval in the expanded form

$$s_i(t) = \sqrt{\frac{2E}{T}} \cos\left[(2i-1)\frac{\pi}{4}\right] \cos(2\pi f_c t) - \sqrt{\frac{2E}{T}} \sin\left[(2i-1)\frac{\pi}{4}\right] \sin(2\pi f_c t)$$

where i=1,2,3,4.Based on the expanded form of Eq. (14), we can make some important observations:

1. In reality, the QPSK signal consists of the sum of two BPSK signals.

2. One BPSK signal, represented by the first term

–

$$\sqrt{2E/T} \cos[(2i-1)\pi/4] \cos[(2\pi f_c t)],$$

defines the product of modulating a binary wave by the sinusoidal carrier π which has unit energy over the symbol duration T. We also recognize that

$$\sqrt{E} \cos\left[(2i-1)\frac{\pi}{4}\right] = \begin{cases} \sqrt{E/2} & \text{for } i = 1, 4 \\ -\sqrt{E/2} & \text{for } i = 2, 3 \end{cases}$$ -----------(15)

We therefore see that this binary wave has an amplitude equal to –

3. The other BPSK signal, represented by the second term

$$-\sqrt{2E/T} \sin\left[(2i-1)\frac{\pi}{4}\right] \sin(2\pi f_c t),$$

defines the product of modulating a different binary wave by the sinusoidal carrier π , which also has unit energy per symbol. This time, we recognize that

$$-\sqrt{E}\sin\left[(2i-1)\frac{\pi}{4}\right] = \begin{cases} -\sqrt{E/2} & \text{for } i = 1, 2 \\ \sqrt{E/2} & \text{for } i = 3, 4 \end{cases} \quad \text{----------(16)}$$

We therefore see that this second binary wave also has an amplitude equal to, albeit in a different way with respect to the index i.

5. The two binary waves defined in Eqs. (15) and (16) share a common value for the symbol duration—namely, T.

6. The two sinusoidal carrier waves identified under points 2 and 3 are in phase quadrature with respect to each other. Moreover, they both have unit energy per symbol duration. We may therefore state that these two carrier waves constitute an orthonormal pair of basis functions.

6. For each possible value of the index i, Eqs. (15) and (16) identity the corresponding dibit, as outlined in Table 1. This table also includes other related entries pertaining to the phase of the QPSK signal, and the amplitudes of the two binary waves identified under points 2 and 3.

Table:3.1 Relation between index i and identity of corresponding dibit and other related matters

Index i	Phase of QPSK signal (radians)	Amplitudes of constituent binary waves: Binary wave 1 $a_1(t)$	Binary wave 2 $a_2(t)$	Input dibit $0 < t < T$
1	$\pi/4$	$+\sqrt{E/2}$	$-\sqrt{E/2}$	10
2	$3\pi/4$	$-\sqrt{E/2}$	$-\sqrt{E/2}$	00
3	$5\pi/4$	$-\sqrt{E/2}$	$+\sqrt{E/2}$	01
4	$7\pi/4$	$+\sqrt{E/2}$	$+\sqrt{E/2}$	11

GENERATION AND COHERENT DETECTION OF QPSK SIGNALS

Generation and coherent detection of QPSK signals, as described here:

(i) Generation

To generate the QPSK signal, the incoming binary data stream is first converted into polar form by a non-return-to-zero level encoder; the encoder output is denoted by b(t).

Symbols 1 and 0 are thereby represented by and where E_b=E/2. The resulting binary wave is next divided by means of a demultiplexer (consisting of a serial- to-parallel converter) into two separate binary waves consisting of the odd- and evennumbered input bits of b(t).These two binary waves, referred to as the demultiplexed components of the input binary wave, are denoted by a1(t) and a2(t). In any signalling interval, the amplitudes of a1(t) and a2(t) are determined in accordance with columns 3 and 4 of Table3. 1, depending on the particular dibit that is being transmitted. The demultiplexed binary waves a1(t) and a2(t) are used to modulate the pair of quadrature carriers—namely, π/4, 3 π/4, 5 π/4 and 7π/4. Finally, the two BPSK signals are subtracted to produce the desired QPSK signals, as depicted in Fig 3.2 (a).

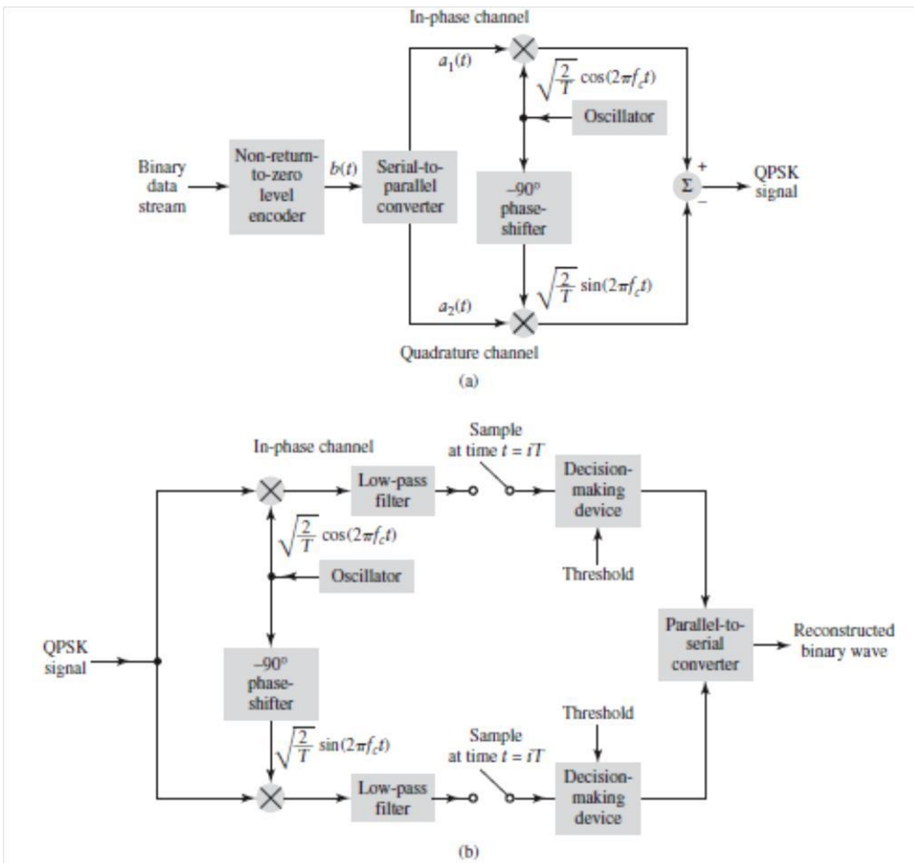


Fig 3.2 Block diagrams of (a) QPSK transmitter and (b) coherent QPSK receiver;

(ii) Detection

The QPSK receiver consists of an in-phase (I)-channel and quadrature (Q)-channel with a common input, as depicted in Fig.(b). Each channel is itself made up of a product modulator, low-pass filter, sampler, and decision-making device. Under ideal conditions,

For the two synchronous samplers, integer i = the I- and Q-channels of the receiver, respectively, recover the demultiplexed components a1 (t) and a2(t) responsible for modulating the orthogonal pair of carriers in the transmitter.

Accordingly, by applying the outputs of these two channels to a multiplexer (consisting of a parallel-to-serial converter), the receiver recovers the original binary sequence.The design of the QPSK receiver builds on the strategy described for the coherent BPSK receiver. Specifically, each of the two low-pass filters in the coherent QPSK receiver of Fig.3.2 (b) must be assigned a bandwidth equal to or greater than the reciprocal of the symbol duration T for satisfactory operation of the receiver.

OFFSET QUADRIPHASE-SHIFT KEYING

In QPSK, the carrier amplitude is maintained constant. However, the carrier phase may jump by 45°

for every two-bit (dibit) duration. This latter property can be of particular concern when the QPSK signal is filtered during the course of transmission over a communication channel. Unfortunately, such a filtering action can cause the carrier amplitude, and therefore the envelope of the QPSK signal, to fluctuate. When the data transmission system contains nonlinear components, fluctuations of this kind are undesirable as they tend to distort the received signal; the net result is a reduced opening of the eye diagram.

The extent of amplitude fluctuations exhibited by QPSK signals may be reduced by using a variant of quadriphase-shift keying known as the offset quadriphase-shift keying (OQPSK).1 In OQPSK, the demultiplexed binary wave labeled a2(t) in Fig.3.2(a) is delayed (i.e., offset) by one bit duration with respect to the other demultiplexed binary wave labelled a1(t) in that figure.

This modification has the effect of confining the likely occurrence of phase transitions to 0° and However, the phase transitions in OQPSK occur twice as frequently but with a reduced range of amplitude fluctuations, compared with QPSK. In addition to the phase transitions, there are also phase transitions in QPSK. We therefore find that amplitude fluctuations in OQPSK due to filtering have a smaller amplitude than in OQPSK.

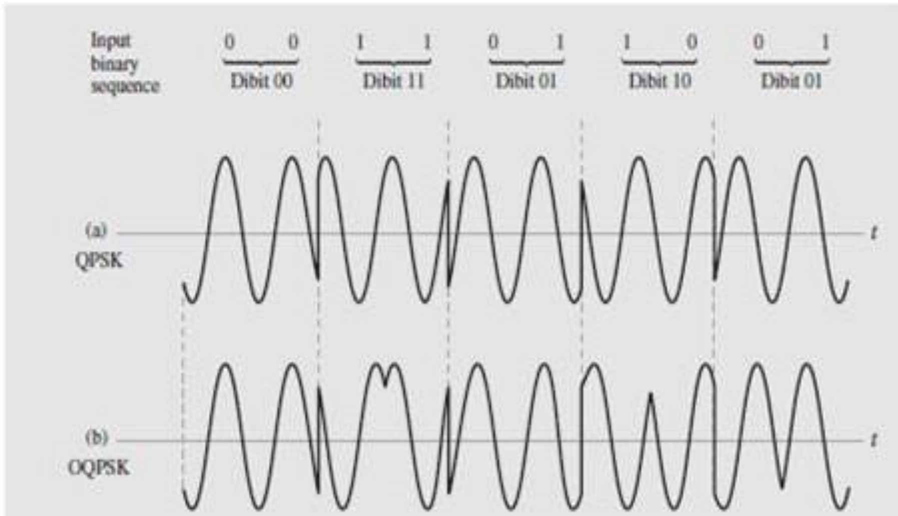


Figure 3.3 Graphical representation of phase transitions in QPSK and OQPSK

FREQUENCY-SHIFT KEYING

BINARY FREQUENCY-SHIFT KEYING (BFSK)

In the simplest form of frequency-shift keying known as binary frequency-shift keying (BFSK), symbols 0 and 1 are distinguished from each other by transmitting one of two sinusoidal waves that differ in frequency by a fixed amount. A typical pair of sinusoidal waves is described by

$$s_i(t) = \begin{cases} \sqrt{\dfrac{2E_b}{T_b}}\cos(2\pi f_1 t), & \text{for symbol 1 corresponding to } i = 1 \\[2ex] \sqrt{\dfrac{2E_b}{T_b}}\cos(2\pi f_2 t), & \text{for symbol 0 corresponding to } i = 2 \end{cases} \qquad \text{------------(17)}$$

Where E_b is the transmitted signal energy per bit. When the frequencies f1 and f2 are chosen in such a way that they differ from each other by an amount equal to the reciprocal of the bit duration T_b, the BFSK signal is referred to as Sunde's BFSK after its originator. This modulated signal is a continuous-phase signal in the sense that phase continuity is always maintained, including the inter-bit switching times. Waveform Figure plots the waveform of Sunde's BFSK produced by the input binary sequence 0011011001 for a bit duration Part (a) of the figure displays the waveform of the input sequence,

and part (b) displays the corresponding waveform of the BFSK signal. The latter part of the figure clearly displays the phase-continuous property of Sunde's BFSK.

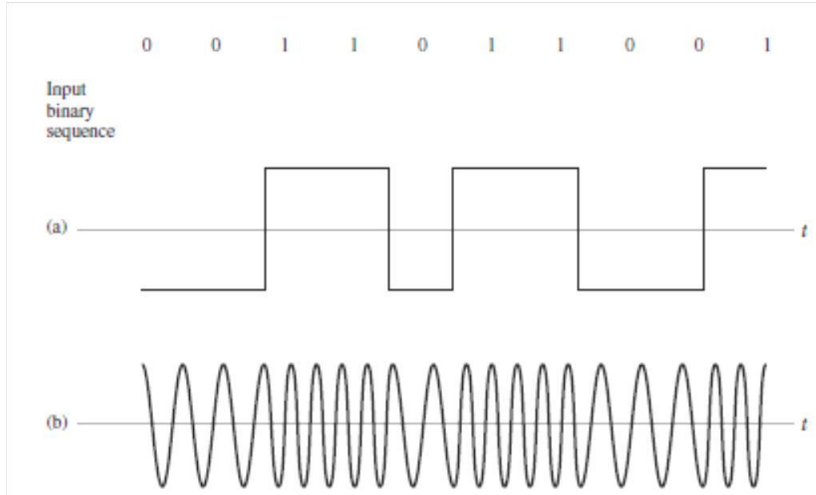


Fig. 3.4 (a) Binary sequence and its non-return-to-zero level-encoded waveform.(b) Sunde's BFSK signal.

CONTINUOUS-PHASE FREQUENCY-SHIFT KEYING

Sunde's BFSK is the simplest form of a family of digitally modulated signals known collectively as continuous-phase frequency-shift keying (CPFSK) signals, which exhibit the following distinctive property:

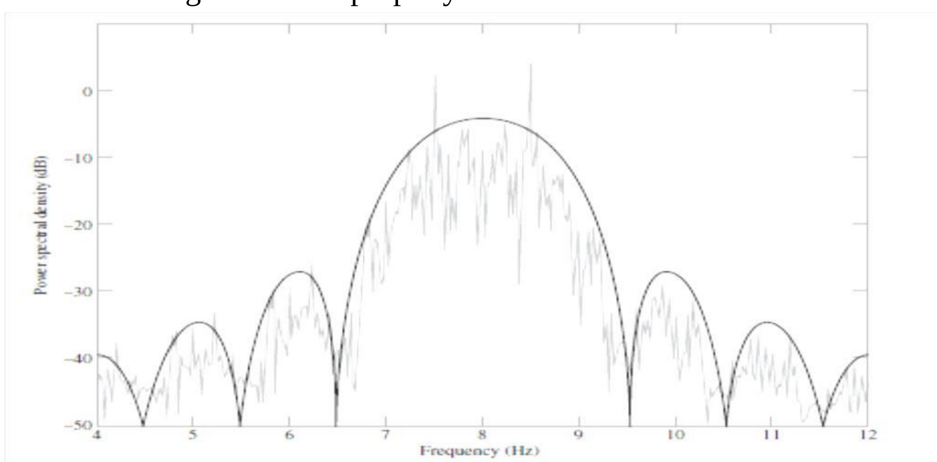


Fig.3.5 Power spectrum

Power spectrum of Sunde's BFSK produced by square wave as the modulating signal for the following parameters: fc = 8 Hz and Tb =1 s. In other words, the CPFSK signal is a continuous-wave modulated wave like any other angle-modulated wave experienced in the analog world, despite the fact that the modulating wave is itself discontinuous. In Sunde's BFSK, the overall excursion in the transmitted frequency from symbol 0 to symbol 1, or vice versa, is equal to the bit rate of the incoming data stream. In another special form of CPFSK known as minimum shift keying (MSK), the binary modulation process uses a different value for the frequency excursion f with the result that this new modulated wave offers superior spectral properties to Sunde's BFSK.

MINIMUM-SHIFT KEYING

In MSK, the overall frequency excursion f from binary symbol 1 to symbol 0, or vice versa, is one half the bit rate, as shown by

$$\delta f = f_1 - f_2 = \frac{1}{2T_b} \quad \text{------------(18)}$$

The unmodulated carrier frequency is the arithmetic mean of the two transmitted frequencies f1 And f2 that is,

$$f_c = \frac{1}{2}(f_1 + f_2) \quad \text{----------(19)}$$

Expressing f1 and f2 in terms of the carrier frequency f_c and overall frequency excursion we have

$$f_1 = f_c + \frac{\delta f}{2}, \quad \text{for symbol 1}$$

$$f_2 = f_c - \frac{\delta f}{2}, \quad \text{for symbol 0}$$

----------(20) and (21)

Accordingly, we formally define the MSK signal as the angle-modulated wave

$$s(t) = \sqrt{\frac{2E_b}{T_b}}\cos[2\pi f_c t + \theta(t)] \quad \text{---------(22)}$$

Where is the phase of the MSK signal. In particular, when frequency f1is transmitted, corresponding to symbol 1, we find from Eqs. (20) and (22) that the phase assumes the value

$$\theta(t) = 2\pi\left(\frac{\delta f}{2}\right)t$$
$$= \frac{\pi t}{2T_b}, \quad \text{for symbol 1}$$

In words, this means that at time $t=T_b$, the transmission of symbol 1 increases the phase of the MSK signal s(t) by π radians. By the same token, when frequency f2 is transmitted, corresponding
to symbol 0, we find from Eqs. (21) and (22) that the phase assumes the value

$$\theta(t) = 2\pi\left(-\frac{\delta f}{2}\right)t$$
$$= -\frac{\pi t}{2T_b}, \quad \text{for symbol 0}$$

-------------(24)

This means that at time $t=T_b$ the transmission of symbol 0 decreases the phase of s(t) by π/2 radians.

The phase changes described in Eqs. 23 and 24) for MSK are radically different from their corresponding counterparts for Sunde's BFSK. Specifically, the phase of Sunde's BFSK signal changes by -π radians at the termination of the interval representing symbol 0, and +π radians for symbol 1. However, the changes –π and +π are exactly the same, modulo 2π.This observation, in effect, means that Sunde's BFSK has no memory; in other words, knowing which particular change occurred in the previous bit interval provides no help in the current bit interval. In contrast, we have a completely different situation in the case of MSK by virtue of the different ways in which the transmissions of symbols 1 and 0 affect the phase as shown in Eqs. (23) and (24). Note also that the overall frequency excursion in MSK is the minimum frequency spacing between symbols 0 and 1 that allows their FSK representations to be coherently orthogonal, hence the terminology "minimum-shift keying."

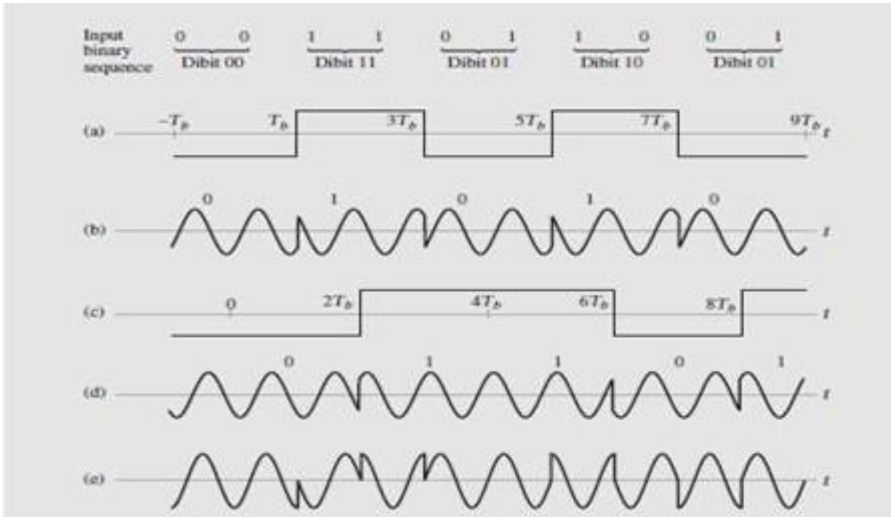


Figure 3.6 OQPSK signal components: (a) modulating signal for in phase component (b) modulated waveform of in phase component (c) modulating signal for quadrature component (d)

modulated waveform of quadrature component (e) waveform of OQPSK signal obtained by subtracting d from b.

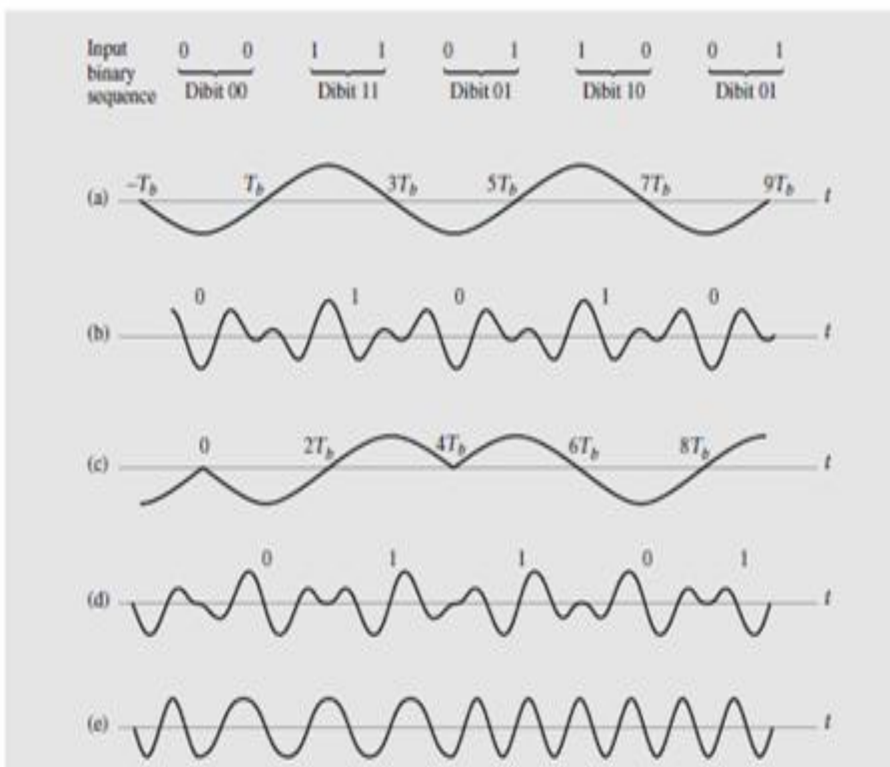


Figure 3.7 MSK signal components (a) modulating signal for in phase component (b) modulated waveform of in phase component (c) modulating signal for quadrature component (d) modulated waveform of quadrature component (e) waveform of MSK signal obtained by subtracting d from b.

FORMULATION OF MINIMUM-SHIFT KEYING

To proceed with the formulation, we refer back to Eq. (22), and use a well-known trigonometric identity to expand the angle-modulated wave (i.e., MSK signal) s(t) as

$$s(t) = \sqrt{\frac{2E_b}{T_b}}\cos(\theta(t))\cos(2\pi f_c t) - \sqrt{\frac{2E_b}{T_b}}\sin(\theta(t))\sin(2\pi f_c t)$$ ---------(25)

In light of this equation, we make two identifications:

(i) $S_I(t)$= is the in-phase (I) component associated with the carrier π----------------------- (26)

(ii) $S_Q(t)$= is the quadrature (Q) component associated with the 90°- Phase shifted carrier--------------(27)

To highlight the bearing of the incoming binary data stream on the composition of s1(t) and s2(t), we reformulate them respectively as follows:

$$s_I(t) = a_1(t)\cos(2\pi f_0 t)$$ -----------(28) and

$$s_Q(t) = a_2(t)\sin(2\pi f_0 t)$$ ---------(29)

The a1(t) and a2(t) are two binary waves that are extracted from the incoming binary data stream through demultiplexing and offsetting, in a manner similar to OQPSK. As such, they take on the value +1 or -1 in symbol (i.e., dibit) intervals of duration $T=2T_b$, where T_b is the bit duration of the incoming binary data stream. The two data signals a1(t) and a2(t) are respectively weighted by the sinusoidal functions cos(2π and sin(2πf0t), where the frequency f0 is to be determined. To define f0,we use Eqs. (28) and (29) to reconstruct the original anglemodulated wave s(t) in terms of the data signals a1(t) and a2(t) ,In so doing, we obtain

$$\theta(t) = -\tan^{-1}\left[\frac{s_Q(t)}{s_I(t)}\right]$$
$$= -\tan^{-1}\left[\frac{a_2(t)}{a_1(t)}\tan(2\pi f_0 t)\right]$$ ------------(30)

on the basis of which we recognize two possible scenarios that can arise:

1. a1(t)=a2(t),This scenario arises when two successive binary symbols (constituting a dibit) in the incoming data stream are the same (i.e., both are 0s or 1s); hence, Eq.
(30) reduces to

$$\theta(t) = -\tan^{-1}[\tan(2\pi f_0 t)]$$
$$= -2\pi f_0 t$$ ----------(31)

2. a2(t)=-a1(t),This second scenario arises when two successive binary symbols (i.e., dibits) in the incoming data stream are different; hence, Eq. (7.31) reduces to

$$\theta(t) = -\tan^{-1}[-\tan(2\pi f_0 t)]$$
$$= 2\pi f_0 t$$ ----------------------(32)

Equations (31) and (32) are respectively of similar mathematical forms as Eqs. (24) and (23). Accordingly, we may now formally define

$$f_0 = \frac{1}{4T_b}$$ -----------------(33)

To sum up, given a non-return-to-zero level encoded binary wave b(t) of prescribed bit T_b duration and a sinusoidal carrier wave of frequency f_c, we may formulate the MSK signal by proceeding as follows:

1. Use the given binary wave b(t) to construct the binary demultiplexed-offset waves a1(t) and a2(t)

2. Use Eq. (33) to determine the frequency f0.

3. Use Eqs. (28) and (29) to determine the in-phase component s1(t) and quadrature component $s_{Q(t)}$ respectively from which the MSK signal s(t) follows.

