UNIT-1

source coding systems

- Communication system converts information into electrical electromagnetic/optical signals appropriate for the transmission medium.
- Analog systems convert analog message into signals that can propagate through the channel.
- Digital systems convert bits(digits, symbols) into signals
  - Computers naturally generate information as characters/bits
  - Most information can be converted into bits
  - Analog signals converted to bits by sampling and quantizing (A/D conversion)
- Digital techniques need to distinguish between discrete symbols allowing regeneration versus amplification
- Good processing techniques are available for digital signals, such as medium.
  - Data compression (or source coding)
  - Error Correction (or channel coding)(A/D conversion)
  - Equalization
  - Security
  - Easy to mix signals and data using digital techniques

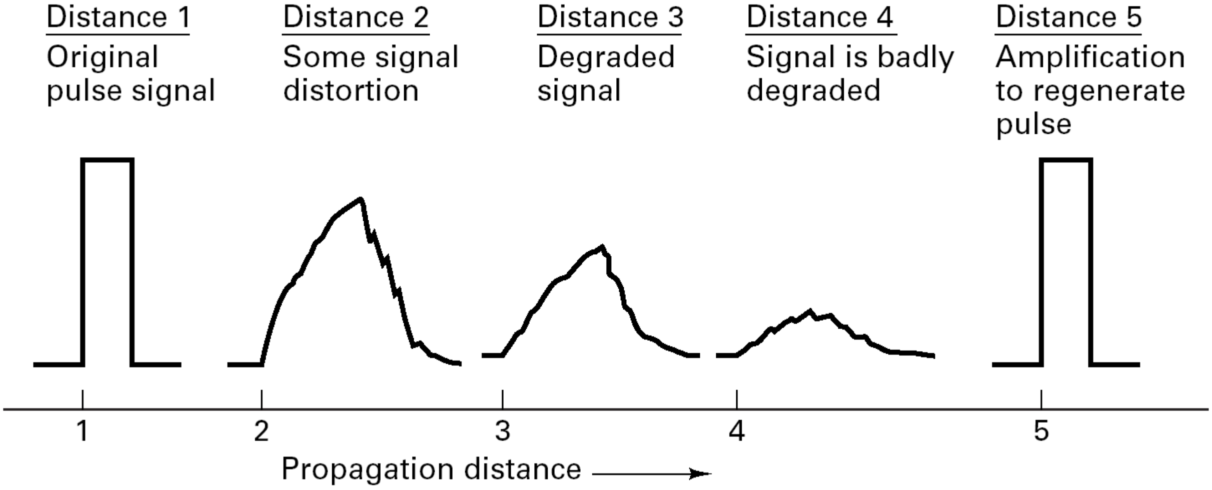


Figure 1.1 Pulse degradation and regeneration.

Elements of Digital Communication Systems:

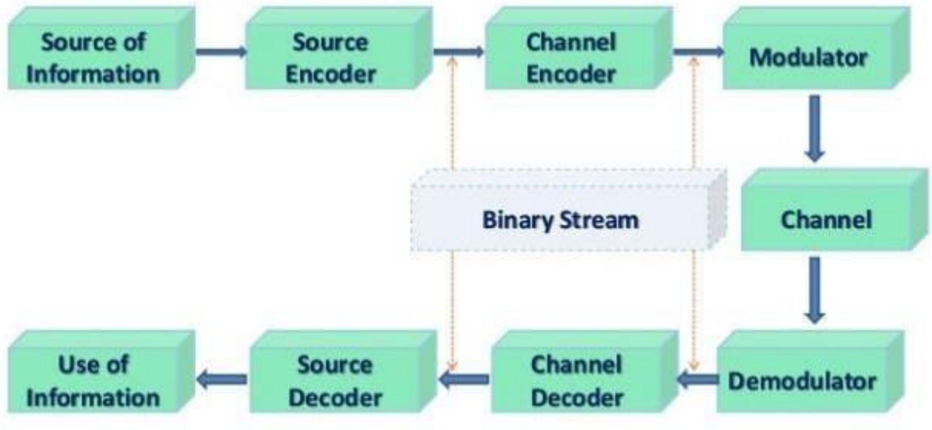


Fig. 1 Elements of Digital Communication Systems

1. **Information Source and Input Transducer:**

   The source of information can be analog or digital, e.g. analog: audio or video signal, digital: like teletype signal. In digital communication the signal produced by this source is converted into digital signal which consists of 1's and 0's. For this we need a source encoder.

2. **Source Encoder:**

   In digital communication we convert the signal from source into digital signal as mentioned above. The point to remember is we should like to use as few binary digits as possible to represent the signal. In such a way this efficient representation of the source output results in little or no redundancy. This sequence of binary digits is called ***information sequence.***

   Source Encoding or Data Compression: the process of efficiently converting the output of whether analog or digital source into a sequence of binary digits is known as source encoding.

3. **Channel Encoder:**

   The information sequence is passed through the channel encoder. The purpose of the channel encoder is to introduce, in controlled manner, some redundancy in the binary information sequence that can be used at the receiver to overcome the effects of noise and interference encountered in the transmission on the signal through the channel.

   For example take k bits of the information sequence and map that k bits to unique n bit sequence called code word. The amount of redundancy introduced is measured by the ratio *n/k* and the reciprocal of this ratio (k/n) is known as *rate of code or code rate.*

4. **Digital Modulator:**

   The binary sequence is passed to digital modulator which in turns convert the sequence into electric signals so that we can transmit them on channel (we will see channel later). The digital modulator maps the binary sequences into signal wave forms , for example if we represent 1 by sin x and 0 by cos x then we will transmit sin x for 1 and cos x for 0. (a case similar to BPSK)

5. **Channel:**

   The communication channel is the physical medium that is used for transmitting signals from transmitter to receiver. In wireless system, this channel consists of atmosphere , for traditional telephony, this channel is wired , there are optical channels, under water acoustic channels etc.We further discriminate this channels on the basis of their property and characteristics, like AWGN channel etc.

6. **Digital Demodulator:**

   The digital demodulator processes the channel corrupted transmitted waveform and reduces the waveform to the sequence of numbers that represents estimates of the transmitted data symbols.

7. **Channel Decoder:**

   This sequence of numbers then passed through the channel decoder which attempts to reconstruct the original information sequence from the knowledge of the code used by the channel encoder and the redundancy contained in the received data

Note: The average probability of a bit error at the output of the decoder is a measure of the performance of the demodulator – decoder combination.

8. **Source Decoder:**

   At the end, if an analog signal is desired then source decoder tries to decode the sequence from the knowledge of the encoding algorithm. And which results in the approximate replica of the input at the transmitter end.

9. **Output Transducer:**

Finally we get the desired signal in desired format analog or digital.

Advantages of digital communication:

- **Can withstand channel noise and distortion much better as long as the noise and the distortion are within limits.**
- **Regenerative repeaters prevent accumulation of noise along the path.**
- **Digital hardware implementation is flexible.**
- **Digital signals can be coded to yield extremely low error rates, high fidelity and well as privacy.**
- **Digital communication is inherently more efficient than analog in realizing the exchange of SNR for bandwidth.**
- **It is easier and more efficient to multiplex several digital signals.**
- **Digital signal storage is relatively easy and inexpensive.**
- **Reproduction with digital messages is extremely reliable without deterioration.**
- **The cost of digital hardware continues to halve every two or three years, while performance or capacity doubles over the same time period.**