NONCOHERENT DIGITAL MODULATION SCHEMES

Coherent receivers, exemplified by the schemes shown in Figs. 3.4(b) and 3.5(b), require knowledge of the carrier wave's phase reference to establish synchronism with their respective transmitters. However, in some communication environments, it is either impractical or too expensive to phase-synchronize a receiver to its transmitter. In situations of this kind, we resort to the use of noncoherent detection by abandoning the use of phase synchronization between the receiver and its transmitter, knowing that when we do so the receiver performance is degraded in the presence of channel noise..

NONCOHERENT DETECTION OF BASK SIGNAL

The generation of BASK signals involves the use of a single sinusoidal carrier of frequency f_c for symbol 1 and switching off the transmission for symbol 0. Now, the system designer would have knowledge of two system parameters:

- The carrier frequency f_c
- The transmission bandwidth, which is determined by the bit duration T_b

It is therefore natural to make use of these known parameters in designing the noncoherent receiver for BASK. Specifically, the receiver consists of a band-pass filter, followed by an envelope detector, then a sampler, and finally a decision-making device, as depicted in Fig. 3.6. The band-pass filter is designed to have a mid-band frequency equal to the carrier frequency f_c and a bandwidth equal to the transmission bandwidth of the BASK signal. Moreover,it is assumed that the intersymbol interference (ISI) produced by the filter is negligible,which, in turn, requires that the rise time and decay time of the response of the filter to a rectangular pulse be short compared to the bit duration Tb. Under these conditions, we find that in response to the incoming BASK signal (assumed to be noise-free), the band-pass filter produces a pulsed sinusoid for symbol 1 and, ideally, no output for symbol 0. Next, the envelope detector traces the envelope of the filtered version of the BASK signal.

Finally,the decision-making device working in conjunction with the sampler, regenerates the original binary data stream by comparing the sampled envelope-detector output against a preset

threshold every Tb seconds; this operation assumes the availability of bit-timing in the receiver. If the threshold is exceeded at time $t=iT_{b,i=0,}$ the receiver decides in favor of symbol 1; otherwise, it decides in favor of symbol 0. In the absence of channel noise and channel distortion, the receiver output (on a bit-by-bit basis) would be an exact replica of the original binary data stream applied to the transmitter, subject to the abovementioned assumptions on the band-pass filter.

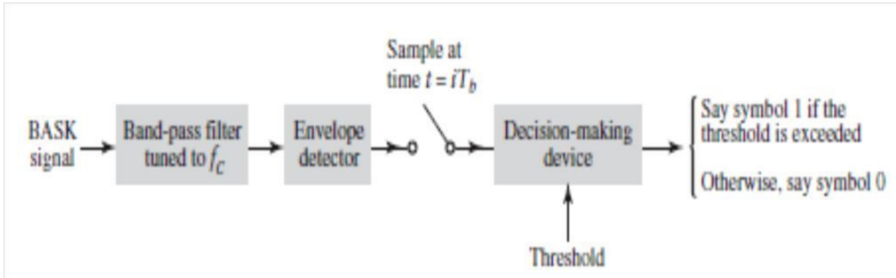


Figure **3.8** Noncoherent BASK receiver; the integer i for the sampler equals 0,

NONCOHERENT DETECTION OF BFSK SIGNALS

In the case of BFSK, the transmissions of symbols 1 and 0 are represented by two carrier waves of frequencies f1 and f2 respectively, with adequate spacing between them. In light of this characterization, we may build on the noncoherent detection of BASK by formulating the noncoherent BFSK receiver of Fig. 3.5. The receiver consists of two paths, one dealing with frequency f1 (i.e., symbol 1) and the other dealing with frequency f2 (i.e., symbol 0)

- Path 1 uses a band-pass filter of mid-band frequency f1. The filtered version of the incoming BFSK signal is envelope-detected and then sampled at time $t=iT_b, i=0, ,$to produce the output v1.
- Path 2 uses a band-pass filter of mid-band frequencyf2. As with path 2, the filtered version of the BFSK signal is envelope-detected and then sampled at time $t=iT_{b,i=0, ,}$to produce the outputv2.

The two band-pass filters have the same bandwidth, equal to the transmission bandwidth of the BFSK signal. Moreover, as pointed out in dealing with BASK, the intersymbol interference produced by the filters is assumed to be negligible.The outputs of the two paths, v1 and v2 are applied to a comparator, where decisions on the composition of the BFSK signal are repeated every T_b seconds. Here again, the availability of bit timing is assumed in the receiver. Recognizing that the upper path corresponds to symbol 1 and the lower path corresponds to symbol 0, the comparator decides in favour of symbol 1 if v1 is greater than v2 at the specified bit-timing instant; otherwise, the decision is made in favor of symbol 0. In a noise-free environment and no channel distortion, the receiver output (on a bit-by-bit basis) would again be a replica of the original binary data stream applied to the transmitter input.

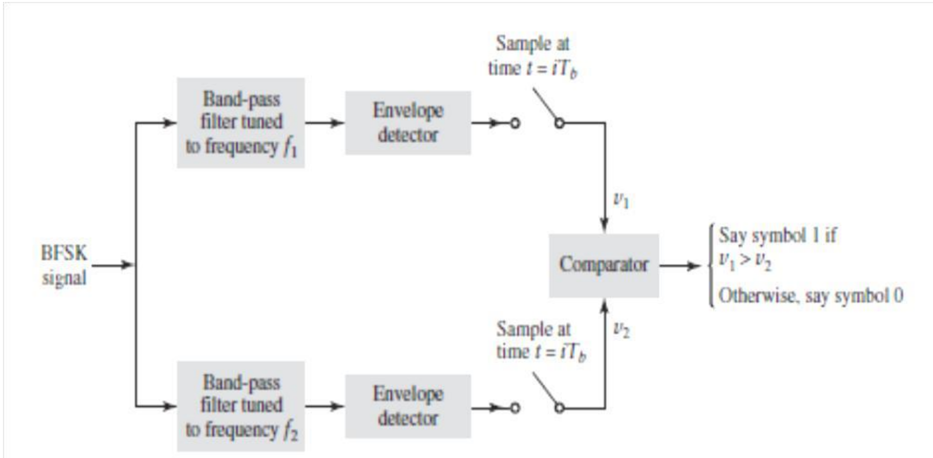


Figure 3.9 Noncoherent BFSK receiver; the two samplers operate synchronously, with i=0, ,

DIFFERENTIAL PHASE-SHIFT KEYING

From the above discussion, we see that both amplitude-shift keying and frequency-shift keying lend themselves naturally to noncoherent detection whenever it is impractical to maintain carrier-phase synchronization of the receiver to the transmitter. But in the case of phase-shift keying, we cannot have noncoherent detection in the traditional sense because the term "noncoherent" means having to do without carrier-phase information. To get around this difficulty, we employ a "pseudo PSK" technique known as differential phase-shift keying (DPSK), which, in a loose sense, does permit the use of noncoherent detection.

DPSK eliminates the need for a coherent reference signal at the receiver by combining two basic operations at the transmitter:

- Differential encoding of the input binary wave
- Phase-shift keying

It is because of this combination that we speak of "differential phase-shift keying." In effect, to send symbol 0, we phase advance the current signal waveform by 180 degrees, and to send symbol 1 we leave the phase of the current signal waveform unchanged. Correspondingly, the receiver is equipped with a storage capability (i.e., memory) designed to measure the relative phase difference between the waveforms received during two successive bit intervals. Provided the unknown phase varies slowly (i.e., slow enough for it to be considered essentially constant over two bit intervals), the phase difference between waveforms received in two successive bit intervals will be essentially independent of θ.

GENERATION AND DETECTION OF DPSK SIGNALS

(i) Generation

The differential encoding process at the transmitter input starts with an arbitrary first bit, serving merely as reference. Let $\{d_k\}$ denote the differentially encoded sequence with this added reference bit. To generate this sequence, the transmitter performs the following two operations:

- If the incoming binary symbol $\{b_k\}$ is 1, then the symbol $\{d_k\}$ is unchanged with respect to the previous symbol d_{k-1}.
- If the incoming binary symbol $\{b_k\}$ is 0, then the symbol $\{d_k\}$ is changed with respect to the previous symbol d_{k-1}.

The differentially encoded sequence $\{d_k\}$, thus generated is used to phase-shift a sinusoidal carrier wave with phase angles 0 and π radians, representing symbols 1 and 0, respectively.The

block diagram of the DPSK transmitter is shown in Fig. 3.5(a). It consists, in part, of a logic network and a one-bit delay element (acting as the memory unit) interconnected so as to convert the raw binary sequence $\{b_k\}$ into a differentially encoded sequence $\{d_k\}$. This sequence is amplitude-level encoded and then used to modulate a carrier wave of frequency f_c thereby producing the desired DPSK signal.

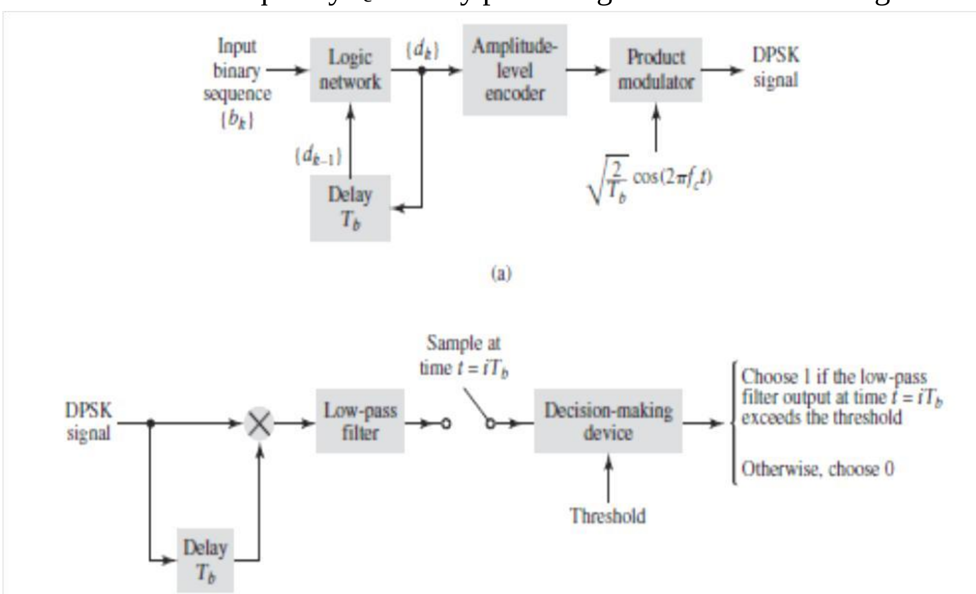


figure 3.11 Block diagrams for (a) DPSK transmitter and (b) DPSK receiver; for the sampler, integer i=0,

(ii) Detection

For the detection of DPSK signals, we take advantage of the fact that the phase-modulated pulses pertaining to two successive bits are identical except for a possible sign reversal. Hence, the incoming pulse is multiplied by the preceding pulse, which, in effect, means that the preceding pulse serves the purpose of a locally generated reference signal. On this basis, we may formulate the receiver of Fig. 3.8(b) for the detection of DPSK signals. Comparing the DPSK detector of Fig. 3.8(b) and the coherent BPSK detector of Fig. 3.3(b), we see that the two receiver structures are similar except for the source of the locally generated reference signal. According to Fig. 3.8(b), the DPSK signal is detectable, given knowledge of the reference bit, which, as mentioned previously, is inserted at the very beginning of the incoming binary data stream. In particular, applying the sampled output of the low-pass filter to a decision-making device supplied with a prescribed threshold, detection of the DPSK signal is accomplished. If the threshold is exceeded, the receiver decides in favor of symbol 1; otherwise, the decision is made in favor of symbol 0. Here again, it is assumed that the receiver is supplied with bit-timing information for the sampler to work properly.

M-ARY DIGITAL MODULATION SCHEMES

By definition, in an M-ary digital modulation scheme, we send any one of M possible signals s1(t),s2(t),......sM(t), during each signaling (symbol) interval of duration T. In almost all applications,$M=2^m$, where m is an integer. Under this condition, the symbol duration,$T=MT_b$, where T_b is the bit duration.M-ary modulation schemes are preferred over binary modulation schemes for transmitting digital data over band-pass channels when the requirement is to conserve bandwidth at the expense of both increased power and increased system complexity.

In practice, we rarely find a communication channel that has the exact bandwidth required for transmitting the output of an information-bearing source by means of binary modulation schemes. Thus, when the bandwidth of the channel is less than the required value, we resort to an M-ary modulation scheme for maximum bandwidth conservation.

M-ARY PHASE-SHIFT KEYING

To illustrate the capability of M-ary modulation schemes for bandwidth conservation, consider first the transmission of information consisting of a binary sequence with bit duration T_b. If we were to transmit this information by means of binary PSK, for example, we would require a channel bandwidth that is inversely proportional to the bit duration T_b .However, if we take blocks of m bits to produce a symbol and use an M-ary PSK scheme $M=2^m$ with and symbol duration $T=mT_b$, then the bandwidth required is proportional to $1/(mT_b)$. This simple argument shows that the use of M-ary PSK provides a reduction in transmission bandwidth by a factor $m=\log_2 M$ over binary PSK.In M-ary PSK, the available phase of 2π radians is apportioned equally and in a discrete way among the M transmitted signals, as shown by the phase- modulated signal

$$s_i(t) = \sqrt{\frac{2E}{T}}\cos\left(2\pi f_c t + \frac{2\pi}{M}i\right), \quad i = 0, 1, \ldots, M-1, \quad 0 \le t \le T \qquad \text{-----------(34)}$$

where E is the signal energy per symbol, and is the carrier frequency f_c. Using a well-known trigonometric identity, we may expand Eq. (34) as

$$s_i(t) = \left[\sqrt{E}\cos\left(\frac{2\pi}{M}i\right)\right]\left[\sqrt{\frac{2}{T}}\cos(2\pi f_c t)\right] - \left[\sqrt{E}\sin\left(\frac{2\pi}{M}i\right)\right]\left[\sqrt{\frac{2}{T}}\sin(2\pi f_c t)\right], \quad i = 0, 1, \ldots, M-1, \quad 0 \le t \le T$$

------------------(35)

The discrete coefficients π and - π are respectively referred to as the in- phase and quadrature components of the M-ary PSK signal $S_i(t)$, We now recognize that

$$\left\{\left[\sqrt{E}\cos\left(\frac{2\pi}{M}i\right)\right]^2 + \left[\sqrt{E}\sin\left(\frac{2\pi}{M}i\right)\right]^2\right\}^{1/2} = \sqrt{E}, \quad \text{for all } i$$

-----------------(36)

Accordingly, M-ary PSK modulation has the unique property that the in-phase and quadrature components of the modulated signal Si(t)are interrelated in such a way that the discrete envelope of the signal is constrained to remain constant at the value for all M.

SIGNAL-SPACE DIAGRAM

The result described in Eq. (36 combined with the fact that the in-phase and quadrature components of M-ary PSK are discrete, leads to an insightful geometric portrayal of M-ary PSK. To explain, suppose we construct a two-dimensional diagram with the horizontal and vertical axes respectively defined by the following pair of orthnormal functions:

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

(37)

And

$$\phi_2(t) = \sqrt{\frac{2}{T}}\sin(2\pi f_c t), \quad 0 \le t \le T$$

--------------(38)

where the band-pass assumption implies orthogonality; the scaling factor assures unit energy over the interval T for both and ϕ2(t).On this basis, we may represent the in-phase component —π and quadrature component - π— for i=0,1,. .... M-1, as a set of points in this two-dimensional diagram, as illustrated in Fig. 7.19 for M = 8. Such a diagram is referred to as a signal-space diagram.

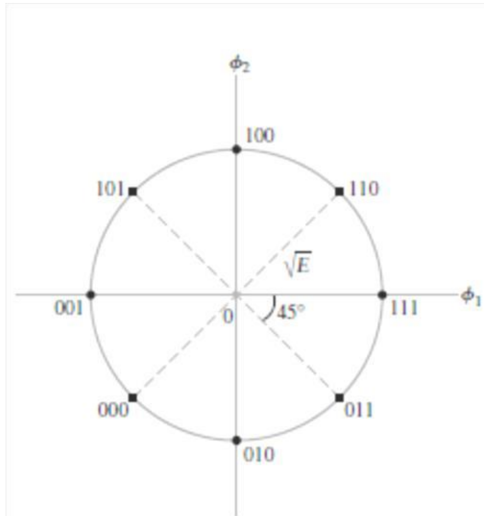


Fig 3.12 Signal-space diagram of 8-PSK.

Figure 3.9 leads us to make three important observations:

1. M-ary PSK is described in geometric terms by a constellation of M signal points distributed uniformly on a circle of radius

2. Each signal point in the figure corresponds to the signal Si(t)of Eq. (34) for a particular value of the index i.

3. The squared length from the origin to each signal point is equal to the signal energy E.In light of these observations, we may now formally state that the signal-space-diagram of Fig. 3.6 completely sums up the geometric description of M-ary PSK in an insightful manner. Note that the 3-bit sequences corresponding to the 8 signal points are Gray-encoded, with only a single bit changing as we move along the constellation in the figure from one signal point to an adjacent one.

M-A RY QUADRATURE AMPLITUDE MODULATION

Suppose next that the constraint of Eq. (36) that characterizes M-ary PSK modulation is removed. Then, the in-phase and quadrature components of the resulting M-ary modulated signal are permitted to be independent of each other. ... lated signal assumes the

$$s_i(t) = \sqrt{\frac{2E_0}{T}}a_i \cos(2\pi f_c t) - \sqrt{\frac{2E_0}{T}}b_i \sin(2\pi f_c t), \quad i = 0, 1, \ldots, M-1, \quad 0 \le t \le T$$

--------(39)

where the level parameter in the in-phase component and the level parameter a_i in the quadrature component b_i are independent of each other for all i. This new modulation scheme is called M-ary quadrature amplitude modulation (QAM). Note also that the constant E0 is the energy of the signal pertaining to a particular value of the index i for which the amplitude of the modulated signal is the lowest. M-ary QAM is a hybrid form of M-ary modulation, in the sense that it combines amplitude-shift keying and phase-shift keying. It includes two special cases:

(i) If b_i=0, for all i, the modulated signal Si(t) of Eq. (39) reduces to

$$s_i(t) = \sqrt{\frac{2E_0}{T}} a_i \cos(2\pi f_c t) \qquad i = 0, 1, \ldots, M-1$$

which defines M-ary amplitude-shift keying (M-ary ASK).

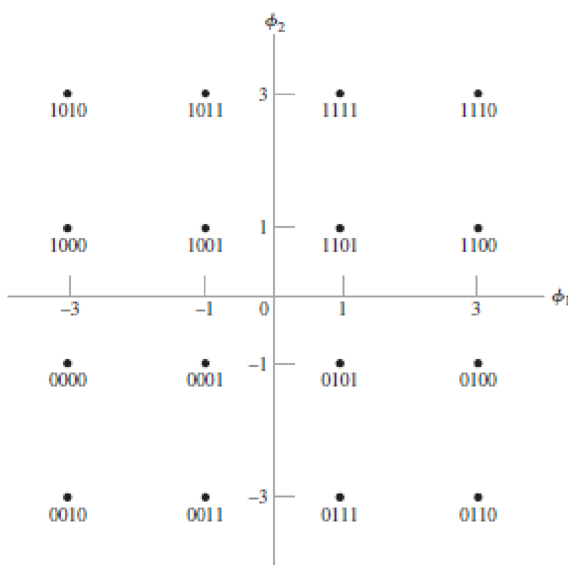
(ii) If E0=E and the constraint

for all i $(Ea_i^2 + Eb_i^2)^{1/2} = \sqrt{E},$

is satisfied, then the modulated signal $S_i(t)$ of Eq. (39) reduces to M-ary PSK.

SIGNAL-SPACE DIAGRAM

Figure 3.10 portrays the signal-space representation of M-ary QAM for M=16, with each signal point being defined by a pair of level parameters a_i and b_i where i=1,2,3,4. This time, we see that the signal points are distributed uniformly on a rectangular grid. The rectangular property of the signal-space diagram is testimony to the fact that the in-phase and quadrature components of M-ary QAM are independent of each other. Moreover, we see from Fig. 7.21 that, unlike M- ary PSK, the different signal points of M-ary QAM are characterized by different energy levels, and so they should be. Note also that each signal point in the constellation corresponds to a specific quadbit, which is made up of 4 bits. Assuming the use of Gray encoding, only one bit is changed as we go from each signal point in the



constellation horizontally or vertically to an adjacent point, as illustrated in Fig. 3.7

Figure 3.10 Signal-space diagram of Gray encoded M-ary QAM for M= 16.

M-A RY FREQUENCY-SHIFT KEYING

However, when we consider the M-ary version of frequency-shift keying, the picture is quite different from that described for M-ary PSK or M-ary QAM. Specifically, in one form of M-ary FSK, the transmitted signals are defined for some fixed integer n as follows:

$$s_i(t) = \sqrt{\frac{2E}{T}}\cos\left[\frac{\pi}{T}(n+i)t\right], \quad \begin{matrix} i = 0, 1, \ldots, M-1 \\ 0 \le t \le T \end{matrix}$$ -------------------(40)

The M transmitted signals are all of equal duration T and equal energy E. With the individual signal frequencies separated from each other by 1/2T hertz, the signals in Eq.(40) are orthogonal; that is, they satisfy the condition

$$\int_0^T s_i(t)s_j(t)\,dt = \begin{cases} E \text{ for } i = j \\ 0 \text{ for } i \ne j \end{cases}$$ ---------------(41)

Like M-ary PSK, the envelope of M-ary FSK is constant for all M, which follows directly from Eq. (40). Hence, both of these M-ary modulation strategies can be used over nonlinear channels.

On the other hand, M-ary QAM can only be used over linear channels because its discrete envelope varies with the index i (i.e., the particular signal point chosen for transmission).

SIGNAL-SPACE DIAGRAM

To develop a geometric representation of M-ary FSK, we start with Eq. (40). In terms of the signals Si(t)defined therein, we introduce a complete set of orthonormal functions:

$$\phi_i(t) = \frac{1}{\sqrt{E}}s_i(t) \quad \begin{matrix} i = 0, 1, \ldots, M-1 \\ 0 \le t \le T \end{matrix}$$ ---------------(42)

Unlike M-ary PSK and M-ary QAM, we now find that M-ary FSK is described by an M- dimensional signal-space diagram, where the number of signal points is equal to the number of coordinates. The visualization of such a diagram is difficult beyond

M=3.

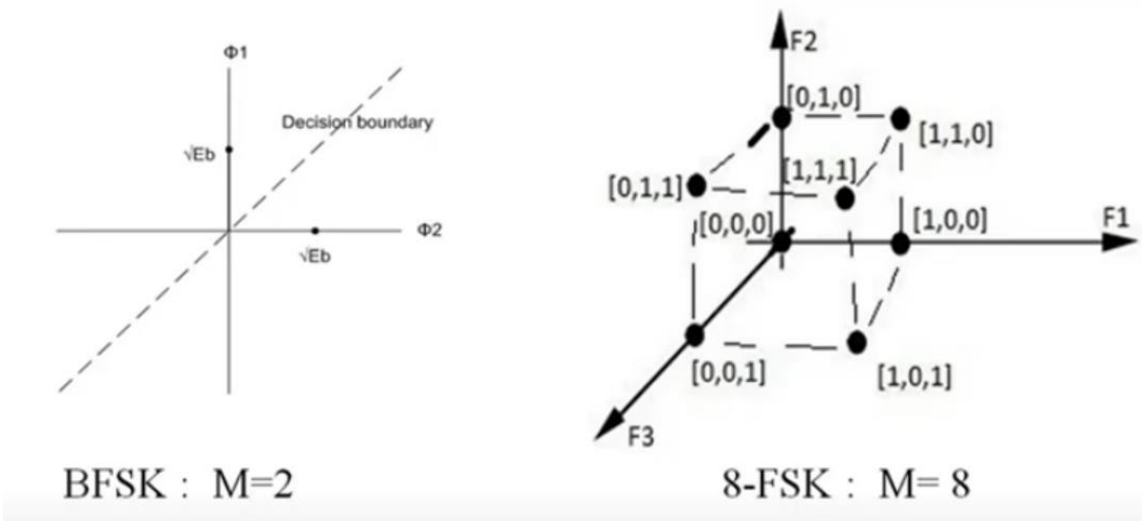


Figure 3.11 Signal-space diagram of BFSK and M-ary FSK

QUESTION BANK

UNIT IV

PART-A

1. Compare coherent and non-coherent detection.
2. A BFSK system employs two signalling frequencies f_1 and f_2. The lower frequency f_1 is 1200 Hz and signaling rate is 500 Baud. Compute f_2.
3. Outline the concept of M-ary modulation systems.
4. Formulate the probability of error for ASK, PSK and FSK modulation schemes.
5. Plot the binary ASK waveform for the bit sequence"1011011".
6. Plot the binary FSK waveform for the bit sequence"1010110".
7. What would be the advantage of the BFSK system over the BASK system, given that they operate on the same binary data stream and the same communication channel? Justify your answer.
8. Plot the binary PSK waveform for the bit sequence"1011001".
9. Consider a binary bit stream, with bit duration Tb=1ms, is modulated by a sinusoidal carrier sin2л4000t. Compute the bandwidth of the modulated signal for BPSK and QPSK.
10. Mention the merits and demerits of DPSK.
11. Compare QPSK and offset QPSK in-terms of phase changes that occurs in the modulated signal and power spectrum.
12. Draw the signal space diagram of 8-PSK.
13. QAM is a combination of which two types of modulation?
14. In 64-PSK, how many bits are transmitted per symbol?

PART-B

1. a) Explain the generation and non-coherent detection of binary amplitude shift keying (BASK) signal and derive the equation for probability of error in the presence of AWGN channel noise. (12marks)

a) A binary sequence 11100101 is applied to a BASK modulator. The bit duration is 1μs and the sinusoidal carrier wave used to represent symbol 1 has a frequency equal to 7MHz. Plot the time domain and frequency domain representation of the transmitted ASK signal. (4 marks)

2 a) Illustrate the generation and non-coherent detection of binary frequency shift keying (BFSK) signal and derive the equation for probability of error in the presence of AWGN channel noise. (12marks)

b) The binary sequence 10110101 is applied to a BFSK modulator. The bit duration is 2μs. The carrier frequencies used to represent symbols 0 and 1 are 2.5 MHz and 3.5 MHz, respectively. (a) Calculate the transmission bandwidth of the BFSK signal. (b) Plot the waveform of the BFSK signal. (4 marks)

3. a) Explain the generation and coherent detection of binary phase shift keying (BPSK) signal and derive the equation for probability of error in the presence of AWGN channel noise. (12marks)

3.b The binary sequence 10100110 is applied to a BPSK system. The bit duration is 10μs. The carrier frequency is 600KHz. (a) Calculate the transmission bandwidth of the system. (b) Plot the waveform of the transmitted signal. (4 marks)

4. a) Elucidate the generation and detection of quadrature phase shift keying (QPSK) signal and derive the expression for probability of error in the presence of AWGN channel noise. (12marks)

4. b) The binary sequence 11000110 with bit duration 1μs is applied to a QPSK modulator. The carrier frequency is 4MHz. Plot the waveform of the QPSK signal and calculate its transmission bandwidth. (4 marks)

5. Discuss the generalized structure of an M-ary FSK transmitter and receiver. Also develop the expression for transmission bandwidth and probability of error. (16 marks)

6 a) Elucidate generation, detection and bandwidth requirements of M-ary PSK modulation. (12 marks)

6.b) Mention the factors that should be considered in choosing the value of M in an M-ary system? (4 marks)

UNIT IV- ERROR CONTROL CODING-SECA1403

Discrete Memoryless channels- Linear Block codes – Hamming codes – Cyclic codes – Convolutional codes –Maximum Likelihood convolutional codes- Viterbi Decoder-Trellis Coded modulation- Turbo codes.

Introduction

Coding theory is concerned with the transmission of data across noisy channels and the recovery of corrupted messages. It has found widespread applications in electrical engineering, digital communication, mathematics and computer science. The transmission of the data over the channel depends upon two parameters. They are transmitted power and channel bandwidth. The power spectraldensity of channel noise and these two parameters determine signal to noise power ratio.

The signal to noise power ratio determine the probability of error of the modulation scheme. Errors are introduced in the data when it passes through the channel. The channel noise interferes the signal. The signal power is reduced. For the given signal to noise ratio, theerror probability can be reduced further by using coding techniques. The coding techniques also reduce signal to noise power ratio for fixed probability of error.

Principle of block coding

For the block of k message bits, (n-k) parity bits or check bits are added. Hence the total bits at the output of channel encoder are ‘n’. Such codes are called (n,k)block codes.Figure illustrates this concept.

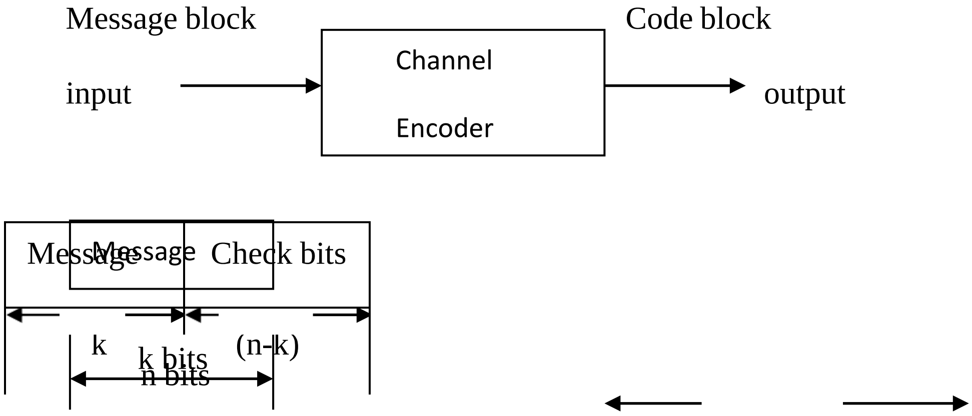


Figure: Functional block diagram of block coder

Types are

Systematic codes:

In the systematic block code, the message bits appear at the beginning of the code word. The message appears first and then check bits are transmitted in a block. This type of code is called systematic code.

Nonsystematic codes:

In the nonsystematic block code it is not possible to identify the message bits and check bits. They are mixed in the block.

Consider the binary codes and all the transmitted digits are binary.

Linear Block Codes

A code is linear if the sum of any two code vectors produces another code vector. This shows that any code vector can be expressed as a linear combination of other code vectors. Consider that the particular code vector consists of $m_1, m_2, m_3, \ldots m_k$ message bits and $c_1, c_2, c_3 \ldots c_q$ check bits. Then this code vector can be written as,

$X=(m_1, m_2, m_3, \ldots m_k c_1, c_2, c_3$

$\ldots c_q)$Here $q=n-k$

Where$q$ are the number of redundant bits added by the encoder.

Code vector can also be written as

$X=(M/C)$

Where M= k-bit message vector

C= q-bit check vector

The main aim of linear block code is to generate check bits and this check bits are mainly used for error detection and correction.

Example :

The (7, 4) linear code has the following matrix as a generator matrix

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \mathbf{g}_2 \\ \mathbf{g}_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

If u = (1 1 0 1) is the message to be encoded, its corresponding code word would be

$$\begin{aligned}\mathbf{v} &= 1\cdot\mathbf{g}_0 + 1\cdot\mathbf{g}_1 + 0\cdot\mathbf{g}_2 + 1\cdot\mathbf{g}_3 \\ &= (1\,1\,0\,1\,0\,0\,0) + (0\,1\,1\,0\,1\,0\,0) + (1\,0\,1\,0\,0\,0\,1) \\ &= (0\,0\,0\,1\,1\,0\,1)\end{aligned}$$

A linear systematic (n, k) code is completely specified by ak × n matrix G of the following form

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \mathbf{g}_2 \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{g}_{k-1} \end{bmatrix} = \left[\begin{array}{ccccc|ccccccc} p_{00} & p_{01} & \cdot & \cdot & p_{0,n-k-1} & 1 & 0 & 0 & . & . & . & 0 \\ p_{10} & p_{11} & \cdot & \cdot & p_{1,n-k-1} & 0 & 1 & 0 & . & . & . & 0 \\ p_{20} & p_{21} & \cdot & \cdot & p_{2,n-k-1} & 0 & 0 & 1 & . & . & . & 0 \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ p_{k-1,0} & p_{k-1,1} & \cdot & \cdot & p_{k-1,n-k-1} & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right]$$

(P matrix | k × k identity matrix)

where $p_{ij} = 0$ or 1

Let u = (u_0, u_1, … , u_{k-1}) be the message to be encoded.The corresponding code word is

$$v = (v_0, v_1, v_2, \ldots, v_{n-1})$$
$$= (u_0, u_1, \ldots, u_{k-1}) \cdot G$$

The components of **v** are

$$v_{n-k+i} = u_i \qquad \text{for } 0 \le i < k$$

$$v_j = u_0 p_{0j} + u_1 p_{1j} + \cdots + u_{k-1} p_{k-1,j} \quad \text{for } 0 \le j < n-k$$

The n – k equations given by above equation are called parity-check equations of the code

Example for Codeword

The matrix **G** given by

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \mathbf{g}_2 \\ \mathbf{g}_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Let **u** = (u_0, u_1, u_2, u_3) be the message to be encoded and **v** = (v_0, v_1, v_2, v_3, v_4, v_5,v_6) bethe corresponding code word

Solution :

$$\mathbf{v} = \mathbf{u} \cdot \mathbf{G} = (u_0, u_1, u_2, u_3) \cdot \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

By matrix multiplication, the digits of the code word v can be determined.

$$\begin{aligned} v_6 &= u_3 \\ v_5 &= u_2 \\ v_4 &= u_1 \\ v_3 &= u_0 \\ v_2 &= u_1 + u_2 + u_3 \\ v_1 &= u_0 + u_1 + u_2 \\ v_0 &= u_0 + u_2 + u_3 \end{aligned}$$

The code word corresponding to the message (1 0 1 1) is (1 0 0 1 0 1 1)

If the generator matrix of an (n, k) linear code is in systematic form, the parity-check matrix may take the following form

$$\mathbf{H} = \begin{bmatrix} \mathbf{I}_{n-k} & \mathbf{P}^T \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & . & . & . & 0 & p_{00} & p_{10} & . & . & . & p_{k-1,0} \\ 0 & 1 & 0 & . & . & . & 0 & p_{01} & p_{11} & . & . & . & p_{k-1,1} \\ 0 & 0 & 1 & . & . & . & 0 & p_{02} & p_{12} & . & . & . & p_{k-1,2} \\ . & & & & & & & & & & & & \\ . & & & & & & & & & & & & \\ . & & & & & & & & & & & & \\ 0 & 0 & 0 & . & . & . & 1 & p_{0,n-k-1} & p_{1,n-k-1} & . & . & . & p_{k-1,n-k-1} \end{bmatrix}$$

Encoding circuit for a linear systematic (n,k) code is shown below.

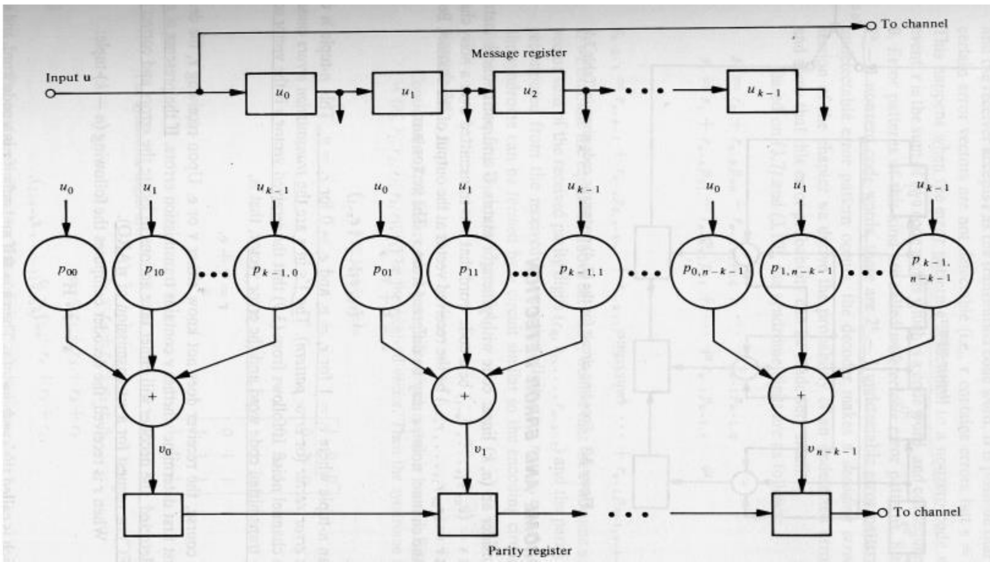


Figure: Encoding Circuit

For the block of k=4 message bits, (n-k) parity bits or check bits are added. Hence the total bits at the output of channel encoder are n=7. The encoding circuit for (7, 4) systematic code is shown below.

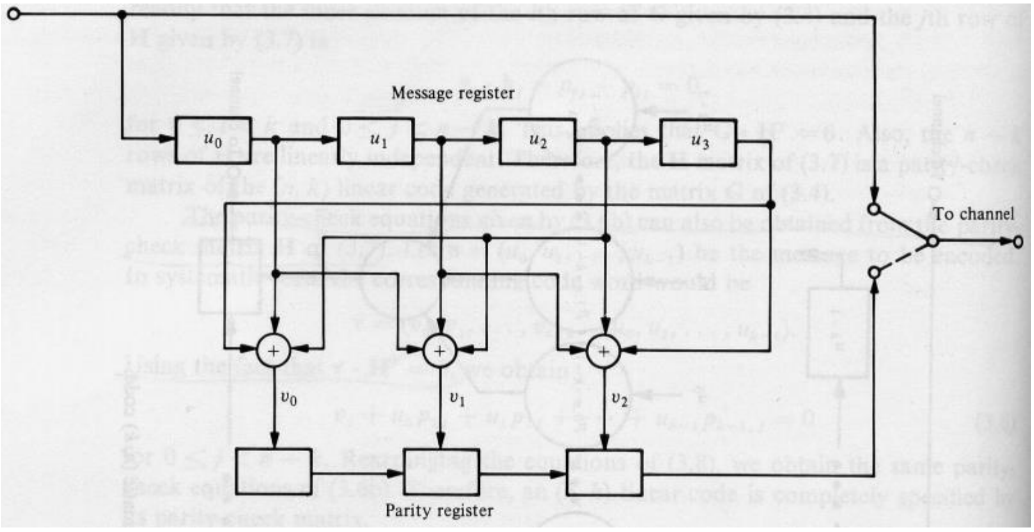
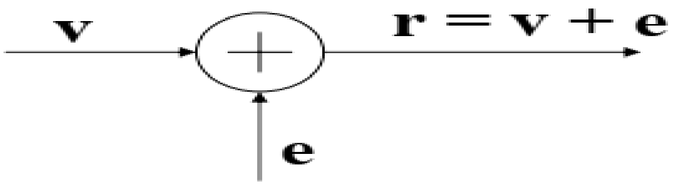


Figure: Encoding Circuit for (7,4) code

Syndrome and Error Detection

Let v = (v_0, v_1, …, v_{n-1}) be a code word that was transmitted over a noisy channel. Letr = (r_0, r_1, …, r_{n-1}) be the received vector at the outputof the channel



Where

$e = r + v = (e_0, e_1, \ldots, e_{n-1})$ is an n-tuple and the n-tuple 'e' is called the error vector (or error pattern).The condition is

$e_i = 1$ for $r_i \neq$
v_i $e_i = 0$ for r_i
$= v_i$

Upon receiving r, the decoder must first determine whether r contains transmission errors. If the presence of errors is detected, the decoder will take actions to locate the errors, correct errors (FEC) and request for a retransmission of v.
When r is received, the decoder computes the following (n – k)-tuple.

$s = r \cdot H^T$
$s = (s_0, s_1, \ldots, s_{n-k-1})$

where s is called the syndrome of r.

The syndrome is not a function of the transmitted codeword but a function of error pattern. So we can construct only a matrix of all possible error patterns with corresponding syndrome.

When s = 0, if and only if r is a code word and hence receiver accepts r as the transmitted code word. When $s \neq 0$, if and only if r is not a code word and hence the presence of errors has been detected. When the error pattern e is identical to a nonzero code word (i.e., r contain errors but $s = r \cdot H^T = 0$), error patterns of this kind are called undetectable error patterns. Since there are $2^k - 1$ non-zero code words, there are $2^k - 1$ undetectable error patterns. The syndrome digits are as follows:

$s_0 = r_0 + r_{n-k} p_{00} + r_{n-k+1} p_{10} + \cdots + r_{n-1} p_{k-1,0}$
$s_1 = r_1 + r_{n-k} p_{01} + r_{n-k+1} p_{11} + \cdots + r_{n-1} p_{k-1,1}$

.

$s_{n-k-1} = r_{n-k-1} + r_{n-k} p_{0,n-k-1} + r_{n-k+1} p_{1,n-k-1} + \cdots + r_{n-1} p_{k-1,n-k-1}$

The syndrome s is the vector sum of the received parity digits $(r_0, r_1, \ldots, r_{n-k-1})$ and the parity-check digits recomputed from the received information digits $(r_{n-k}, r_{n-k+1}, \ldots, r_{n-1})$.

The below figure shows the syndrome circuit for a linear systematic (n, k) code.

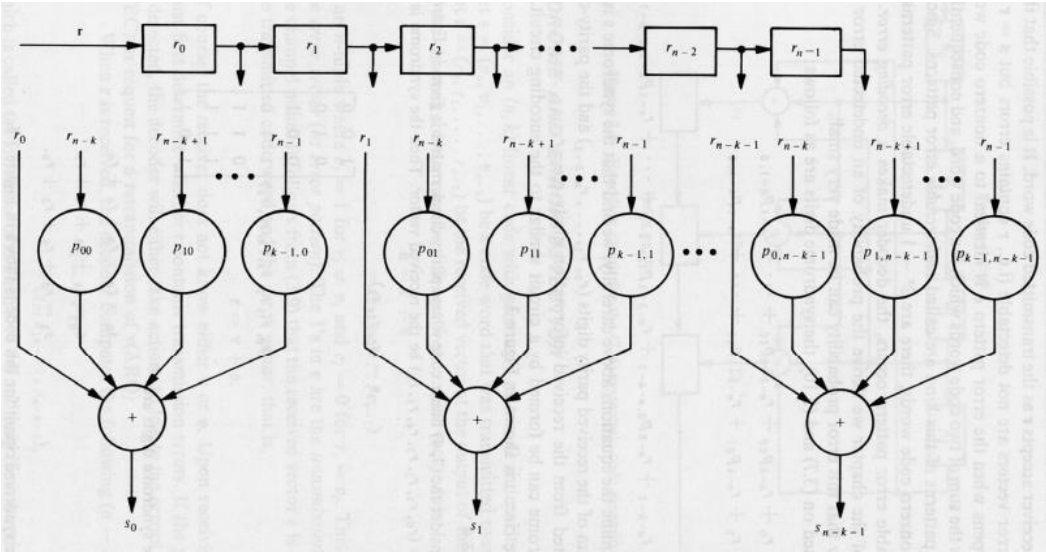


Figure: Syndrome Circuit

Error detection and error correction capabilities of linear block codes:

If the minimum distance of a block code C is d_{min}, any two distinct code vector of C differ in at least d_{min} places. A block code with minimum distance d_{min} is capable of detecting all the error pattern of d_{min}– 1 or fewer errors.

However, it cannot detect all the error pattern of d_{min} errors because there exists at least one pair of code vectors that differ in d_{min} places and there is an error pattern of d_{min} errors that will carry one into the other. The random-error-detecting capability of a block code with minimum distance d_{min} is d_{min}– 1.

An (n, k) linear code is capable of detecting 2n – 2k error patterns of length n Among the 2n – 1 possible non zero error patterns, there are 2k – 1 error patterns that are identical to the 2k – 1 non zero code words. If any of these 2k – 1 error patterns occurs, it alters the transmitted code word v into another code word w, thus w will be received and itssyndrome is zero.

If an error pattern is not identical to a nonzero code word, the received vector r will not be a code word and the syndrome will not be zero.

Hamming Codes:

These codes and their variations have been widely used for error control in digital communication and data storage systems.

For any positive integer m ≥ 3, there exists a Hamming code with the following parameters:
Code length: n = 2m – 1
Number of information symbols: k = 2m – m – 1
Number of parity-check symbols: n – k = m
Error-correcting capability: t = 1(dmin= 3)

The parity-check matrix H of this code consists of all the non zero m-tuple as its columns (2m-1)

In systematic form, the columns of H are arranged in the following form

$H = [I_m\ Q]$

where I_m is an m × m identity matrix

The sub matrix Q consists of 2m – m – 1 columns which are the m-tuples of weight 2 or more. The columns of Q may be arranged in any order without affecting the distance property and weight distribution of the code.

In systematic form, the generator matrix of the code is

G = [QT I2m–m–1]

where QT is the transpose of Q and I 2m–m–1 is an (2m – m – 1) ×(2m – m – 1) identity matrix.

Since the columns of H are nonzero and distinct, no two columns add to zero. Since H consists of all the nonzero m-tuples as its columns, the vector sum of any two columns, say h_i and h_j, must also be a column in H, say $h_l h_i + h_j + h_l = 0$.The minimum distance of a Hamming code is exactly 3.

Using H' as a parity-check matrix, a shortened Hamming code can be obtained with the following parameters :

Code length: n = 2m – l – 1

Number of information symbols: k = 2m – m – l – 1

Number of parity-check symbols: n – k = m

Minimum distance : $d_{min} \geq 3$

When a single error occurs during the transmission of a code vector, the resultant syndrome is nonzero and it contains an odd number of 1’s (e x H’T corresponds to a column in H’).When double errors occurs, the syndrome is nonzero, but it contains even number of 1’s.

Decoding can be accomplished in the following manner:

i) If the syndrome s is zero, we assume that no error occurred

ii) If s is nonzero and it contains odd number of 1’s, assume that a single error occurred. The error pattern of a single error that corresponds to s is added to the received vector for error correction.

iii) If s is nonzero and it contains even number of 1’s, an uncorrectable error pattern has been detected.

Problems:

1.

The parity check bits of a (8,4) block code are generated by

$$c_0 = m_0 + m_1 + m_3$$
$$c_1 = m_0 + m_1 + m_2$$
$$c_2 = m_0 + m_2 + m_3$$
$$c_3 = m_1 + m_2 + m_3$$

where m_1, m_2, m_3 and m_4 are the message digits.

(a) Find the generator matrix and the parity check matrix for this code.

(b) Find the minimum weight of this code.

(c) Find the error-detecting capabilities of this code.

(d) Show through an example that this code can detect three errors/codeword.

Solution

1(a) $\mathbf{c} = [c_0 \cdots c_8] = [b_0 \cdots b_3 m_0 \cdots m_3] = [m_0 \cdots m_3]\left[\begin{matrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{matrix} \vdots \mathbf{I}_4\right]$

$$\text{Therefore, } \mathbf{G} = \left[\begin{matrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{matrix} \vdots \mathbf{I}_4\right]$$

$$\text{and then } \mathbf{H} = \left[\mathbf{I}_4 \vdots \begin{matrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{matrix}\right]$$

(b)

m	C
0000	0000 0000
0001	1011 0001
0010	0111 0010
0011	1100 0011
0100	1101 0100
0101	0110 0101
0110	1010 0110
0111	0001 0111
1000	1110 1000
1001	0101 1001
1010	1001 1010
1011	0010 1011
1100	0011 1100
1101	1000 1101
1110	0100 1110
1111	1111 1111

Therefore, minimum weight = 4

(c) d_{min} = minimum weight = 4

Therefore, error-detecting capability = $d_{min} - 1 = 3$

(d) Suppose the transmitted code be 00000000 and the received code be 11100000.

$$\mathbf{s} = \mathbf{r}\mathbf{H}^{\mathrm{T}} = [11100000] \begin{bmatrix} & 1 & 1 & 0 & 1 \\ \mathbf{I}_4 \vdots & 1 & 1 & 1 & 0 \\ & 1 & 0 & 1 & 1 \\ & 0 & 1 & 1 & 1 \end{bmatrix}^T = [1110] \neq \mathbf{0}$$

Binary Cyclic codes:

Cyclic codes are the sub class of linear block codes.