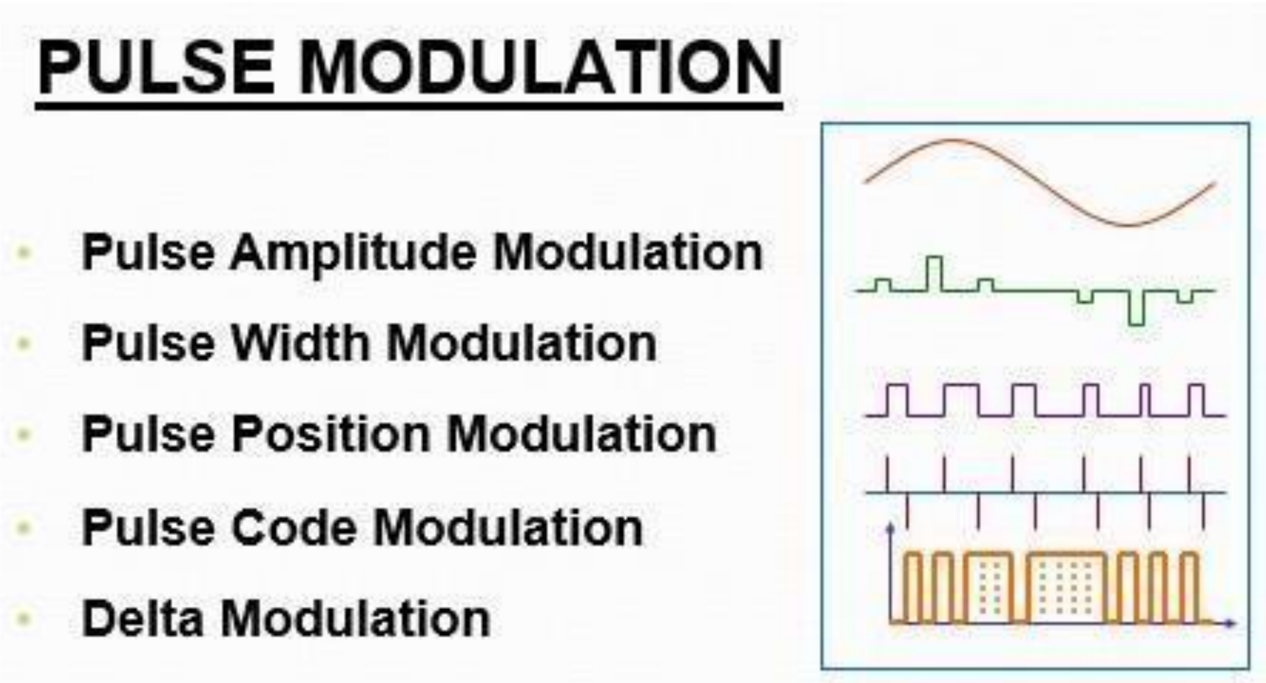
Disadvantages

- **TDM digital transmission is not compatible with the FDM**
- **A Digital system requires large bandwidth.**

Introduction to Pulse Modulation

What is the need for Pulse Modulation?

- Many Signals in Modern Communication Systems are digital
- Also, analog signals are transmitted digitally.
- Reduced distortion and improvement in signal to noise ratios.
- PAM, PWM, PPM, PCM and DM.
- In CW modulation schemes some parameter of modulated wave varies continuously with message.
- In Analog pulse modulation some parameter of each pulse is modulated by a particular sample value of the message.
- Pulse modulation is of two types
  - Analog Pulse Modulation
    - Pulse Amplitude Modulation (PAM)
    - Pulse width Modulation (PWM)
    - Pulse Position Modulation (PPM)
  - Digital Pulse Modulation
    - Pulse code Modulation (PCM)
    - Delta Modulation (DM)



Pulse Code Modulation:

Three steps involved in conversion of analog signal to digital signal

- Sampling
- Quantization
- Binary encoding

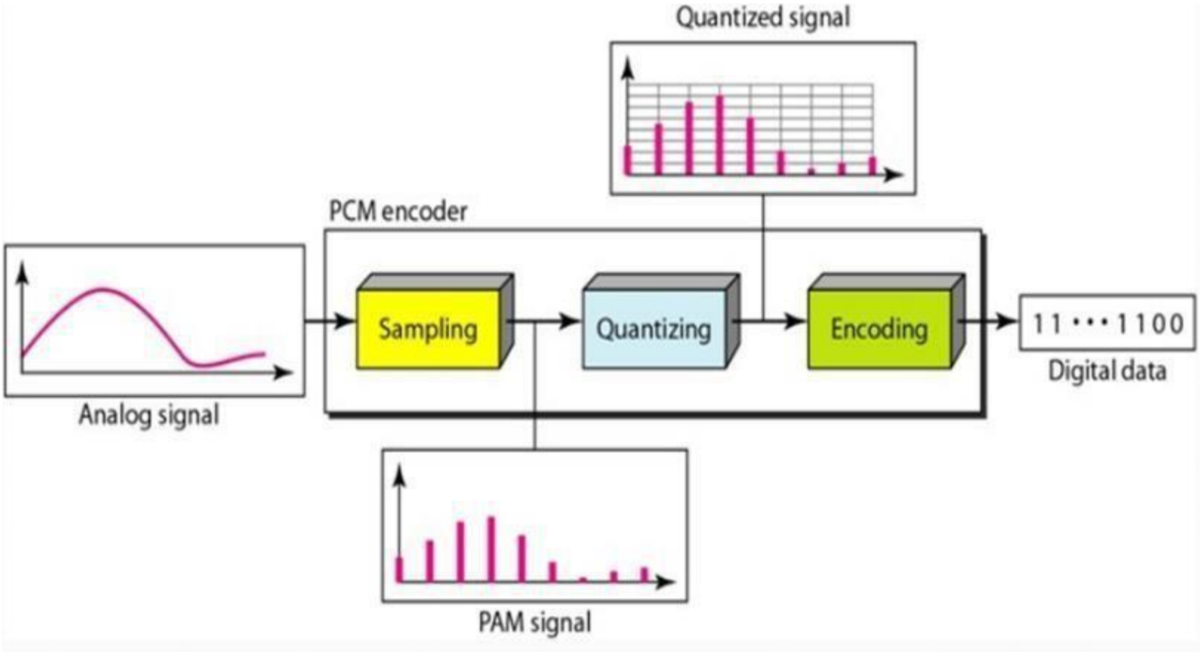


Fig. 2 Conversion of Analog Signal to Digital Signal

Note: Before sampling the signal is filtered to limit bandwidth.

<u>Elements of PCM System</u>:

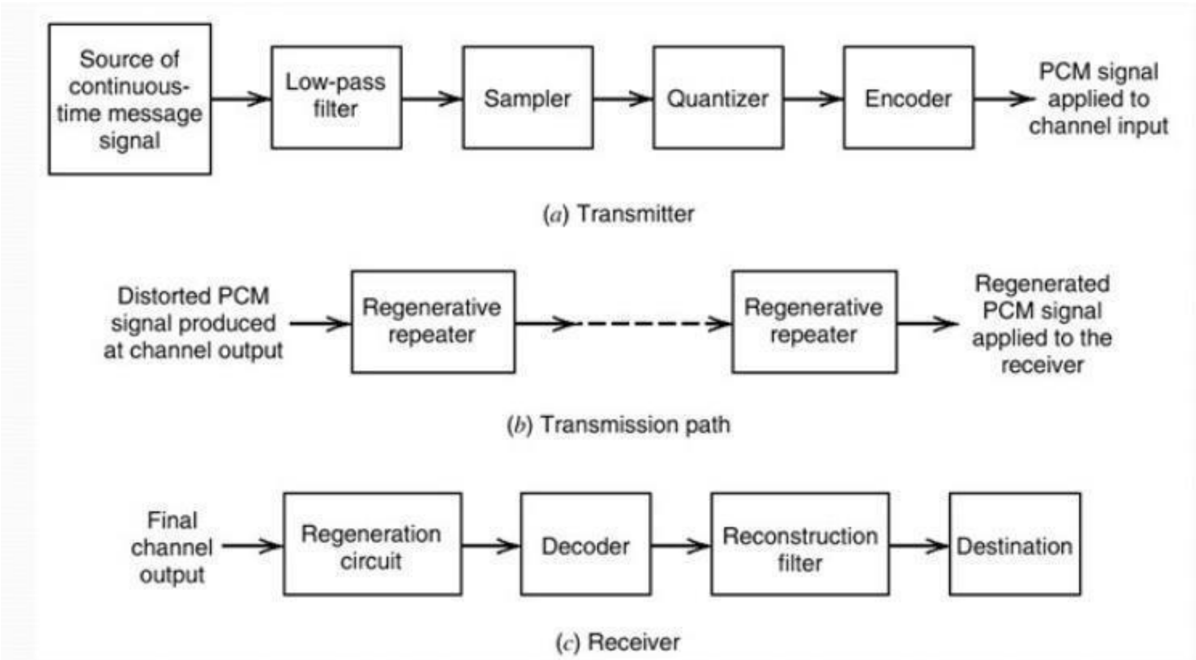


Fig. 3 Elements of PCM System

<u>Sampling:</u>

- Process of converting analog signal into discrete signal.
- Sampling is common in all pulse modulation techniques