Cyclic codes can be in systematic or non systematic form.

Definition:

A linear code is called a cyclic code if every cyclic shift of the code vector produces some other code vector.

Properties of cyclic codes:

(i) Linearity (ii) Cyclic

Linearity: This property states that sum of any two code words is also a valid code word.

$$X_1+X_2=X_3$$

Cyclic: Every cyclic shift of valid code vector produces another valid code vector.

Consider an n-bit code vector

$$X = \{x_{n-1}, x_{n-2}, \ldots\ldots\ldots\ldots x_1, x_0\}$$

Here x_{n-1}, x_{n-2} ….x_1, x_0 represent individual bits of the code vector 'X'.

If the above code vector is cyclically shifted to left side i.e., One cyclic shift of X gives,

$$X' = \{x_{n-2} \ldots x_1, x_0, x_{n-1}\}$$

Every bit is shifted to left by one position.

Algebraic Structures of Cyclic Codes:

The code words can be represented by a polynomial. For example consider the n-bit code word X = {x_{n-1},x_{n-2},.....................x_1,x_0}.

This code word can be represented by a polynomial of degree less than or equal to (n-1) i.e.,

$$X(p)=x_{n-1}p^{n-1}+x_{n-2}p^{n-2}+ \dots\dots\dots\dots +x_1p+x_0$$

Here X(p) is the polynomial of degree (n-1)

p- Arbitrary variable of the polynomial

The power of p represents the positions of the codeword bits i.e.,

p^{n-1} – MSB

p^0 -- LSB

p -- Second bit from LSB side

Polynomial representation due to the following reasons

(i) These are algebraic codes, algebraic operations such as addition, multiplication, division, subtraction etc becomes very simple.
(ii) Positions of the bits are represented with help of powers of p in a polynomial.

<u>Generation of code words in Non-systematic form:</u>

Let M= $\{m_{k-1}, m_{k-2},\dots\dots\dots\dots m_1,m_0\}$ be 'k' bits of message vector. Then it can be represented by the polynomial as,

$M(p)=m_{k-1}p^{k-1}+m_{k-2}p^{k-2}+\dots\dots\dots\dots+m_1p+m_0$

Let X(p) be the code word polynomial

X(p)=M(p)G(p)

G(p) is the generating polynomial of degree 'q'

For (n,k) cyclic codes, q=n-k represent the number of parity bits.

The generating polynomial is given as

$G(p)= p^q+g_{q-1}p^{q-1}+ \dots\dots\dots +g_1p+1$

Where $g_{q-1}, g_{q-2}, \dots\dots\dots\dots g_1$ are the parity bits.

If M_1, M_2, M_3...................................etc are the other message vectors, then the corresponding code vectors can be calculated as

$$X_1(p) = M_1(p)\, G(p)$$

$$X_2(p) = M_2(p)\, G(p)$$

$$X_3(p) = M_3(p)\, G(p)$$

Generation of Code vectors in systematic form:

$$X = (k \text{ message bits} : (n-k) \text{ check bits}) = (m_{k-1}, m_{k-2}, \ldots\ldots m_1, m_0 : c_{q-1}, c_{q-2}, \ldots\ldots c_1, c_0)$$

$$C(p) = c_{q-1}p^{q-1} + c_{q-2}p^{q-2} + \ldots\ldots + c_1 p + c_0$$

The check bit polynomial is obtained by

$$C(p) = \text{rem}\left[\frac{p^q M(p)}{G(p)}\right]$$

Generator and Parity Check Matrices of cyclic codes:

Non systematic form of generator matrix:

Since cyclic codes are sub class of linear block codes, generator and parity check matrices can also be defined for cyclic codes.

The generator matrix has the size of k x n.

Let generator polynomial given by equation

$G(p) = p^q + g_{q-1}p^{q-1} + \ldots\ldots + g_1 p + 1$

Multiply both sides of this polynomial by p^i i.e.,

$$p^i G(p) = p^{i+q} + g_{q-1}p^{i+q-1} \ldots\ldots + g_1 p^{i+1} + p^i \quad \text{and } i = (k-1), (k-2), \ldots\ldots 2, 1, 0$$

Systematic form of generator matrix:

Systematic form of generator matrix is given by

$$G = [I_k : P_{kxq}]_{kxn}$$

The t^{th} row of this matrix will be represented in the polynomial form as follows

$$t^{th} \text{ row of } G = p^{n-t} + R_t(p)$$

Where t= 1, 2, 3 ....k

Lets divide p^{n-t} by a generator matrix G(p). Then we express the result of this division in terms of quotient and remainder i.e.,

$$\frac{p^{n-t}}{G(p)} = Quotient + \frac{Remainder}{G(p)}$$

Here remainder will be a polynomial of degree less than q, since the degree of G(p) is ‘q’.

The degree of quotient will depend upon value of t

Lets represent Remainder =

Rt(p)Quotient =

Qt(p)

$$\frac{p^{n-t}}{G(p)} = Q_t(p) + \frac{Rt(p)}{G(p)}$$

$$p^{n-t} = Qt(p)G(p) + Rt(p)$$

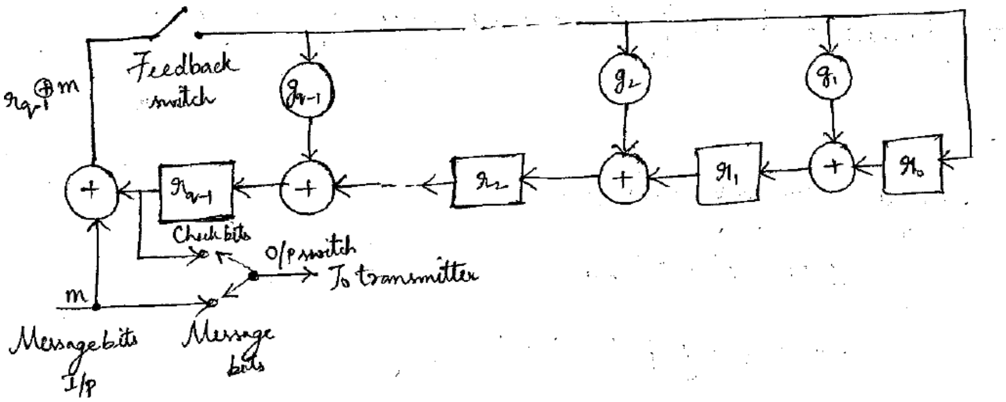
And t= 1,2, ...k

$$p^{n-t} + R(p) = Qt(p)G(p)$$

Represents t^{th} row of systematic generator matrix

Parity check matrix **H = [P^T : Iq]qxn**

<u>Encoding using an (n-k) Bit Shift Register:</u>



The feedback switch is first closed. The output switch is connected to message input. All the shift registers are initialized to zero state. The ‘k’ message bits are shifted to the transmitter as well as shifted to the registers.

After the shift of 'k' message bits the registers contain 'q' check bits. The feedback switch is now opened and output switch is connected to check bits position. With the every shift, the check bits are then shifted to the transmitter.

The block diagram performs the division operation and generates the remainder. Remainder is stored in the shift register after all message bits are shifted out.

Syndrome Decoding, Error Detection and Error Correction:

In cyclic codes also during transmission some errors may occur. Syndrome decoding can be used to correct those errors.

Lets represent the received code vector by Y.

If 'E' represents the error vector then the correct code vector can be obtained as

X=Y+E or Y=X+E

In the polynomial form we can write above equation as

$$Y(p) = X(p)+E(p)$$

$$X(p) = M(p)G(p)$$

$$Y(p)= M(p)G(p) + E(p)$$

$$\frac{Y(p)}{G(p)} = Quotient + \frac{Remainder}{G(p)}$$

If Y(p)=X(p)

$$\frac{X(p)}{G(p)} = Quotient + \frac{Remainder}{G(p)}$$

$$\frac{Y(p)}{G(p)} = Q(p) + \frac{R(p)}{G(p)}$$

$$Y(p)=Q(p)G(p) + R(p)$$

Clearly R(p) will be the polynomial of degree less than or equal to q-1

$$Y (p) =Q (p) G (p) +R (p)$$

$$M(p)G(p)+E(p)=Q(p)G(p)+R(p)$$

$$E(p)=M(p)G(p)+Q(p)G(p)+R(p)$$

$$E(p)=[M(p)+Q(p)]G(p)+R(p)$$

This equation shows that for a fixed message vector and generator polynomial, an error pattern or error vector 'E' depends on remainder R.

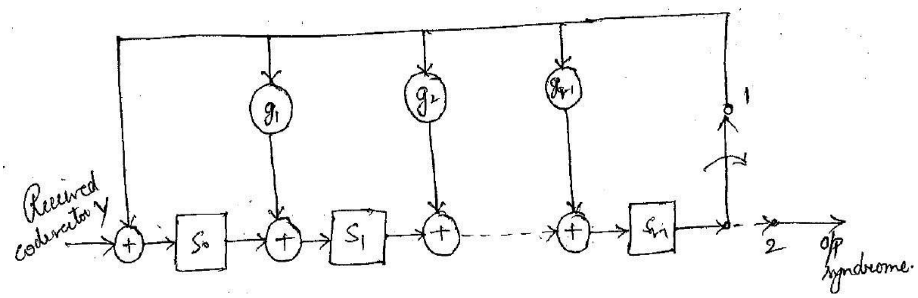
For every remainder 'R' there will be specific error vector. Therefore we can call the remainder vector 'R' as syndrome vector 'S', or R(p)=S(p). Therefore

$$\frac{Y(p)}{G(p)} = Q(p) + \frac{S(p)}{G(p)}$$

Thus Syndrome vector is obtained by dividing received vector Y (p) by G (p) i.e.,

$$S(p) = rem\left[\frac{Y(p)}{G(p)}\right]$$

Block Diagram of Syndrome Calculator:



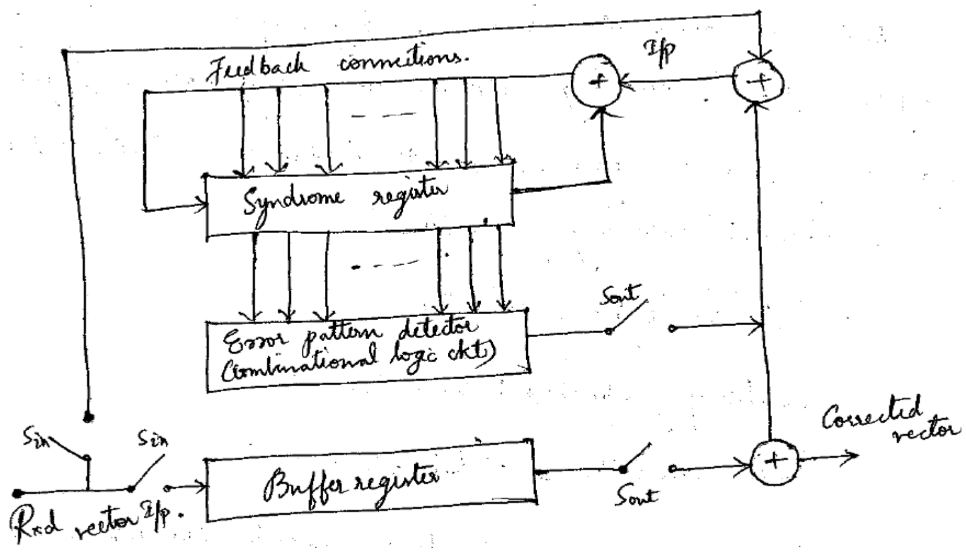
There are 'q' stage shift register to generate 'q' bit syndrome vector. Initially all the shift register contents are zero & the switch is closed in position 1.

The received vector Y is shifted bit by bit into the shift register. The contents of flip flops keep changing according to input bits of Y and values of g1,g2 etc.

After all the bits of Y are shifted, the 'q' flip flops of shift register contain the q bit syndrome vector. The switch is then closed to position 2 & clocks are applied to shift register. The output is a syndrome vector S= (Sq-1, Sq-2 ….S1, S0)

Decoder of Cyclic Codes:

Once the syndrome is calculated, then an error pattern is detected for that particular syndrome. When the error vector is added to the received code vector Y, then it gives corrected code vector at the output.



The switch named Sout is opened and Sin is closed. The bits of the received vector Y are shifted into the buffer register as well as they are shifted in to the syndrome calculator. When all the n bits of the received vector Y are shifted into the buffer register and Syndrome calculator the syndrome register holds a syndrome vector.

Syndrome vector is given to the error pattern detector. A particular syndrome detects a specific error pattern.

Sin is opened and Sout is closed. Shifts are then applied to the flip flop of buffer registers, error register, and syndrome register.

The error pattern is then added bit by bit to the received vector. The output is the corrected error free vector.

Convolution codes

Definition of Convolutional Coding

... (4.4.1)

and $x_2 = m \oplus m_2$ ... (4.4.2)

A convolutional coding is done by combining the fixed number of input bits. The input bits are stored in the fixed length shift register and they are combined with the help of mod-2 adders. This operation is equivalent to binary convolution and hence it is called *convolutional coding*. This concept is illustrated with the help of simple example given below.

m, m_1 and m_2 (equation 4.4.1 and equation 4.4.2). The output switch then samples x_1 then x_2. Thus the output bit stream for successive input bits will be,

... (4.4.3)

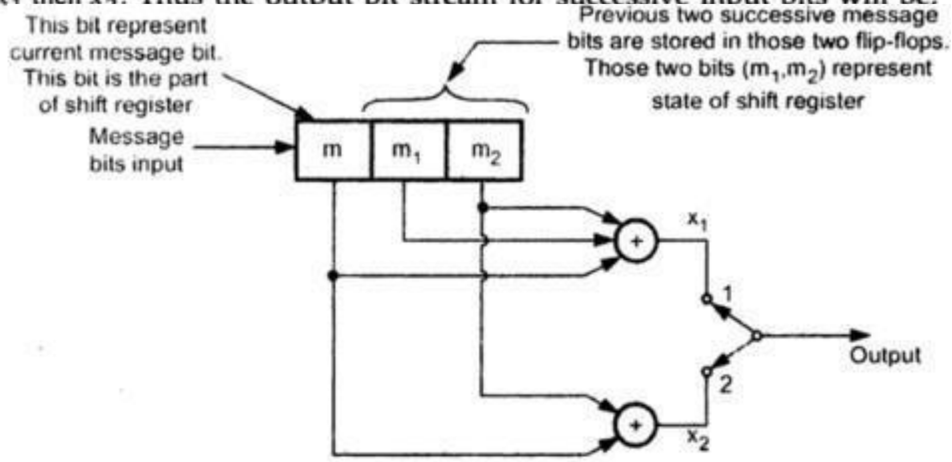


Fig. 4.4.1 Convolutional encoder with k = 3, k = 1 and n = 2

Operation :

Whenever the message bit is shifted to position 'm', the new values of x_1 and x_2 are generated depending upon m, m_1 and m_2. m_1 and m_2 store the previous two message bits. The current bit is present in m. Thus we can write, Here note that for every input message bit two encoded output bits x_1 and x_2 are transmitted. In other words, for a single message bit, the encoded code word is two bits i.e. for this convolutional encoder,

Number of message bits, k = 1

Number of encoded output bits for one message bit, n = 2

4.4.1.1 Code Rate of Convolutional Encoder

The code rate of this encoder is,

$$r = \frac{k}{n} = \frac{1}{2} \qquad \text{... (4.4.4)}$$

In the encoder of Fig. 4.4.1, observe that whenever a particular message bit enters a shift register, it remains in the shift register for three shifts i.e.,

First shift → Message bit is entered in position 'm'.

Second shift → Message bit is shifted in position m_1.

Third shift → Message bit is shifted in position m_2.

And at the fourth shift the message bit is discarded or simply lost by overwriting. We know that x_1 and x_2 are combinations of m, m_1, m_2. Since a single message bit remains in m during first shift, in m_1 during second shift and in m_2 during third shift; it influences output x_1 and x_2 for 'three' successive shifts.

4.4.1.2 Constraint Length (K)

The constraint length of a convolution code is defined as the number of shifts over which a single message bit can influence the encoder output. It is expressed in terms of message bits.

For the encoder of Fig. 4.4.1 constraint length K = 3 bits. This is because in this encoder, a single message bit influences encoder output for three successive shifts. At the fourth shift, the message bit is lost and it has no effect on the output.

4.4.1.3 Dimension of the Code

The dimension of the code is given by n and k. We know that 'k' is the number of message bits taken at a time by the encoder. And 'n' is the encoded output bits for one message bits. Hence the dimension of the code is (n, k). And such encoder is called (n, k) convolutional encoder. For example, the encoder of Fig. 4.4.1 has the dimension of (2, 1).

4.4.2 Time Domain Approach to Analysis of Convolutional Encoder

Let the sequence $\{g_0^{(1)}, g_1^{(1)}, g_2^{(1)} \ldots\ldots g_m^{(1)}\}$ denote the impulse response of the adder which generates x_1 in Fig. 4.4.1 Similarly, Let the sequence $\{g_0^{(2)}, g_1^{(2)}, g_2^{(2)} \ldots\ldots g_m^{(2)}\}$ denote the impulse response of the adder which generates x_2 in Fig. 4.4.1. These impulse responses are also called *generator sequences* of the code.

Let the incoming message sequence be $\{m_0, m_1, m_2 \ldots\ldots\}$. The encoder generates the two output sequences x_1 and x_2. These are obtained by convolving the generator sequences with the message sequence. Hence the name convolutional code is given. The sequence x_1 is given as,

$$x_1 = x_i^{(1)} = \sum_{l=0}^{M} g_l^{(1)} m_{i-l} \qquad \text{i = 0, 1, 2 ....} \qquad \ldots (4.4.6)$$

Here $m_{i-l} = 0$ for all $l > i$. Similarly the sequence x_2 is given as,

$$x_2 = x_i^{(2)} = \sum_{l=0}^{M} g_l^{(2)} m_{i-l} \qquad \text{i = 0, 1, 2 ....} \qquad \ldots (4.4.7)$$

Note : All additions in above equations are as per mod-2 addition rules.

As shown in the Fig. 4.4.1, the two sequences x_1 and x_2 are multiplexed by the switch. Hence the output sequence is given as,

$$\{x_i\} = \left\{ x_0^{(1)}\ x_0^{(2)}\ x_1^{(1)}\ x_1^{(2)}\ x_2^{(1)}\ x_2^{(2)}\ x_3^{(1)}\ x_3^{(2)} \ldots\ldots \right\} \qquad \ldots (4.4.8)$$

$$v_1 = x_i^{(1)} = \left\{ x_0^{(1)}\ x_1^{(1)}\ x_2^{(1)}\ x_3^{(1)} \ldots\ldots \right\}$$

$$v_2 = x_i^{(2)} = \left\{ x_0^{(2)}\ x_1^{(2)}\ x_2^{(2)}\ x_3^{(2)} \ldots\ldots \right\}$$

Observe that bits from above two sequences are multiplexed in equation (4.4.8) The sequence $\{x_i\}$ is the output of the convolutional encoder.

Transform Domain Approach to Analysis of Convolutional Encoder

In the previous section we observed that the convolution of generating sequence and message sequence takes place. These calculations can be simplified by applying the transformations to the sequences. Let the impulse responses be represented by polynomials. i.e.,

$$g^{(1)}(p) = g_0^{(1)} + g_1^{(1)}p + g_2^{(1)}p^2 + \dots\dots + g_M^{(1)}p^M \qquad \dots (4.4.13)$$

$$g^{(2)}(p) = g_0^{(2)} + g_1^{(2)}p + g_2^{(2)}p^2 + \dots\dots + g_M^{(2)}p^M \qquad \dots (4.4.14)$$

Thus the polynomials can be written for other generating sequences. The variable 'p' is unit delay operator in above equations. It represents the time delay of the bits in impulse response.

Similarly we can write the polynomial for message polynomial i.e.,

$$m(p) = m_0 + m_1p + m_2p^2 + \dots\dots\dots + m_{L-1}p^{L-1} \qquad \dots (4.4.15)$$

Here L is the length of the message sequence. The convolution sums are converted to polynomial multiplications in the transform domain. i.e.,

$$\boxed{\begin{aligned} x^{(1)}(p) &= g^{(1)}(p) \cdot m(p) \\ x^{(2)}(p) &= g^{(2)}(p) \cdot m(p) \end{aligned}} \qquad \dots (4.4.16)$$

The above equations are the output polynomials of sequences $x_i^{(1)}$ and $x_i^{(2)}$.

Code Tree, Trellis and State Diagram for a Convolution Encoder

Now let's study the operation of the convolutional encoder with the help of code tree, trellis and state diagram. Consider again the convolutional encoder of Fig. 4.4.1. It is reproduced below for convenience.

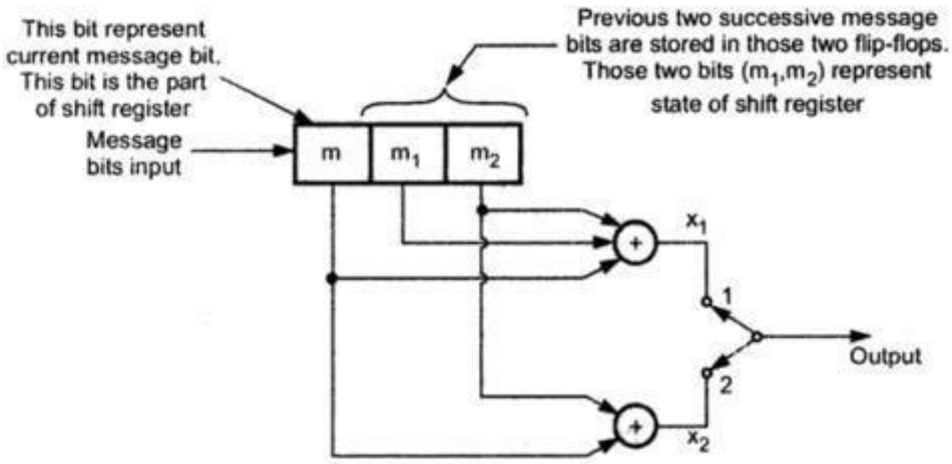


Fig. 4.4.4 Convolutional encoder with k = 1 and n = 2

States of the Encoder

In Fig. 4.4.4 the previous two successive message bits m_1 and m_2 represents state. The input message bit m affects the 'state' of the encoder as well as outputs x_1 and x_2 during that state. Whenever new message bit is shifted to 'm', the contents of m_1 and m_2 define new state. And outputs x_1 and x_2 are also changed according to new state m_1, m_2 and message bit m. Let's define these states as shown in Table 4.4.1.

Let the initial values of bits stored in m_1 and m_2 be zero. That is $m_1 m_2 = 00$ initially and the encoder is in state 'a'.

m_2	m_1	State of encoder
0	0	a
0	1	b
1	0	c
1	1	d

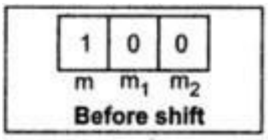
Table 4.4.1 States of the encoder of Fig. 4.4.4

Development of the Code Tree

Let us consider the development of code free for the message sequence m = 110. Assume that $m_1 m_2 = 00$ initially.

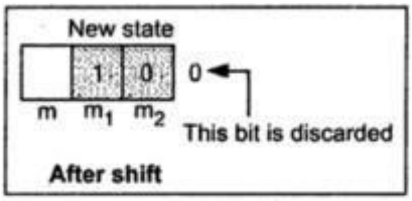
1) When $m = 1$ i.e. first bit

The first message input is m = 1. With this input x_1 and x_2 will be calculated as



$$x_1 = 1 \oplus 0 \oplus 0 = 1$$

$$x_2 = 1 \oplus 0 = 1$$



The values of $x_1 x_2 = 11$ are transmitted to the output and register contents are shifted to right by one bit position as shown.

Thus the new state of encoder is $m_2 m_1 = 01$ or *'b'* and output transmitted are $x_1 x_2 = 11$. This shows that if encoder is in state 'a' and if input is $m = 1$ then the next state is *'b'* and outputs are $x_1 x_2 = 11$. The first row of Table 4.4.2 illustrates this operation.

The last column of this table shows the code tree diagram. The code tree diagram starts at node or state *'a'*. The diagram is reproduced as shown in Fig. 4.4.5.

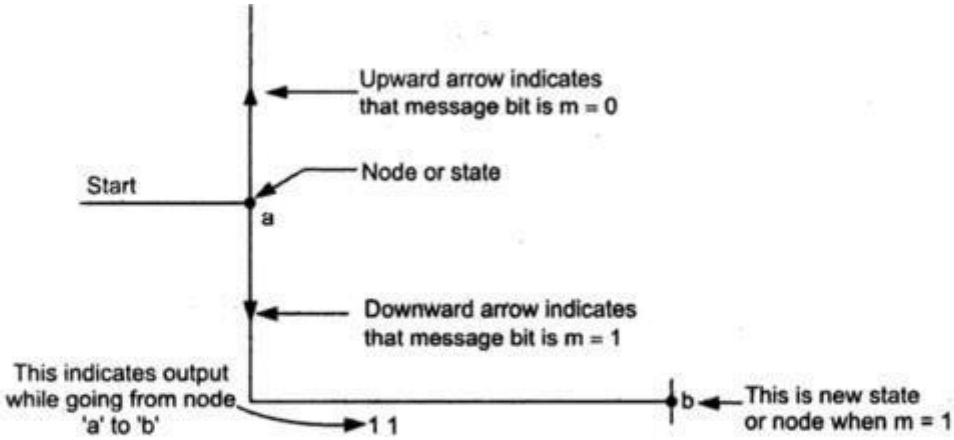
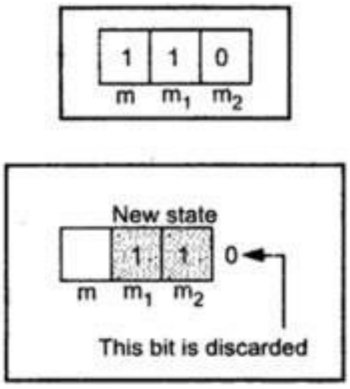


Fig. 4.4.5 Code tree from node *'a'* to *'b'*

Observe that if $m = 1$ we go downward from node *'a'*. Otherwise if $m = 0$, we go upward from node *'a'*. It can be verified that if $m = 0$ then next node (state) is 'a' only. Since $m = 1$ here we go downwards toward node b and output is 11 in this node (or state).

2) When $m = 1$ i.e. second bit

Now let the second message bit be 1. The contents of shift register with this input will be as shown below.



$x_1 = 1 \oplus 1 \oplus 0 = 0$

$x_2 = 1 \oplus 0 = 1$

These values of $x_1 x_2 = 01$ are then transmitted to output and register contents are shifted to right by one bit. The next state formed is as shown.

Thus the new state of the encoder is $m_2 m_1 = 11$ or *'d'* and the outputs transmitted are $x_1 x_2 = 01$. Thus the encoder goes from state 'b' to state *'d'* if input is '1' and transmitted output $x_1 x_2 = 01$. This operation is illustrated by Table 4.4.2 in second row. The last column of the table shows the code tree for those first and second input bits.

Example 4.4.8 : *Determine the state diagram for the convolutional encoder shown in Fig. 4.4.32. Draw the trellis diagram through the first set of steady state transitions. On the second trellis diagram, show the termination of trellis to all zero state.*

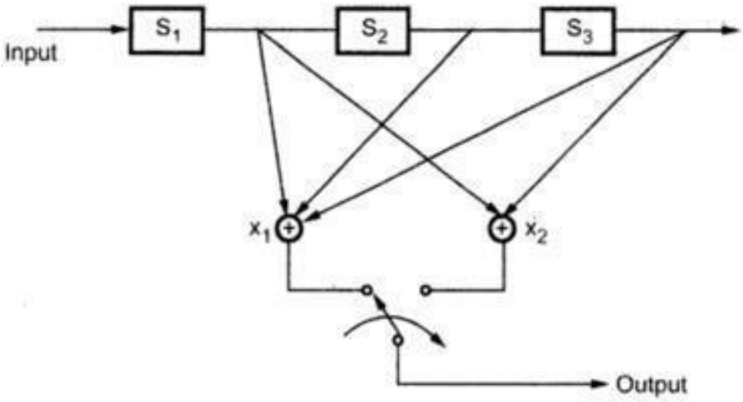


Fig. 4.4.32 Convolutional encoder of example 4.4.8

Sol. : (i) To determine dimension of the code :

For every message bit $(k=1)$, two output bits $(n=2)$ are generated. Hence this is rate $\frac{1}{2}$ code. Since there are three stages in the shift register, every message bit will affect output for three successive shifts. Hence constraint length, $K=3$. Thus,

$$k = 1, \quad n=2 \text{ and } K=3$$

ii) To obtain the state diagram :

First, let us define the states of the encoder.

$s_3 s_2 = 0\,0$, state 'a'

$s_3 s_2 = 0\,1$, state 'b'

$s_3 s_2 = 1\,0$, state 'c'

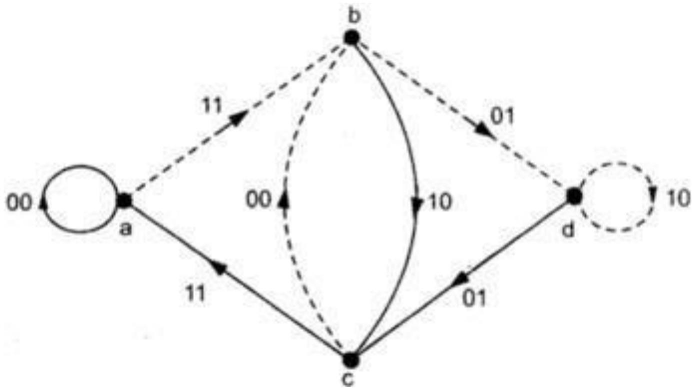
$s_3 s_2 = 1\,1$, state 'd'

A table is prepared that lists state transitions, message input and outputs. The table is as follows :

Sr. No.	Current state $s_3\ s_2$	Input s_1	Outputs $x_1 = s_1 \oplus s_2 \oplus s_3$ $x_2 = s_1 \oplus s_3$		Next state $s_2\ s_1$
1	a = 0 0	0	0	0	0 0, i.e. a
		1	1	1	0 1, i.e. b
2	b = 0 1	0	1	0	1 0, i.e. c
		1	0	1	1 1, i.e. d
3	c = 1 0	0	1	1	0 0, i.e. a
		1	0	0	0 1, i.e. b
4	d = 1 1	0	0	1	1 0, i.e. c
		1	1	0	1 1, i.e. d

Table 4.4.8 : State transition table

Based on above table, the state diagram can be prepared easily. It is shown below in Fig. 4.4.33.



iii) To obtain trellis diagram for steady state :

From Table 4.4.9, the code trellis diagram can be prepared. It is steady state diagram. It is shown below.

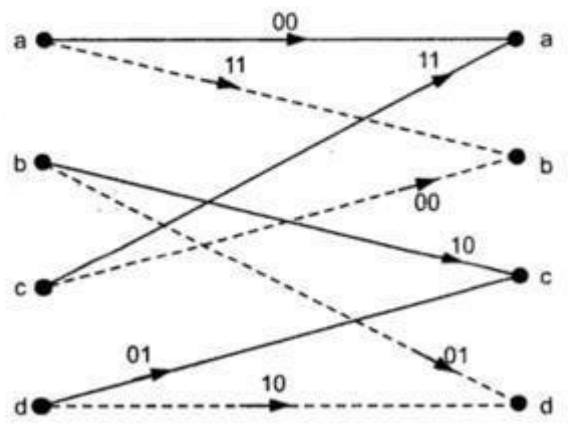


Fig. 4.4.34 Code trellis diagram for steady state

Decoding methods of Convolution code:

1.Veterbi decoding

2.Sequential

decoding3.Feedback

decoding

Veterbi algorithm for decoding of convolution codes (maximam likelihood decoding):

Let represent the received signal by y.

Convolutional encoding operates continuously on input dataHence

there areno code vectorsand blocks such as.

Metric:it is the discrepancybetwen the received signal y and the decoding signal atparticular node .this metric can be added over few nodes a particular path

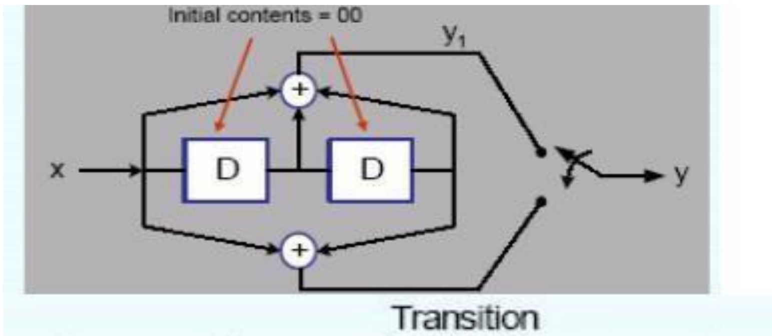
Surviving path: this is the path of the decoded signalwith minimum metric

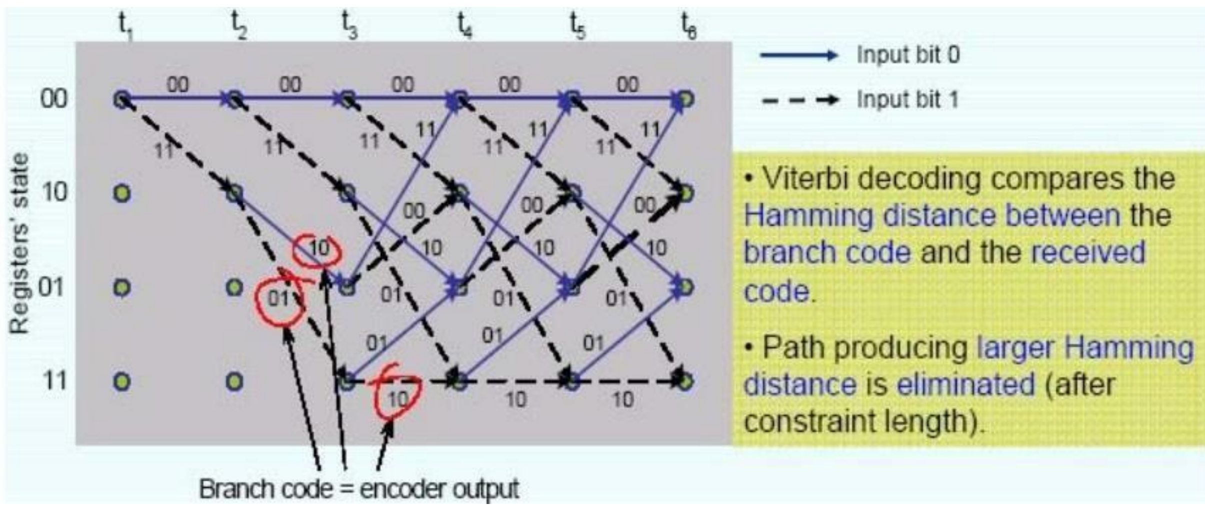
In veterbi decoding ametric isassigned to each surviving path

Metric of the particular is obtained by adding individual metric on the nodes along that path.

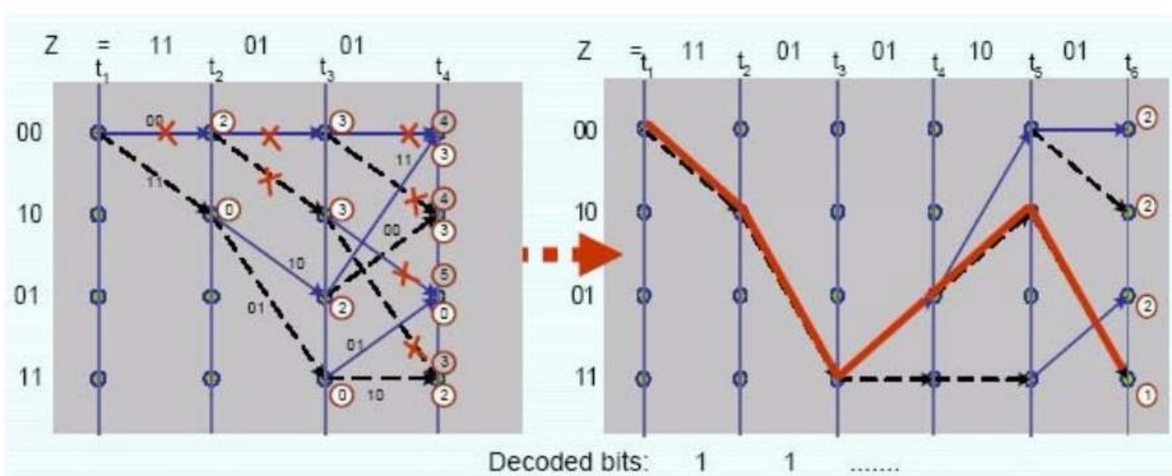
Y is decoded as the surviving path with smallest metric.

Example:





Input data : m =1 1 0 1 1
Codeword : X = 11 01 01 00 01
Received code : Z = 11 01 01 10 01



Trellis Coded Modulation (TCM)

TCM concepts

Euclidean Distance

A straight line distance between any two points is called the **Euclidean distance**. For a point p_1 at(x_1, y_1) and another point p_2 at (x_2, y_2), the Euclidean distance is given by the familiar formula

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

It is implicit that this distance is the shortest distance between these points. The Euclideandistance is an analog concept, the very concept of *distance* that we normally use day-to-day in theworld of real numbers. For signals, we define this distance in the I-Q plane. In Figure 1 we have a8PSK signal constellation. The radius is equal to 1 and represents the maximum amplitude. Each point of the constellation is a certain combination of a particular amplitude and phase. The distance between these points is can be measured in the manner described above and these are given in the Figure below. The distances given in Figure 1 are squared and are called **Squared Euclidean Distance (SED).** The smallest of these distances is called the **Minimum Squared Euclidean Distance (MSED)**, designated as d^2_{min} for a particular constellation.

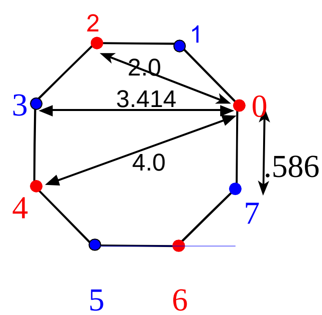


Figure 1 – 8 PSK constellation and squared Euclidean distances between symbols

Hamming distance

Just as real numbers have a concept of distance, so do the binary numbers. Take two binary numbers, *011011* and *101101*. The distance between these is the number of places these two numbers differ. And that number is 4. This distance is called the

Hamming distance between these numbers. The distance would be zero, if these two numbers were the same. A zero distancemeans the numbers are the same, same interpretation as in Euclidean concept of distance.

We distinguish these two types of distances by recognizing that one belongs to the analog worldof real numbers and the other to the binary world. Both concepts are useful in signal processing. In coding Hamming distance is most often used as a performance metric whereas it is Euclideandistance in the analog world.

Distance between sequences

We can also talk about Euclidean distances between *sequences* by comparing distances betweencorresponding points of the sequences. Let's take for example an 8PSK signal that consists of a sequence of these symbols.

S_0 S_3 S_2 S_1 S_0

In bits, we can map these as: 000 011 010 101
100 000

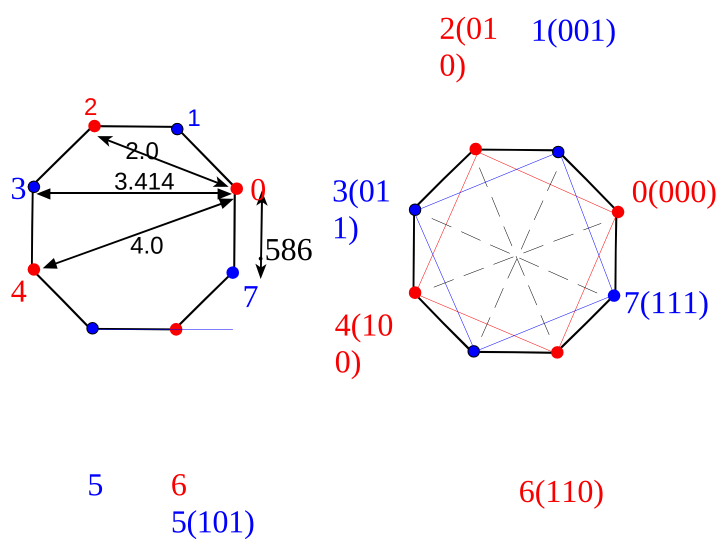


Figure 2 – Euclidean and sequence Hamming distance

The Euclidean distance for this sequence is the distance between each symbol in this sequence and a reference sequence. If we designate the **all-zero-symbols as the reference sequence**, thenthe squared Euclidean distance (SED) is the distance between each one of these symbols and thesymbol S_0.

s_0 to s_0 = 0.0, s_0 to s_1 = .586, s_0 to s_2 = 2.0, s_0 to s_3 = 3.414

The **Sum of the Squared Euclidean Distances (SSED)**, also called d^2_{free} of this sequence, fromthe all zero sequence is 3.414 + 2.0 + 0.586 = 6.0

This cumulative distance gives a feeling of how easy or difficult it would be to mistake one sequence for another. For the reference sequence we could have used any other sequence than theall zero, and the results would be the same. However using an all-zero sequence is convenient and conventional.

Trellis Coded Modulation (TCM)

TCM uses many diverse concepts from signal processing. In simplest terms it is a combination ofcoding and modulation, hence it name **Trellis Coded Modulation**, where the word trellis stands for the use of trellis (also called convolutional) codes. Whereas we normally talk about coding and modulation as two independent aspects of the communications link, in TCM they are combined.

TCM is a complex concept to understand particulalry due to the non-linear nature of the performance. It uses ideas from modulation and coding as well as dynamic programming, lattice structures and matrix math. A convolutional code that has optimum performance when used independently may not be optimum in TCM. Gray coding is helpful in uncoded signalling and constellation mapping, but not always so in TCM. So it is a not an easy topic and my hats off (if Iwore one) to Mr. Ungerboeck and others who came up with it.

Fortunately, there is not a lot of math to deal with here. But you will need to know concepts ofconvolutional codes, trellis, lattice, cosets, and coset generators.

Communications theory says that it is best to design codes in long sequences of messages. The **allowed sequences** should be very different from each other. The receiver can then make a decision between sequences using their statistics rather than on symbol-by-symbol basis. Whendecoding this way, the probability of error is an inverse function of the sequence length. In general form the probability of error between sequences is given by the expression

$$p \approx e^{-d^2_{min}/2\sigma^2}$$

where d_{min} is the sequence Euclidean distance between sequences and is σ^2 the noise power. We measure the performance of TCM (and many other schemes) by **Asymptotic Coding Gain (ACG)**. This is the gain obtained over some baseline performance at high SNR in a Gaussian environment. AGC is not achievable in practice because we do not transmit signals at high SNRs,have hardware and channel imperfections that depart from **Additive White Gaussian Noise (AWGN)** assumptions. So recognize that all gains quoted herein are maximum possible only in theory. Actual numbers are determined by test and simulation in the given environment.

The functions of a TCM consist of a **Trellis code** and a **constellation mapper** as shown in Figure 3. TCM combines the functions of a convolutional coder of rate $R = k/k+1$ and a M-ary signal mapper that maps $M = 2^k$ points.

input points into a larger constellation of $M = 2^{k+1}$ constellation

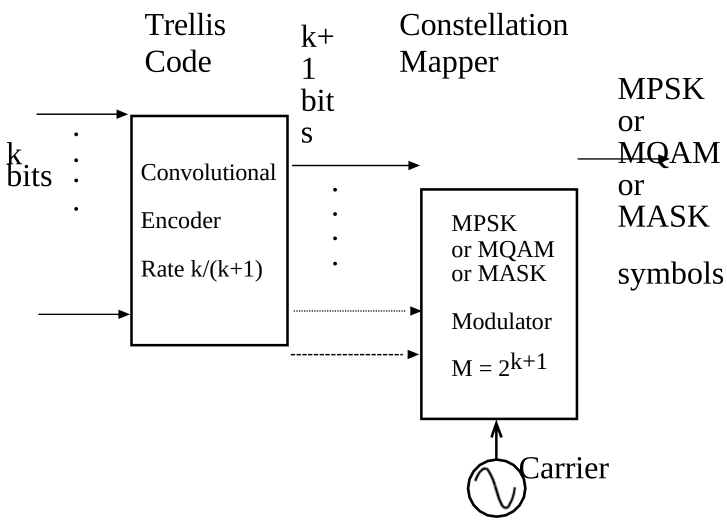


Figure 3 – A general trellis coded modulation

For k = 2, we have a code of rate 2/3 that takes a QPSK signal (M = 4) and putsout a 8-PSK signal (M = 8). So instead of expanding the bandwidth as the signal goes from QPSK to 8PSK, itinstead doubles the constellation points. It is kind of an upgrading system, where you take a chosen signal and upgrade it to another with larger number of constellation points as shown in Figure 4 below.

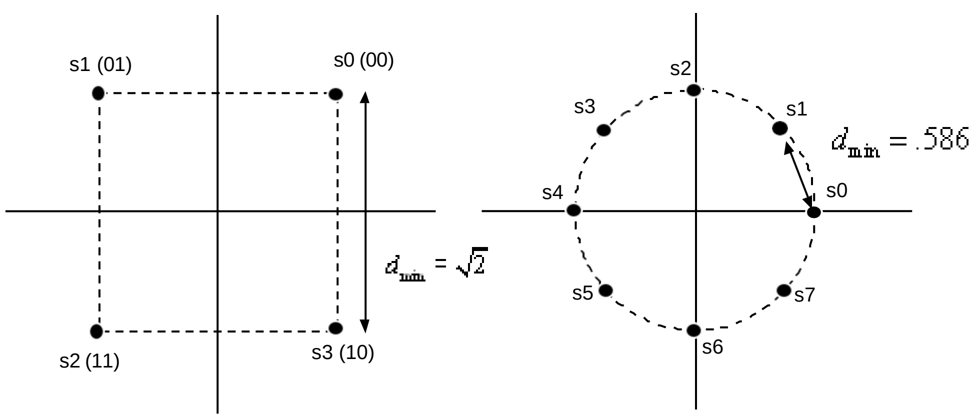


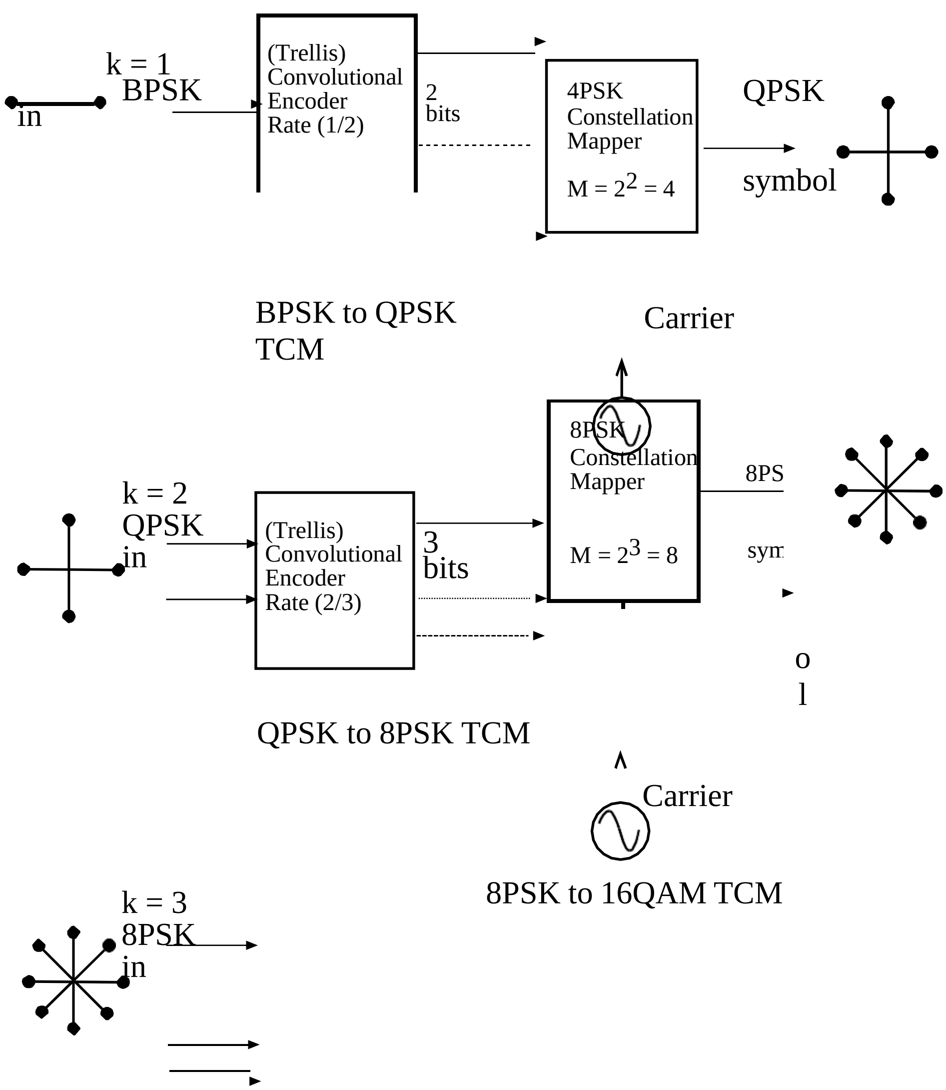
Figure 4 – Constellation doubling in TCM. A QPSK signal transmitted using a 8PSKconstellation.