- The signal is sampled at regular intervals such that each sample is proportional to amplitude of signal at that instant
- Analog signal is sampled every $T_s\ Secs$, called sampling interval. $f_s=1/T_S$ is called sampling rate or sampling frequency.
- $f_s=2f_m$ is Min. sampling rate called **Nyquist rate.** Sampled spectrum (ω) is repeating periodically without overlapping.
- Original spectrum is centered at $\omega=0$ and having bandwidth of ω_m. Spectrum can be recovered by passing through low pass filter with cut-off ω_m.
- For $f_s<2f_m$ sampled spectrum will overlap and cannot be recovered back. This is called **aliasing.**

Sampling methods:

- Ideal – An impulse at each sampling instant.
- Natural – A pulse of Short width with varying amplitude.
- Flat Top – Uses sample and hold, like natural but with single amplitude value.

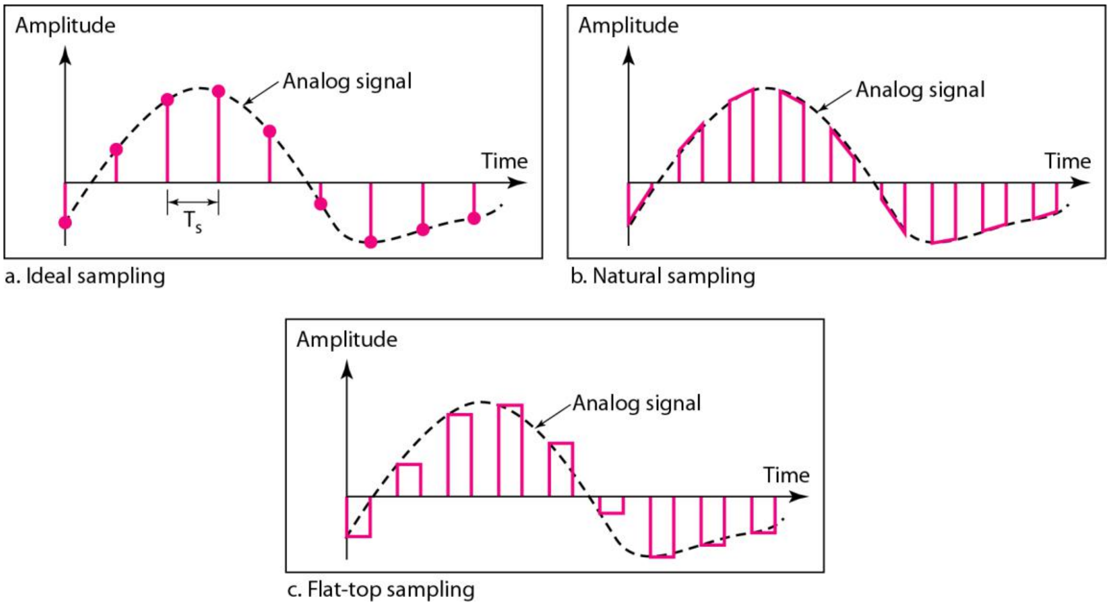


Fig. 4 Types of Sampling

Sampling of band-pass Signals:

- A band-pass signal of bandwidth $2f_m$ can be completely recovered from its samples.

  Min. sampling rate $=2\times Bandwidth$

  $=2\times 2f_m=4f_m$

- Range of minimum sampling frequencies is in the range of $2\times BW\ to\ 4\times BW$

Instantaneous Sampling or Impulse Sampling:

- Sampling function is train of spectrum remains constant impulses throughout frequency range. It is not practical.

Natural sampling:

- The spectrum is weighted by a **sinc** function.
- Amplitude of high frequency components reduces.