Main points of a TCM are

1. TCM is **bandwidth efficient modulation** which accomplishes this by the use ofconvolutional coding.
2. It conserves bandwidth by doubling the number of constellation points of the signal. Thisway the bit rate increases but the symbol rate stays the same.

3. Convolutional coding constrains allowed symbol transitions, creating **sequence coding**.
4. Unlike a true Convolutional coding, not all incoming bits are coded.
5. Increasing the constellation size reduces Euclidean distances between the constellation points but sequence coding offers a coding gain that overcomes the power disadvantageof going to the higher constellation.
6. Performance is measured by coding gain over an uncoded signal.
7. The decoding metric is the Euclidean distance and not Hamming distance.
8. Ungerboeck originally proposed TCM which used **set-partitioning** and small number ofstates with code rates that varied with the input signal type.
9. Pragmatic TCM uses a less than perfect rate ½ convolutional code with constraint lengthequal to 7 or 9. This is a widely available code and its use makes TCM less expensive toimplement.
10. The **constellation mapping** in set partitioning is based on natural numbering where asgray coding is preferred in pragmatic TCM.

TCM is a general concept and by varying k, we can create a QPSK, 8PSK or higher levelsignals as shown in Figure 5. All of these are types of TCM.



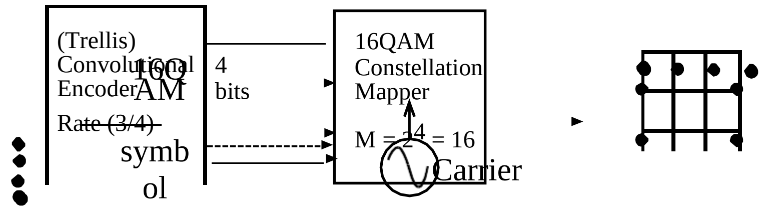


Figure 5 – General trellis coded modulation
(a) BPSK, code rate 1/2, output QPSK (b) QPSK, code rate = 2/3, output 8PSK
(c) 8PSK, code rate = 3/4, output 16QAM

Notice that in each case, the code rate is different. But they are all called TCM. The coding addsjust one extra bit to the symbol bit size. The symbol size increases from k bits to k + 1 bits. If coding increases the bit rate by 1 extra bit, then we need to double the constellation size to accommodate this bit as we see below, where L is original number of bits per symbol.

$$2^{L+1} = 2 \times 2^{L}$$

Overview of Turbo Codes ¾ The Turbo code concept was first introduced by C. Berrou in 1993. ¾ The name was derived from an iterative decoding algorithm used to decode these codes where, like a turbo engine, part of the output is reintroduced at the input and processed again. ¾ Performance close to the Shannon limit ¾ Drawbacks: − high decoding complexity − high latency due to interleaving and iterative decoding

Interleaving (1)

- ***Most well known codes have been designed for AWGN*** (Additive White Gaussian Noise) channels, i.e. the errors caused by the channel are randomly distributed and statistically independent.

  Thus when the channel is AWGN-channel these codes increase the reliability in the transmission of information.

- ***If the channel exhibits burst-error characteristics*** (due to the time-correlated multipath fading), these error clusters are not usually corrected by these codes.

  A burst of errors: a sequence of bit errors the 1st and last of which are 1s.

Interleaving (2)

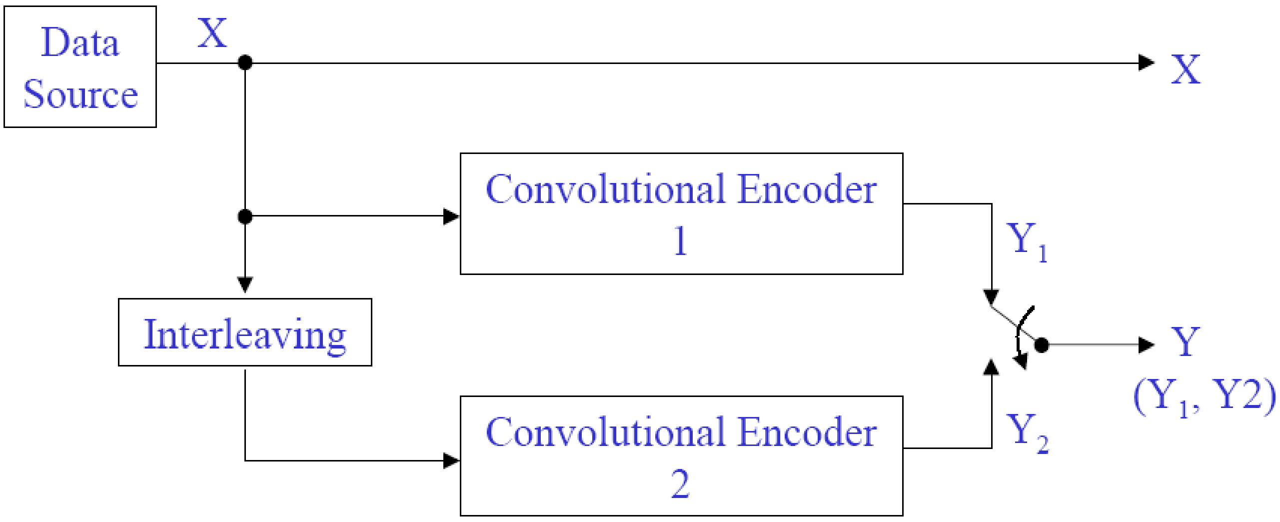
Solution: an interleaver

Its objective is to interleave the coded data in such a way that the bursty channel is transformed into a channel having independent errors and thus a code designed for independent channel errors (short burst) can be used.

- Disadvantages:
  - High latency (the entire interleaved block must be received before the critical data can be returned)
- *In turbo codes, the interleaver is used to permute the input bits such that the two encoders are operating on the same set of input bits, but different input sequences.*

Principles of Turbo Codes ¾ Formally, turbo codes should be described as parallelconcatenated systematic convolutional codes ¾ Components • Two or more systematic convolutional codes (usually identical, rate 1/2) • Pseudo-random interleaver • Soft-output iterative decoder, etc. ¾ Parallel concatenation • The encoders operate on the same set of input bits, rather than one encoding the output of the other • One reason for excellence performance ¾ The interleaver is used to permute the input bits • Length of the code is determined by the interleaver

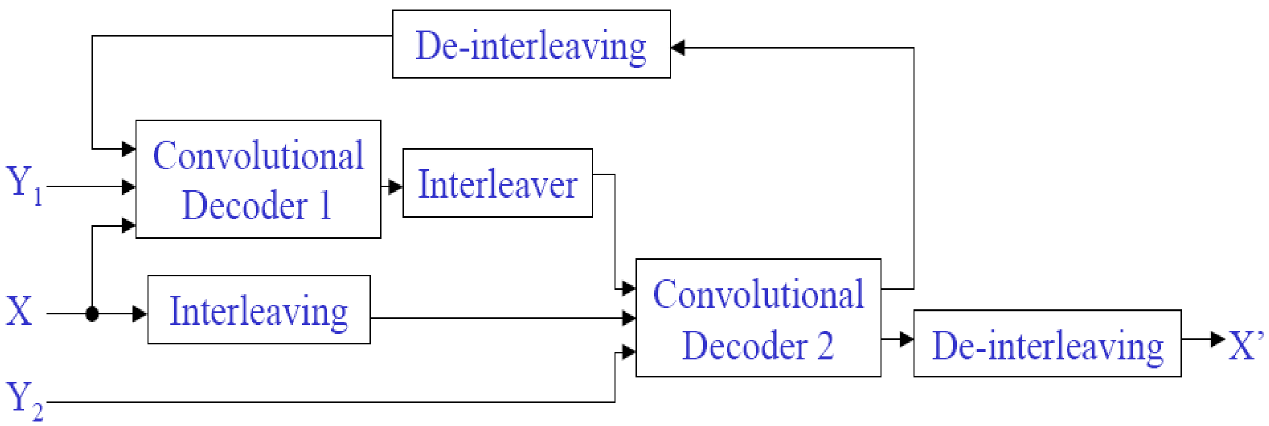
Turbo Codes : Encoder



X: Information

Y_i: Redundancy Information

Turbo Codes : Decoder



X': Decoded Information

1. State Channel Coding Theorem and its need.
2. Analyze the need for error control codes.
3. Outline the features of linear code
4. Interpret the code rate of a block code.
5. Demonstrate the significance of minimum distance of a block code
6. Express the syndrome properties of linear block code
7. Distinguish Hamming Distance and Hamming weight
8. Deduce the Hamming distance between 101010 and 010101. If the minimum Hamming distance of a (n, k) linear block code is 3
9. what is the minimum Hamming weight?
10. Summarize the advantages and disadvantages of Hamming codes
11. Discuss two properties of generator polynomial
12. List the properties of cyclic codes.
13. Illustrate the systematic code word with its structure.
14. When a binary code does is said to be cyclic codes?
15. Propose the generator polynomial of a cyclic codes.
16. Generate the cyclic code for (n, k) syndrome calculator.
17. The code vector [1110010] is sent, the received vector is [1100010]. Identify the Syndrome. What is meant by constraint length of a convolutional encoder?
18. Quote about convolutional code. How is it different from block codes?
19. Show how Trellis diagram is used to represent the code generated by convolutional coder and mention its advantages.
20. Determine the various techniques/algorithms used in encoding and decoding of convolutional code.

PART-B

1. Consider the (7,4) linear block code with generator matrix

$$\begin{pmatrix} 1000 & 101 \\ 0100 & 111 \\ 0010 & 110 \\ 0001 & 011 \end{pmatrix}$$

   I. Find all the code vectors.
   II. Find parity check matrix.
   III. Minimum weight of this code.

2. For a systematic (6, 3) linear block code, P= $\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}$. Analyze all the possible code vectors.
3. (i) Describe the steps involved in the generation of linear block codes. (ii) Explain the properties of syndrome.
4. Illustrate how the errors are corrected using hamming code with an example.
5. Recall syndrome decoding and explain its property with appropriate example.
6. Assume that the code word C=10110 for the (6,3) case is transmitted and the vector R=001110 is received. Show how a decoder using the syndrome lookup table can correct the error. Take generator matrix as G= $\begin{bmatrix} 110100 \\ 011010 \\ 101001 \end{bmatrix}$
7. An error control code has the following parity check matrix $H = \begin{bmatrix} 101100 \\ 110010 \\ 011001 \end{bmatrix}$
   I. What is the generator matrix G?
   II. Find the codeword that begins with 101…
   III. Decode the received codeword 110110.

IV. Comment on error correction and detection capability of this code.

.

8. Find the (7,4) systematic and non-systematic cyclic code words of the message word 1101.Assume the generator polynomial as $1+x^2+x^3$
9. Determine how Viterbi decoding algorithm is used for convolutional code.
10. Draw the state diagram of rate ½ convolutional encoder given in the figure below.

11. A convolutional code is described by the following generator sequences, g (1)= {1,0,1}, g(2)= {1,0,0}, g(3)= {1,1,1}. (i) Draw the encoder to this code (ii) Draw the state diagram (iii) If the message sequence is 10110, Design the code word.
12. Compare linear block codes and convolutional codes. (ii) State the advantages, disadvantages and applications of convolutional codes.
13. Develop the cyclic codes with the linear and cyclic property. Also represent the cyclic property of a code word in polynomial notation.
14. Draw the diagram of the ½ rate convolutional encoder with generator polynomials $G_1(D)=1+D$ $G_2(D)=1+D+D^2$ And complete the encoder output for input sequence 101101.
15. Find the (7,4) systematic and non-systematic cyclic code words of the message word 1101.Assume the generator polynomial as $1+x^2+x^3$.
16. For a systematic (6,3) linear block code $G = \begin{bmatrix} 1\,0\,0\,0\,1\,0 \\ 0\,0\,1\,1\,0\,1 \\ 0\,1\,1\,1\,1\,0 \end{bmatrix}$ (i) Solve for all the code vectors (ii) Draw encoder circuit for the above code (iii) Predict minimum hamming weight
17. For a systematic linear block code, the three parity check digits P1, P2,P3 are given by

    Pk,n-k = $\begin{bmatrix} 101 \\ 111 \\ 110 \\ 011 \end{bmatrix}$

    (i) Construct generated matrix.
    (ii) Assess the t code generated by the matrix.
    (iii) Determine error correcting capacity.
    (iv) Decode the received words with an example.
18. Explain about code tree, code trellis and state diagrams. Compare code tree with trellis diagram.
19. A convolutional code is described by g1=[1 0 0], g2=[1 1 1], g3=[1 0 1] (i) Build the encoder corresponding to the code. (ii) Develop the code tree and state diagram for this code. (iii) Draw the trellis diagram.

UNIT V – SPREAD SPECTRUM MODULATION –SECA1403

Introduction:

Initially developed for military applications during II world war, that was less sensitive to intentional interference or jamming by third parties. Spread spectrum technology has blossomed into one of the fundamental building blocks in current and next-generation wireless systems.

Problem of radio transmission

Narrow band can be wiped out due to interference. To disrupt the communication, the adversary needs to do two things,

(a) to detect that a transmission is taking place and

(b) to transmit a jamming signal which is designed to confuse the receiver.

Solution

A spread spectrum system is therefore designed to make these tasks as difficult as possible.

Firstly, the transmitted signal should be difficult to detect by an adversary/jammer, i.e., the signal should have a low probability of intercept (LPI).

Secondly, the signal should be difficult to disturb with a jamming signal, i.e., the transmitted signal should possess an anti-jamming (AJ) property

Remedy

spread the narrow band signal into a broad band to protect against interference

In a digital communication system the primary resources are **Bandwidth** and **Power.** The study of digital communication system deals with efficient utilization of these two resources, but there are situations where it is necessary to sacrifice their efficient utilization in order to meet certain other design objectives.

For example to provide a form of secure communication (i.e. the transmitted signal is not easily detected or recognized by unwanted listeners) the bandwidth of the transmitted signal is increased in excess of the minimum bandwidth necessary to transmit it. This requirement is catered by a technique known as **"Spread Spectrum Modulation".**

The primary advantage of a Spread – Spectrum communication system is its ability to reject '**Interference**' whether it be the unintentional or the intentional interference.

The definition of Spread – Spectrum modulation may be stated in two parts.

1. Spread Spectrum is a mean of transmission in which the data sequence occupies a BW (Bandwidth) in excess of the minimum BW necessary to transmit it.

2. The Spectrum Spreading is accomplished before transmission through the use of a code that is independent of the data sequence. The Same code is used in the receiver to despread the received signal so that the original data sequence may be recovered.

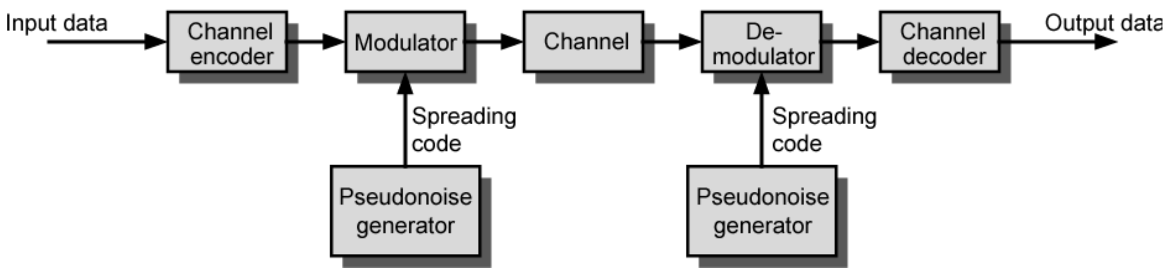


Fig. 5.1Block diagram for spread spectrum communication

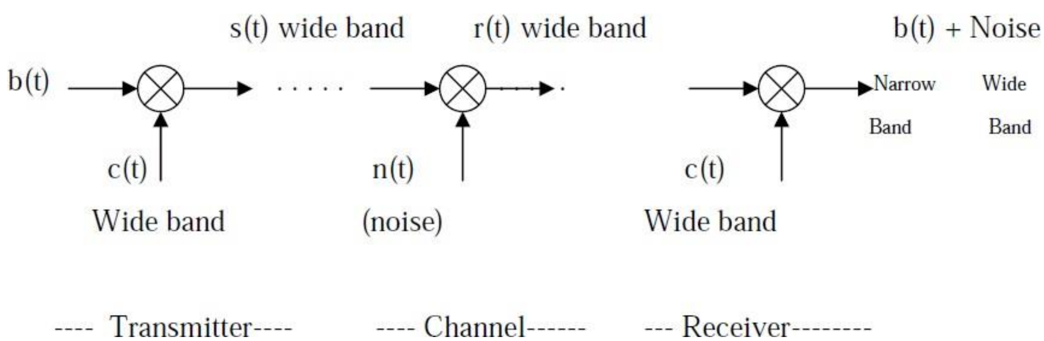


Fig:5.2 Spread spectrum technique.

b(t) = Data Sequence to be transmitted (Narrow Band);
c(t) = Wide Band code ;

s(t) = c(t) * b(t) – (wide Band)

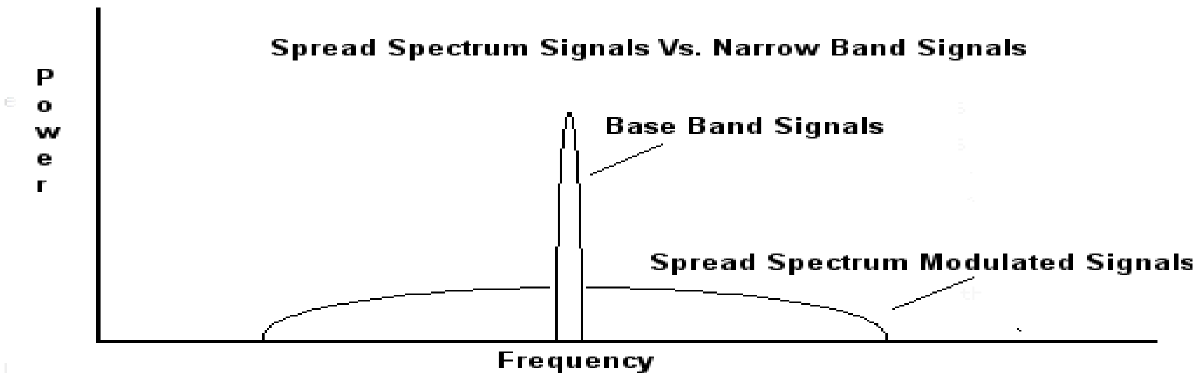


Fig:5.3 Spectrum of signal before & after spreading

PSUEDO-NOISE SEQUENCE:

Generation of PN sequence:

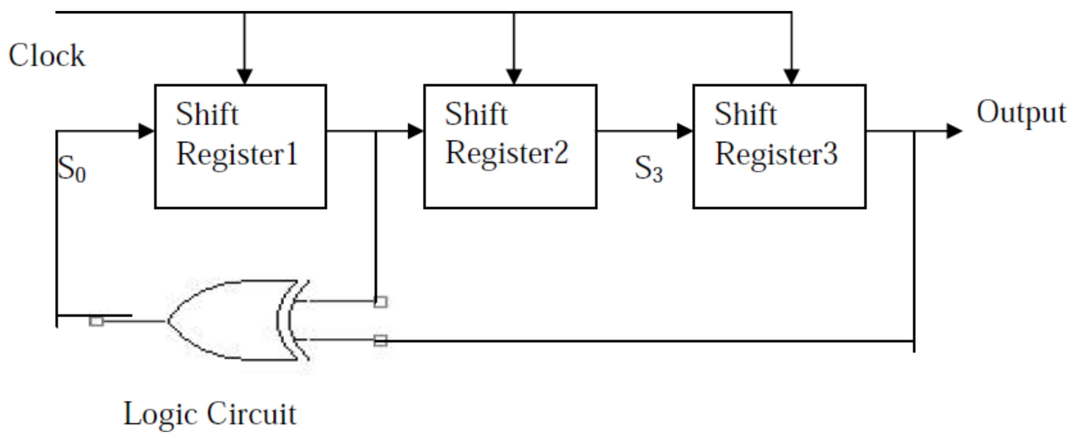


Fig: 5.4 Maximum-length sequence generator for n=3

A feedback shift register is said to be Linear when the feedback logic consists of entirely mod-2-address (Ex-or gates). In such a case, the zero state is not permitted. The period of a PN sequence produced by a linear feedback shift register with 'n' flip flops cannot exceed 2^n-1.

When the period is exactly 2^n-1, the PN sequence is called a '**maximum length sequence'** or '**m-sequence'**.

Example1: Consider the linear feedback shift register shown in above figure

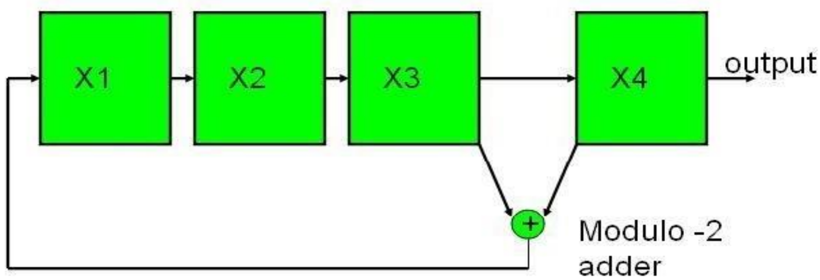
Involve three flip-flops. The input s_0 is equal to the mod-2 sum of S_1 and S_3. If the initial state of the shift register is 100. Then the succession of states will be as follows.

100,110,011,011,101,010,001,100 . . . . . .

The output sequence (output S_3) is therefore. 00111010 ....Which repeats itself with period $2^3–1 = 7$ (n=3). Maximal length codes are commonly used PN codes In binary shift register, the maximum length sequence is

$$N = 2^m - 1$$

chips, where **m** is the number of stages of flip-flops in the shift register.



Linear Feedback Shift Register with modulo two adder

At each clock pulse

- Contents of register shifts one bit right.
- Contents of required stages are modulo 2 added and fed back to input.

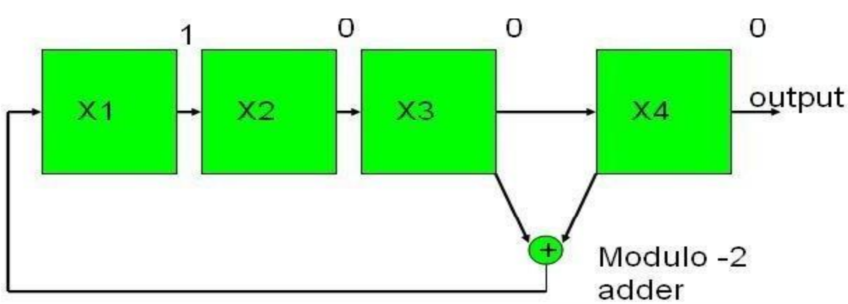


Fig: 5.5Initial stages of Shift registers 1000

Let initial status of shift register be 1000

1	0	0	0
0	1	0	0
0	0	1	0
1	0	0	1
1	1	0	0
0	1	1	0
1	0	1	1
0	1	0	1
1	0	1	0
1	1	0	1
1	1	1	0
1	1	1	1
0	1	1	1
0	0	1	1
0	0	0	1
1	0	0	0

- We can see for shift Register of length m=4. .At each clock the change in state of flip-flop is shown.
- Feed back function is modulo two of X_3 and X_4.
- After 15 clock pulses the sequence repeats. Output sequence is 0 0 0 1 0 0 1 1 0 1 0 1 1 1 1

Properties of PN Sequence

Randomness of PN sequence is tested byfollowing properties

1. Balance property
2. Run length property
3. Autocorrelation property

1. Balance property

In each Period of the sequence , number of binary ones differ from binary zeros by at most one digit.

Consider output of shift register 0 0 0 1 0 0 1 1 0 1 0 1 1 1 1 Seven zeros and eight ones -meets balance condition.

2. Run length property

Among the runs of ones and zeros in each period, it is desirable that about one half the runs of each type are of length 1, one- fourth are of length 2 and one-eighth are of length 3 and so-on.

Consider output of shift register
Number of runs =8

0 0 0 ______ 1 ___ 0 0 ____ 1 1 ____ 0 ___ 1 ___ 0 ___ 1 1 1 1

3 1 2 2 1 1 1 4

3. Auto correlation property

Auto correlation function of a maximal length sequence is periodic and binary valued. Autocorrelation sequence of binary sequence in polar format is given by

$$R_c\,k = \frac{1}{N}\sum_{n=1}^{N} c_n\, c_{n-k}$$

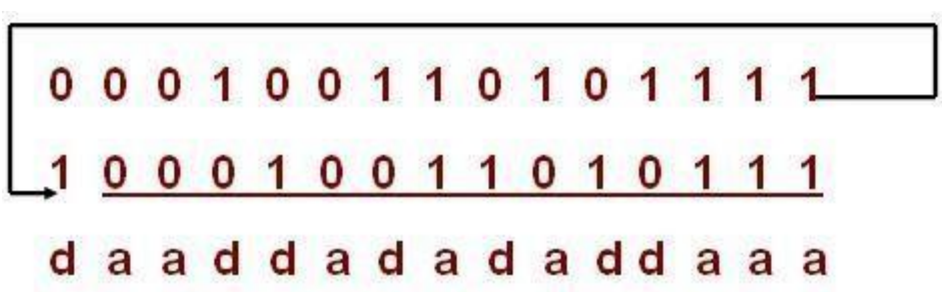
Where N is length or the period of the sequence, k is the lag of auto correlation function.

$$R_c = \begin{cases} 1 & f\ k = 1N \\ -\frac{1}{N} & if\ \neq 1N \end{cases}$$

Where 1 is any Integer. We can also state the auto correlation function is

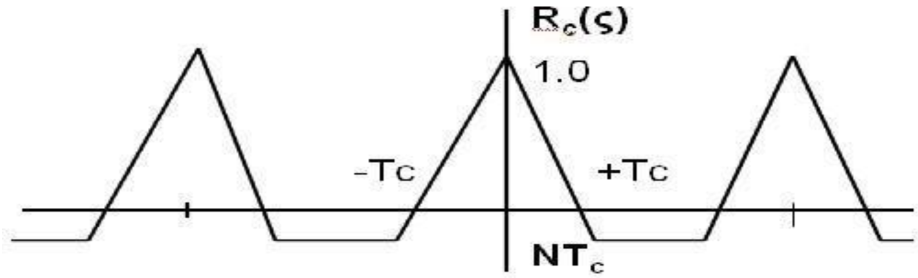
$$R_c\ k = \frac{1}{N}$$

{ No. of agreements – No. of disagreements in comparison of one full period }
Consider output of shift register for l=1



$$R_c\,k = \frac{1}{15}\quad 7 - 8 = -\frac{1}{15}$$

Yields PN autocorrelation as



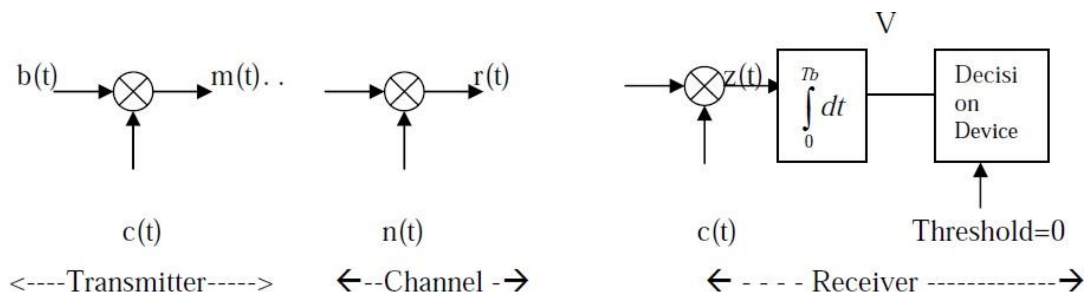
PN autocorrelation function.

Range of PN Sequence Lengths

Length 0f Shift Register, m	**PN Sequence Length,**
7	**127**
8	**255**
9	**511**
10	**1023**
11	**2047**
12	**4095**
13	**8191**
17	**131071**
19	**524287**

Notion of Spread Spectrum:

An important attribute of Spread Spectrum modulation is that it can provide protection against externally generated interfacing signals with finite power. Protection against jamming (interfacing) waveforms is provided by purposely making the information – bearing signal occupy a BW far in excess of the minimum BW necessary to transmit it. This has the effect of making the transmitted signal a noise like appearance so as to blend into the background. Therefore Spread Spectrum is a method of 'camouflaging' the information – bearing signal.



Let { bK} denotes a binary data sequence.

{ cK } denotes a PN sequence.

b(t) and c(t) denotes their NRZ polar representation respectively.

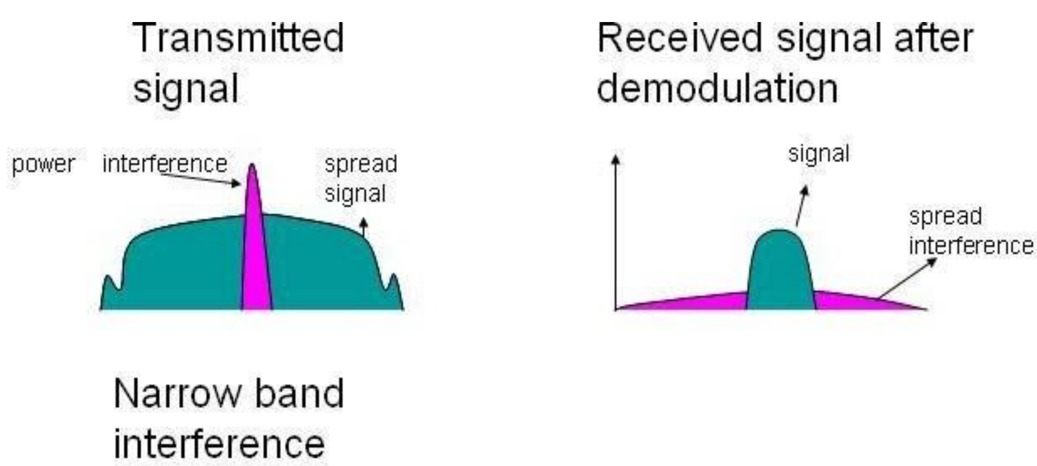
The desired modulation is achieved by applying the data signal b(t) and PN signal c(t) to a product modulator or multiplier. If the message signal b(t) is narrowband and the PN sequence signal c(t) is wide band, the product signal m(t) is also wide band. The PN sequence performs the role of a '**<u>SpreadingCode</u>**".

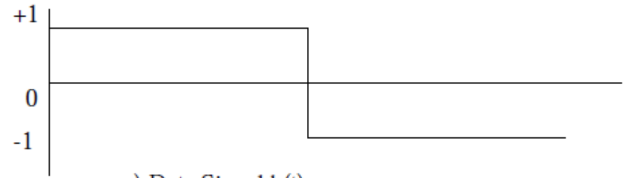
For base band transmission, the product signal m(t) represents the transmitted signal. Therefore m(t) = c(t).b(t)

The received signal r(t) consists of the transmitted signal m(t) plus an additive interference noise n(t), Hence

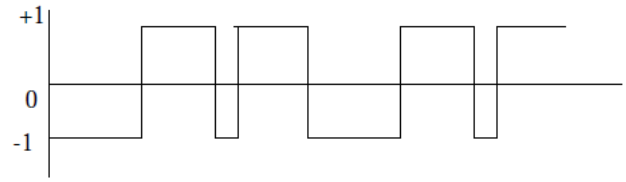
r(t) = m(t) + n(t)

= c(t).b(t) + n(t)

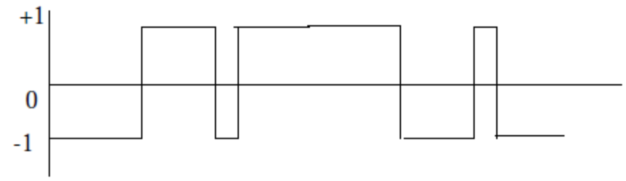




a) Data Signal b(t)



b)Spreading Code c(t)



c)Product signal or base band transmitted signal m(t)

To recover the original message signal b(t), the received signal r(t) is applied to a demodulator that consists of a multiplier followed by an integrator and a decision device. The multiplier is supplied with a locally generated PN sequence that is exact replica of that used in the transmitter. The multiplier output is given by

$$Z(t) = r(t).c(t)$$

$$= [b(t) * c(t) + n(t)]\, c(t) = c^2(t).b(t) + c(t).n(t)$$

The data signal b(t) is multiplied twice by the PN signal c(t), where as unwanted signal n(t) is multiplied only once. But $c^2(t) = 1$, hence the above equation reduces to

$$Z(t) = b(t) + c(t).n(t)$$

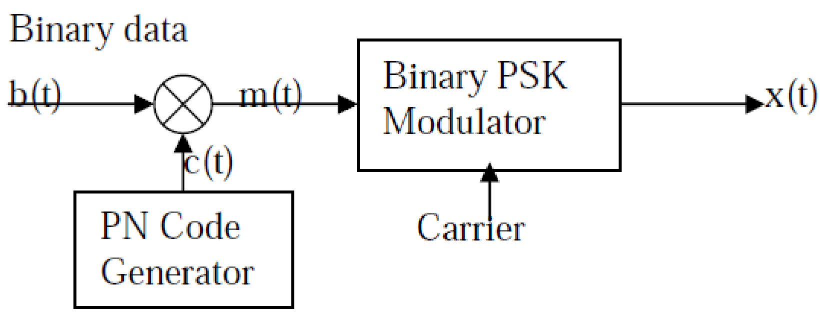
Now the data component b(t) is narrowband, where as the spurious component c(t)n(t) is wide band. Hence by applying the multiplier output to a base band (low pass) filter most of the power in the spurious component c(t)n(t) is filtered out. Thus the effect of the interference n(t) is thus significantly reduced at the receiver output.

The integration is carried out for the bit interval $0 \le t \le T_b$ to provide the sample value V. Finally, a decision is made by the receiver.

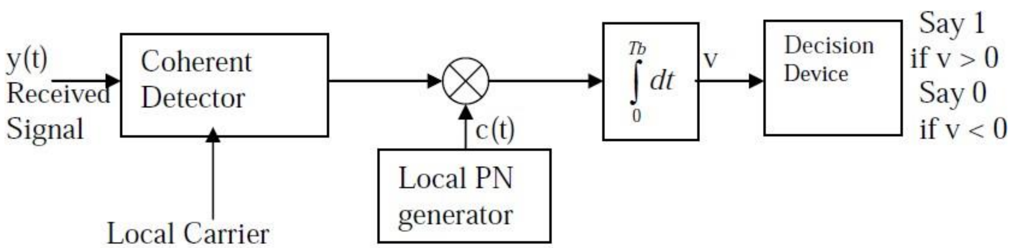
If V > Threshold Value ‘0’, say binary symbol ‘1’ If V < Threshold Value ‘0’, say binary symbol ‘0’

Direct – Sequence Spread Spectrum with coherent binary Phase shift

Keying:-



a) Transmitter



b) Receiver

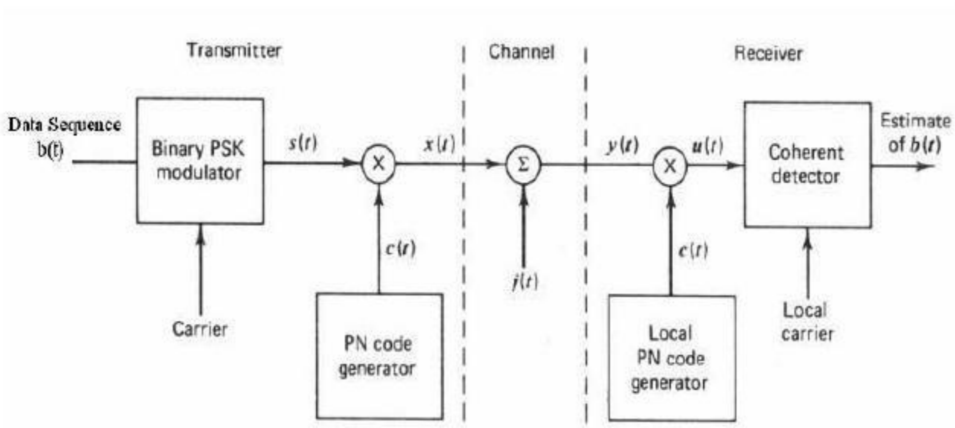


Fig: model of direct – sequence spread binary PSK system(alternative form)

To provide band pass transmission, the base band data sequence is multiplied by a Carrier by means of shift keying. Normally binary phase shift keying (PSK) is used because of its advantages. The transmitter first converts the incoming binary data sequence $\{b_k\}$ into an NRZ waveform b(t), which is followed by two stages of modulation.

The first stage consists of a multiplier with data signal b(t) and the PN signal c(t) as inputs. The output of multiplier is m(t) is a wideband signal. Thus a narrow – band data sequence is transformed into a noise like wide band signal.

The second stage consists of a binary Phase Shift Keying (PSK) modulator. Which converts base band signal m(t) into band pass signal x(t). The transmitted signal x(t) is thus a direct – sequence spread binary PSK signal. The phase modulation θ(t) of x(t) has one of the two values '0' and 'π' (180^{0}) depending upon the polarity of the message signal b(t) and PN signal c(t) at time t.

Polarity of PN & Polarity of PN signal both +, + or - - Phase '0'

Polarity of PN & Polarity of PN signal both +, - or - + Phase 'π'

		Polarity of data sequence b(t)	
		+	-
Polarity of PN sequence C(t)	+	**0**	π
	-	π	**0**

The receiver consists of two stages of demodulation.

In the first stage the received signal y(t) and a locally generated carrier are applied to a coherent detector (a product modulator followed by a low pass filter), Which converts band pass signal into base band signal.

The second stage of demodulation performs Spectrum despreading by multiplying the output of low-pass filter by a locally generated replica of the PN signal c(t), followed by integration over a bit interval T_b and finally a decision device is used to get binary sequence.

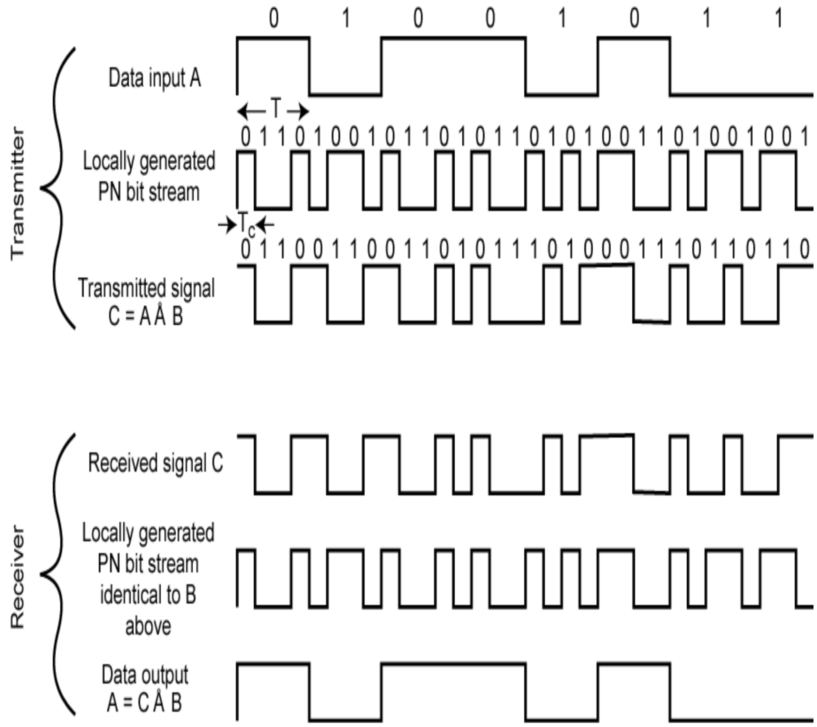


Fig : Direct Sequence Spread Spectrum Example

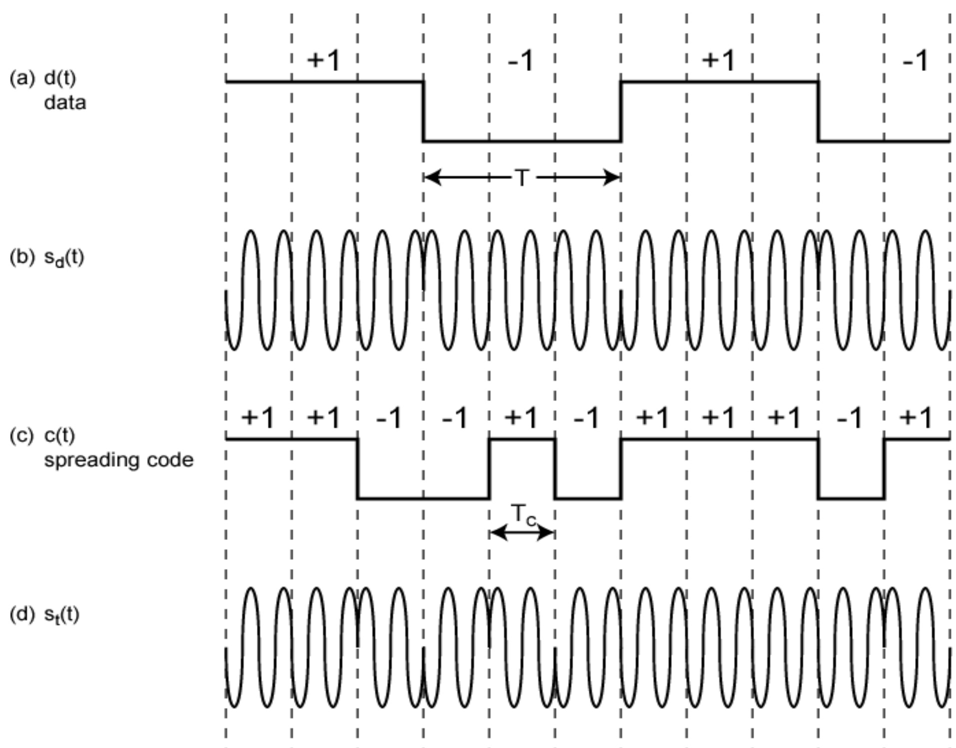


Fig : 5.5Direct Sequence Spread Spectrum Using BPSK Example

Signal Space Dimensionality and Processing Gain

- Fundamental issue in SS systems is how much protection spreading can provide against interference.
- SS technique distribute low dimensional signal into large dimensional signal space (hide the signal).
- Jammer has only one option; to jam the entire space with fixed total power or to jam portion of signal space with large power.

Consider set of orthonormal basis functions;

$$\varphi_k(t) = \begin{cases} \sqrt{\frac{2}{T_c}} \cos 2\Pi f_c t & kT_c \le t \le (t+1) T_c \\ 0 & \text{Otherwise} \end{cases}$$

$$\varphi_k(t) = \begin{cases} \sqrt{\frac{2}{T_c}} \sin 2\Pi f_c t & kT_c \le t \le (t+1) T_c \\ 0 & \text{Otherwise} \end{cases} \quad = 0,1 \ldots \ldots \ldots \ldots \ldots N-1$$

Where $\mathbf{T_C}$ is chip duration, **N** is number of chips per bit. Transmitted

signal x(t) for the interval of an information bit is

$$x(t) = c(t)s(t)$$

$$\varphi_k(t) = \pm \sqrt{\frac{2}{T_c}}\, c(t) \cos 2\Pi f_c t$$

$$\sqrt{E_b} \quad N-1$$

$$\varphi_k \quad t \quad = \pm \quad \frac{}{N} \quad c_k\varphi_k\ (t) \qquad\qquad 0 \le \quad \le T_b$$
$$k=0$$

where, Eb is signal energy per bit.

PN Code sequence { c0, c1, ……cN-1} with ck= ± 1, Transmitted signal x(t) is therefore N dimensional and requires N orthonormal functions to represent it. j(t) represent interfering signal (jammer). As said jammer tries to places all its available energy in exactly same N dimension signal space. But jammer has no knowledge of signal phase. Hence tries to place equal energy in two phase coordinates that is cosine and sine.

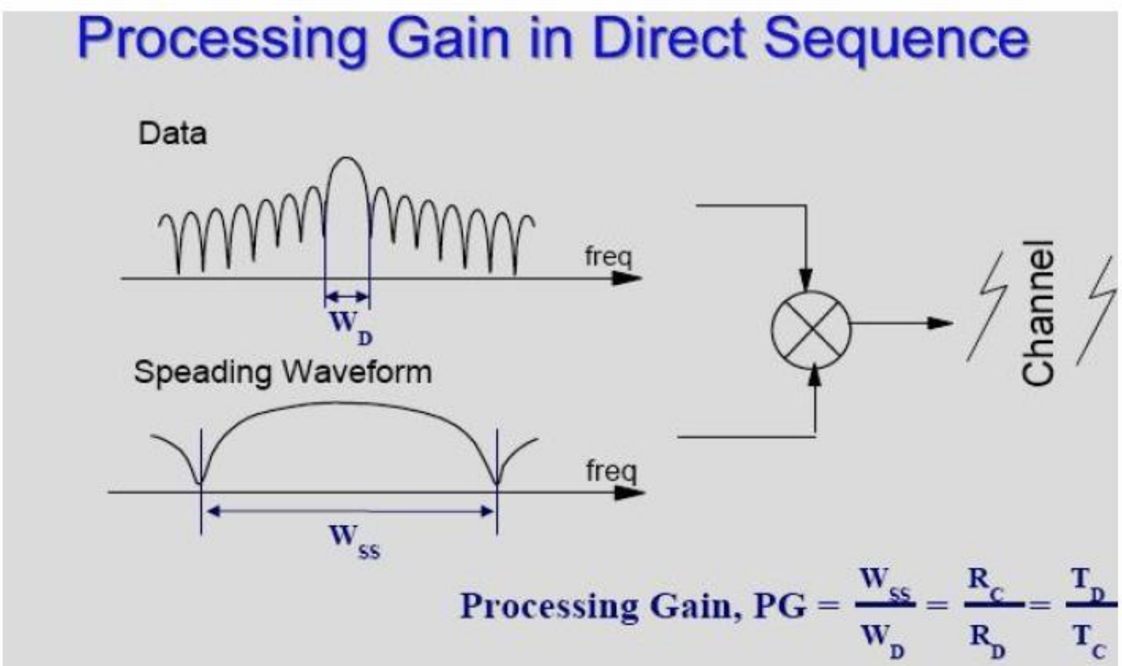
$$(SNR)_0$$

Expressing SNR in decibels

$$10log_{10}(SNR)_0 = 10log_{10}(SNR)_i + 3 + 10log_{10}\ PG\ ,\ dB$$

Where $P = \frac{Tb}{Tc}$

3db term on right side accounts for gain in SNR due to coherent detection. Last term accounts for gain in SNR by use of spread spectrum. **PG is called Processing Gain.**



The bandwidth of PN sequence c(t) , of main lobe is W_c

To calculate probability of error, we consider output component **v** of coherent detector as sample value of random variable

$$V = \pm \sqrt{E_b} + V_{cj}$$

E_b is signal energy per bit and V_{cj} is noise component

Decision rule is, if detector output exceeds a threshold of zero volts; received bit is symbol 1 else decision is favored for zero.

- Average probability of error P_e is nothing but conditional probability which depends on random variable V_{cj}.
- As a result receiver makes decision in favor of symbol 1 when symbol 0 transmitted and vice versa
- Random variable Vcj is sum of N such random variables. Hence for Large N it can assume Gaussian distribution.
- As mean and variance has already been discussed , zero mean and variance

Probability of error can be calculated from simple formula for DS/BPSK system

<u>Example1</u>

A pseudo random sequence is generated using a feed back shift register of length m=4. The chip rate is 107 chips per second. Find the following a) PN sequence length b) Chip duration of PN sequence c) PN sequence period

<u>Solution</u>

a) Length of PN sequence N = 2^m-1= 2^4-1 =15
b) Chip duration T_c = 1/chip rate =1/107 = 0.1µsec
c) PN sequence period T = NT_c
= 15 x 0.1µ sec = 1.5µ sec

<u>Example2</u>

A direct sequence spread binary phase shift keying system uses a feedback shift register of length 19 for the generation of PN sequence. Calculate the processing gain of the system.