Flat top sampling:

- Here top of the samples remains constant.
- In the spectrum high frequency components are attenuated due sinc pulse roll off. This is known as **Aperture effect.**
- If pulse width increases aperture effect is more i.e. more attenuation of high frequency components.

Sampling Theorem:

Statement of sampling theorem

1) ***A band limited signal of finite energy, which has no frequency components higher than W Hertz, is completely described by specifying the values of the signal at instants of time separated by $\frac{1}{2W}$ seconds and***

2) ***A band limited signal of finite energy, which has no frequency components higher than W Hertz, may be completely recovered from the knowledge of its samples taken at the rate of 2W samples per second.***

The first part of above statement tells about sampling of the signal and second part tells about reconstruction of the signal. Above statement can be combined and stated alternately as follows :

A continuous time signal can be completely represented in its samples and recovered back if the sampling frequency is twice of the highest frequency content of the signal. i.e.,

$$f_s \geq 2W$$

Here f_s is the sampling frequency and

W is the higher frequency content

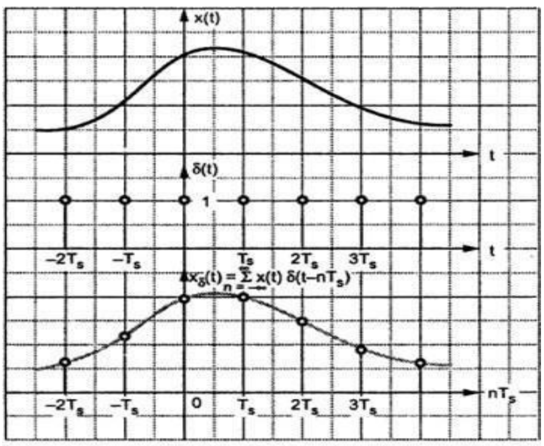


Fig. 5 CT and its DT signal

Proof of sampling theorem

There are two parts : (I) Representation of $x(t)$ in terms of its samples

(II) Reconstruction of $x(t)$ from its samples.

Part I : Representation of $x(t)$ in its samples $x(nT_s)$

Step 1 : Define $x_\delta(t)$

Step 2 : Fourier transform of $x_\delta(t)$ i.e. $X_\delta(f)$

Step 3 : Relation between $X(f)$ and $X_\delta(f)$

Step 4 : Relation between $x(t)$ and $x(nT_s)$

Step 1 : Define $x_\delta(t)$

The sampled signal $x_\delta(t)$ is given as,

$$x_\delta(t) = \sum_{n=-\infty}^{\infty} x(t)\,\delta(t-nT_s) \qquad \ldots 1$$

Here observe that $x_\delta(t)$ is the product of x_δ and impulse train $\delta(t)$ as shown in above fig In the above equation $\delta(t-nT_s)$ indicates the samples placed at $\pm T_s$, $\pm 2T_s$, $\pm 3T_s$ ... and so on.

Step 2 : FT of $x_\delta(t)$ i.e. $X_\delta(f)$

Taking FT of equation (1.3.1).

$$X_\delta(f) = FT\left\{\sum_{n=-\infty}^{\infty} x(t)\,\delta(t-nT_s)\right\}$$

$$= FT\ \{\text{Product of } x(t) \text{ and impulse train}\}$$

We know that FT of product in time domain becomes convolution in frequency domain. i.e.,

$$X_\delta(f) = FT\ \{x(t)\} * FT\ \{\delta(t-nT_s)\} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad 2$$

By definitions, $x(t) \xleftrightarrow{FT} X(f)$ and

$$\delta(t-nT_s) \xleftrightarrow{FT} f_s \sum_{n=-\infty}^{\infty} \delta(f-nf_s)$$

Hence equation (1.3.2) becomes,

$$X_\delta(f) = X(f) * f_s \sum_{n=-\infty}^{\infty} \delta(f-nf_s)$$

Since convolution is linear,

$$X_\delta(f) = f_s \sum_{n=-\infty}^{\infty} X(f) * \delta(f-nf_s)$$

$$= f_s \sum_{n=-\infty}^{\infty} X(f - nf_s) \qquad \textit{By shifting property of impulse function}$$

$$= \cdots f_s X(f-2f_s) + f_s X(f-f_s) + f_s X(f) + f_s X(f-f_s) + f_s X(f-2f_s) + \cdots$$

Comments

(i) The RHS of above equation shows that X(f) is placed at $\pm f_s, \pm 2f_s, \pm 3f_s, \cdots$

(ii) This means X(f) is periodic in f_s.