Solution

Given length of shift register = m =19 Therefore
length of PN sequence N = $2^m - 1$

$= 2^{19} - 1$

Processing gain PG = T_b/T_c =N in db =$10\log_{10}N = 10 \log_{10} (2^{19})$ = 57db

Example3

A Spread spectrum communication system has the following parameters. Information bit duration Tb = 1.024 msecs and PN chip duration of 1µsecs. The average probability of error of system is not to exceed 10^{-5}. calculate a) Length of shift register b) Processing gain c) jamming margin

Solution

Processing gain PG =N= Tb/Tc =1024
corresponding length of shift register m = 10

In case of coherent BPSK For Probability of error 10^{-5}. [Referring to error function table] E_b/N_0 =10.8

Therefore jamming margin

$$jamming\ margin(\text{dB}) = Processing\ gain(\text{dB}) - 10\ log_{10} \left(\frac{E_b}{N_0}\right)_{min}$$

$$jamming\ margin_{dB} = 10\ log_{10}\ PG_{dB} - 10\ log_{10} \frac{E_b}{N_0}$$

$$min$$

$$jamming\ margin_{dB} = 10\ log_{10}1024 - 10\ log_{10}10.8$$

$$jamming\ margin_{dB} = 30.10 - 10.33 = 19.8\ dB$$

Frequency – Hop Spread Spectrum:

In a frequency – hop Spread – Spectrum technique, the spectrum of data modulated carrier is widened by changing the carrier frequency in a pseudo – random manner. The type of spread – spectrum in which the carrier hops randomly form one frequency to another is called **Frequency – Hop (FH) Spread Spectrum.**

Since frequency hopping does not covers the entire spread spectrum instantaneously. We are led to consider the rate at which the hop occurs. Depending upon this we have two types of frequency hop.

1. Slow frequency hopping:- In which the symbol rate R_s of the MFSK signal is an integer multiple of the hop rate R_h. That is several symbols are transmitted on each frequency hop.
2. Fast – Frequency hopping:- In which the hop rate R_h is an integral multiple of the MFSK symbol rate R_s. That is the carrier frequency will hoop several times during the transmission of one symbol. A common modulation format for frequency hopping system is that of M- ary frequency – shift – keying (MFSK).

Slow frequency hopping:-

Fig Shows the block diagram of an FH / MFSK transmitter, which involves frequency modulation followed by mixing.

The incoming binary data are applied to an M-ary FSK modulator. The resulting modulated wave and the output from a digital frequency synthesizer are then applied to a mixer that consists of a multiplier followed by a band – pass filter. The filter is designed to select the sum frequency component resulting from the multiplication process as the transmitted signal. An ‘k’ bit segments of a PN sequence drive the frequency synthesizer, which enables the carrier frequency to hop over 2^n distinct values. Since frequency synthesizers are unable to maintain phase coherence over successive hops, most frequency hops spread spectrum communication system use non coherent M-ary modulation system.

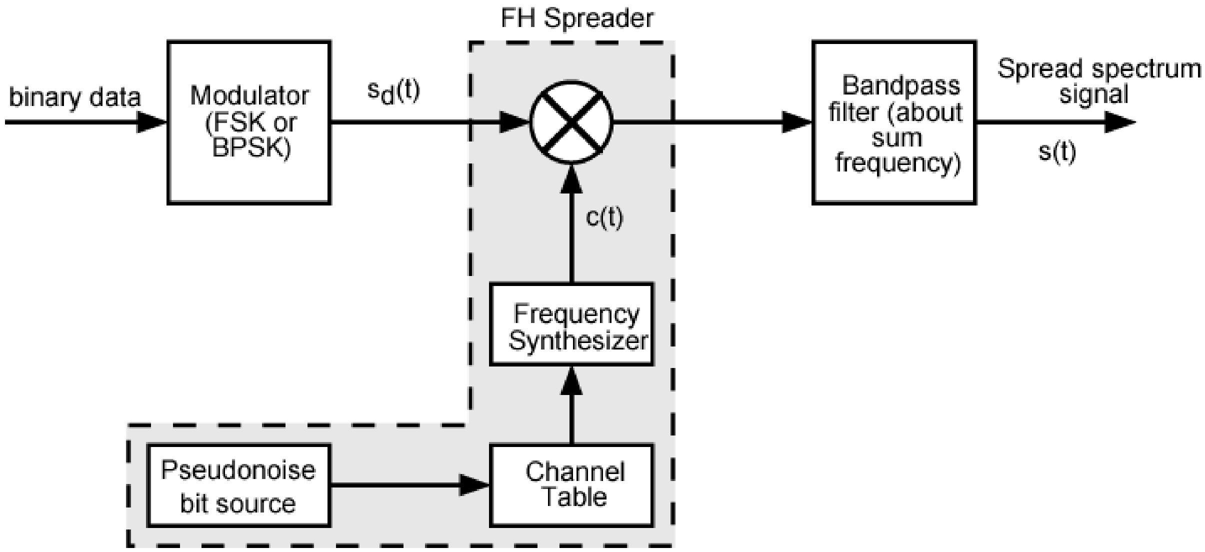


Fig 5.6:- Frequency hop spread transmitter

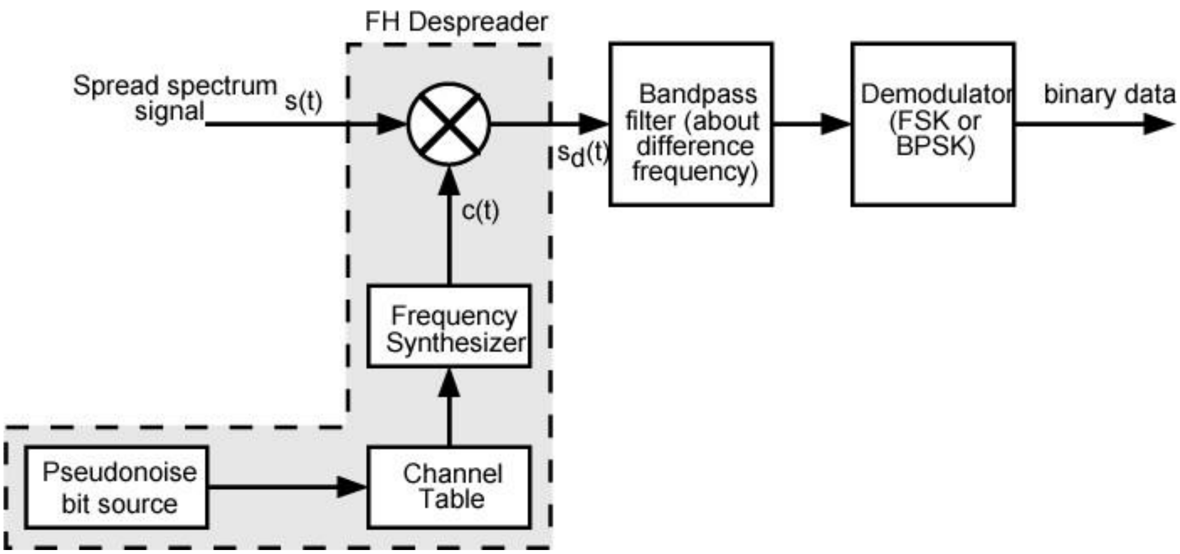


Fig 5.7 :- Frequency hop spread receiver

In the receiver the frequency hopping is first removed by mixing the received signal with the output of a local frequency synthesizer that is synchronized with the transmitter. The resulting output is then band pass filtered and subsequently processed by a non coherent M-ary FSK demodulator. To implement this M-ary detector, a bank of M non coherent matched filters, each of which is matched to one of the MFSK tones is used. By selecting the largest filtered output, the original transmitted signal is estimated.

An individual FH / MFSK tone of shortest duration is referred as a chip. The chip rate R_c for an FH / MFSK system is defined by

$R_c = Max(R_h, R_s)$

Where R_h is the hop rate and R_s is Symbol Rate

In a slow rate frequency hopping multiple symbols are transmitted per hop. Hence each symbol of a slow FH / MFSK signal is a chip. The bit rate R_b of the incoming binary data. The symbol rate R_s of the MFSK signal, the chip rate R_c and the hop rate R_n are related by

$R_c = R_s = R_b / k \geq R_h$ where

$k = \log_2 M$

Fast frequency hopping:-

A fast FH / MFSK system differs from a slow FH / MFSK system in that there are multiple hops per m-ary symbol. Hence in a fast FH / MFSK system each hop is a chip.

Fast Frequency Hopping	Slow Frequency Hopping
Several frequency hops Per modulation	Several modulation symbols per hop
Shortest uninterrupted waveform in the system is that of hop	Shortest uninterrupted waveform in the system is that of data symbol
Chip duration =hop duration	Chip duration=bit duration.

The following figure illustrates the variation of the frequency of a slow FH/MFSK signal with time for one complete period of the PN sequence. The period of the PN sequence is $2^4 - 1 = 15$.

The FH/MFSK signal has the following parameters:

Number of bits per MFSK symbol K = 2. Number of MFSK tones $M = 2^K = 4$

Length of PN segment per hop k = 3; Total number of frequency hops $2^k = 8$

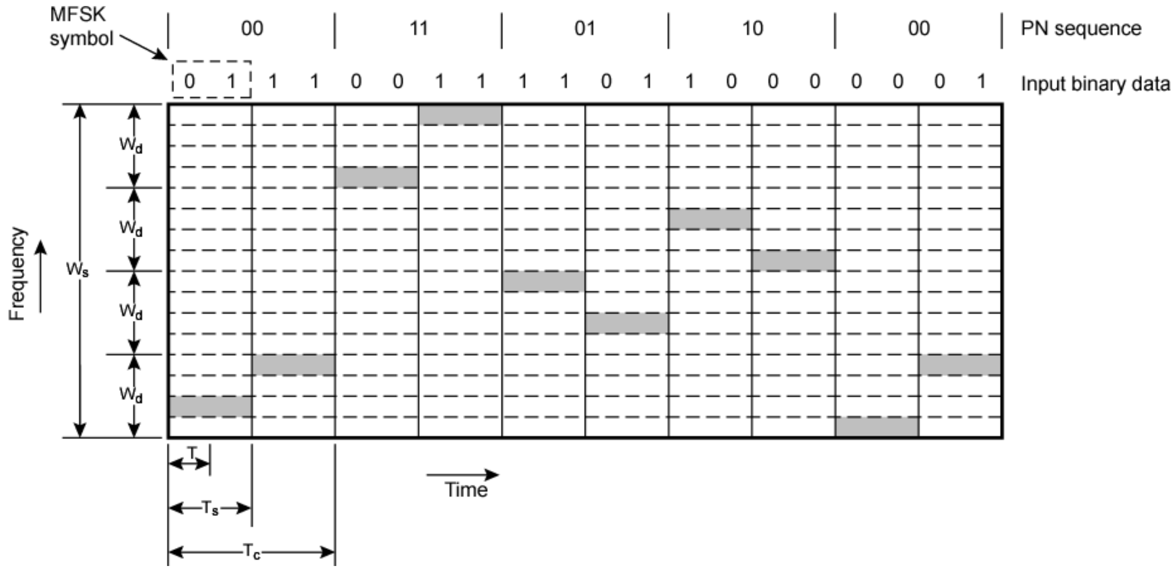


Fig.5.8 Slow frequency hopping

The following figure illustrates the variation of the transmitted frequency of a fast FH/MFSK signal with time.

The signal has the following parameters:

Number of bits per MFSK symbol K = 2. Number of MFSK tones M = 2^K = 4 Length of PN segment per hop k = 3; Total number of frequency hops 2^k = 8

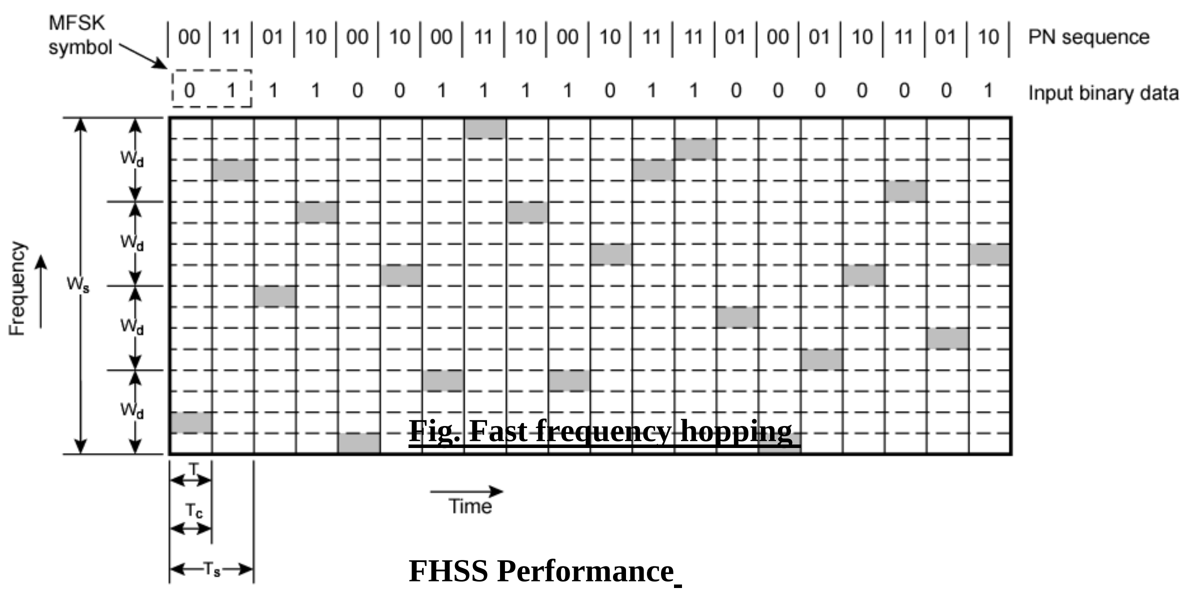


Fig. Fast frequency hopping

FHSS Performance

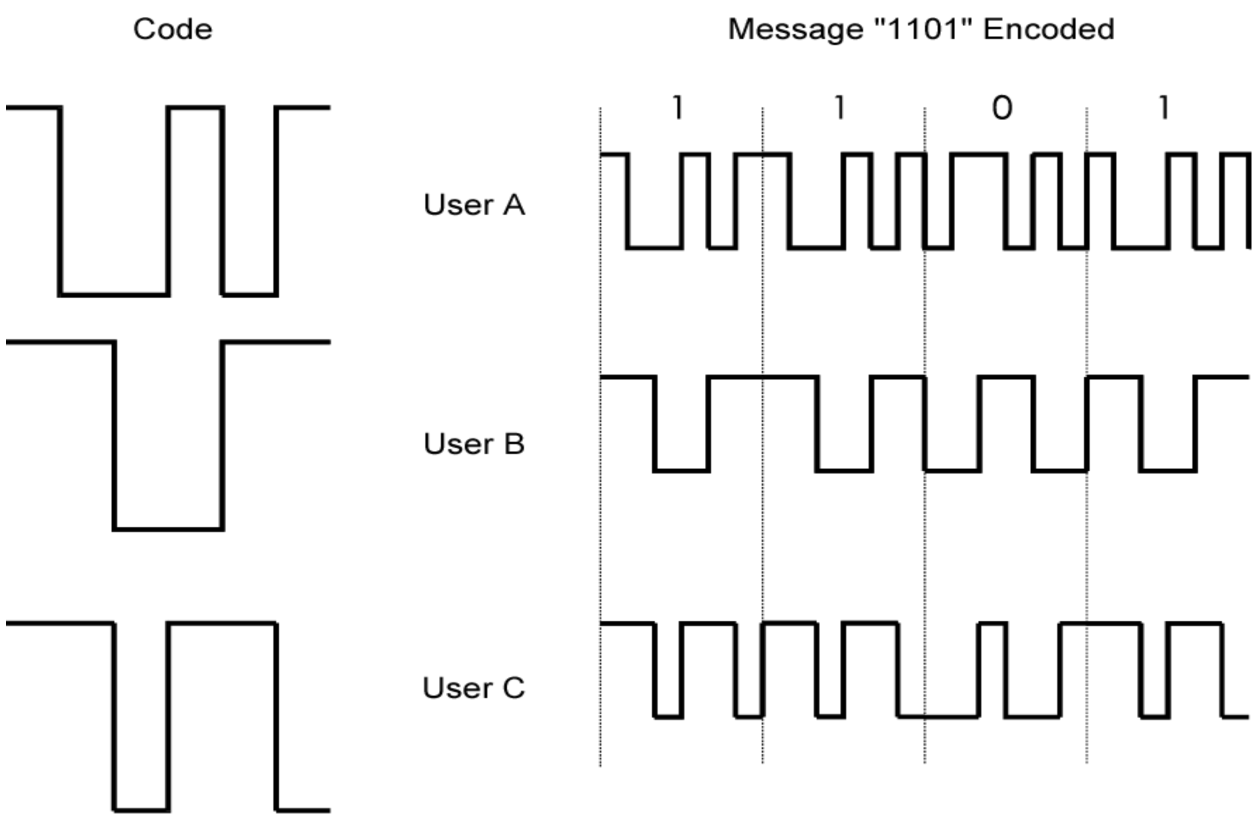
Considerations:

- Typically large number of frequencies used
  - Improved resistance to jamming

Code Division Multiple Access (CDMA):

- Multiplexing Technique used with spread spectrum
- Start with data signal rate *D*
  - Called bit data rate
- Break each bit into *k* chips according to fixed pattern specific to each user
  - User's code
- New channel has chip data rate *kD* chips per second
- E.g. *k*=6, three users (A,B,C) communicating with base receiver R
- Code for A = <1,-1,-1,1,-1,1>
- Code for B = <1,1,-1,-1,1,1>
- Code for C = <1,1,-1,1,1,-1>

CDMA Example:



- Consider A communicating with base
- Base knows A’s code
- Assume communication already synchronized
- A wants to send a 1
  - Send chip pattern <1,-1,-1,1,-1,1>
    - A’s code
- A wants to send 0
  - Send chip[pattern <-1,1,1,-1,1,-1>
    - Complement of A’s code
- Decoder ignores other sources when using A’s code to decode
  - Orthogonal codes
  -

CDMA for DSSS:

- n users each using different orthogonal PN sequence
- Modulate each users data stream
  - Using BPSK
- Multiply by spreading code of user

CDMA in a DSSS Environment:

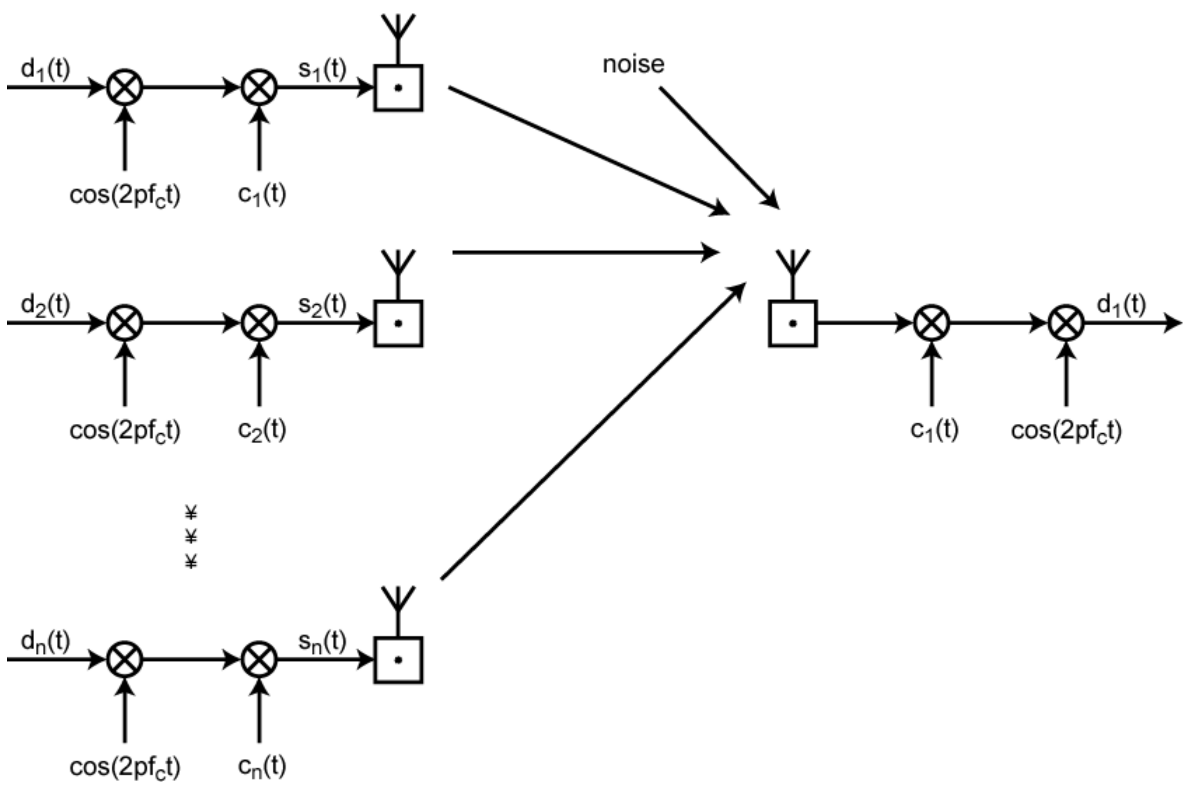


Fig 5.9 CDMA in a DSSS Environment

DS-CDMA has various features and benefits. First, DSCDMA is robust to frequency-selective fading.

Second, DSCDMA compensates for the effect of a multipath propagation at the receiver by exploiting rake filters, which can collect the transmitted energy spread over multiple rays. Third, DS-CDMA also allows receivers to distinguish among signals simultaneously transmitted by multiple devices. Because of these reasons, CDMA increases the number of to reuse channel and decrease the number of the packet retransmission. Therefore CDMA results in decreased energy consumption and increased network throughputs. Finally, DS-CDMA have an excellent security, noise/jamming immunity

TEXT / REFERENCE BOOKS

1. Simon Haykins, "Communication Systems" John Wiley, 5th Edition, March 2009.
2. Taub. HDL Schilling, G Saha, "Principles of Communication"3rd edition,2007.
3. John G. Proakis, Masoud Salehi, "Digital Communication", McGraw Hill 5th edition November 6, 2007.
4. Bernard Sklar, "Digital Communication, Fundamentals and Application", Pearson Education Asia, 2nd Edition, Jan. 21, 2001.

QUESTION BANK
UNIT V

PART-A

1. Give the beneficial attributes of spread spectrum systems.
2. Write the properties of pseudo noise (PN) sequences.
3. What advantage is there in making the message bit rate a sub-multiple of the PN sequence rate ?
4. Given the data rate is 100Kbps and the chip rate is 1Mbps, calculate the processing gain.
5. What is pulse jamming?
6. Compare frequency hopping and direct sequence spread spectrum.
7. Define processing gain?
8. Define run length property of PN sequence.
9. What do you mean by slow frequency hopping?
10. Give examples of real life wireless systems that employ DSSS or FHSS.

PART-B

1. Discuss about the potential applications of spread spectrum techniques in wireless communications.
(16 marks)

2. a) Describe how linear feedback shift registers can be used to generate PN sequence for given degree and sequence length.
(8marks)

2.b) Discuss briefly about the importance of primitive polynomials in PN sequence generation. (8 marks)

3.a) Design a PN sequence generator circuit using shift registers for the primitive polynomial x^4+x+1 of degree 4. (8 marks)

3.b) Plot the output PN sequence of your circuit w.r.t to a clock frequency of 1KHz and initial seed value [1 0 0 0]. (8 marks)

3. Discuss the hardware system configuration of a Direct sequence spread spectrum transmitter and receiver with Binary Phase shift keying modulation scheme in detail. (16 marks)
4. Derive the expression for processing gain of Direct sequence spread spectrum system. (16 marks)
5. Explain the transmitter and receiver architecture of a frequency hopping spread spectrum system. (16 marks)
6. Illustrate fast frequency hopping and slow frequency hopping spread spectrum with examples. (16 marks)
7. Explain how spread spectrum systems can reliably receive signals that are "buried in the noise". (16 marks)
8. Assume that in order to obtain a decent BER, the required E_b/N_o is 12 dB. Furthermore, assume that the duration of the information bit T_b is 0.5 ms. Given that the required jamming margin is approximately 21.11 dB, what is the required processing gain? What is the resulting chip-duration? What is the required length of the shift register that produces the corresponding maximum length PN sequence? (16 marks)