(iii) If sampling frequency is f_s = 2W, then the spectrums *X(f)* just touch each other.

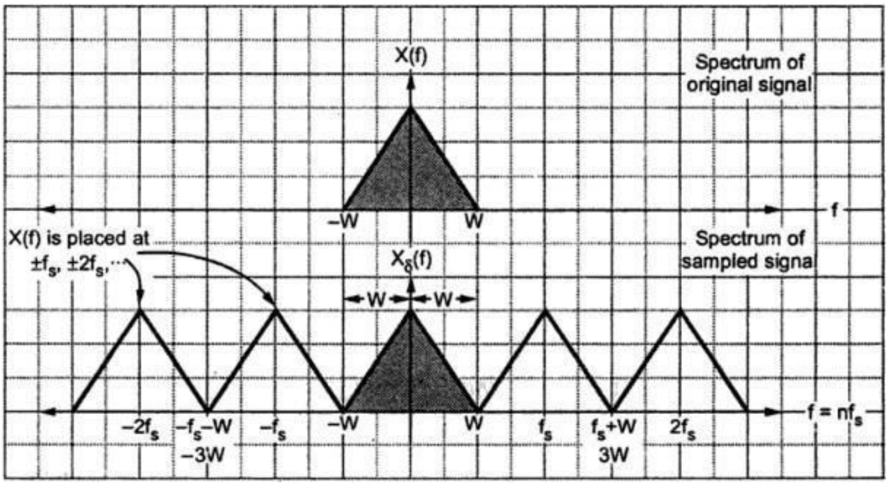


Fig. 6 Spectrum of original signal and sampled signal

Step 3 : Relation between *X(f)* and $X_\delta(f)$

Important assumption : Let us assume that f_s = 2W, then as per above diagram.

$$X_\delta(f) = f_s X(f) \qquad \text{for } -W \le f \le W \text{ and } f_s = 2W$$

or

$$X(f) = \frac{1}{f_s} X_\delta(f) \qquad \ldots \quad 3$$

Step 4 : Relation between $x(t)$ and $x(nT_s)$

DTFT is,

$$X(\Omega) = \sum_{n=-\infty}^{\infty} x(n) e^{-j\Omega n}$$

∴

$$X(f) = \sum_{n=-\infty}^{\infty} x(n) e^{-j2\pi f n} \qquad \ldots \quad 4$$

In above equation 'f' is the frequency of DT signal. If we replace $X(f)$ by $X_\delta(f)$, then 'f' becomes frequency of CT signal. i.e.,

$$X_\delta(f) = \sum_{n=-\infty}^{\infty} x(n) e^{-j2\pi \frac{f}{f_s} n}$$

In above equation 'f' is frequency of CT signal. And $\frac{f}{f_s}$ = Frequency of DT signal in equation 4 Since $x(n) = x(nT_s)$, i.e. samples of $x(t)$, then we have,

$$X_\delta(f) = \sum_{n=-\infty}^{\infty} x(nT_s) e^{-j2\pi f n T_s} \text{ since } \frac{1}{f_s} = T_s$$

Putting above expression in equation 3 ,

$$X(f) = \frac{1}{f_s} \sum_{n=-\infty}^{\infty} x(nT_s) e^{-j2\pi f n T_s}$$

Inverse Fourier Transform (IFT) of above equation gives $x(t)$ i.e.,

$$x(t) = IFT\left\{\frac{1}{f_s} \sum_{n=-\infty}^{\infty} x(nT_s) e^{-j2\pi f n T_s}\right\} \quad \dots \quad 5$$

Comments :

i) Here $x(t)$ is represented completely in terms of $x(nT_s)$.

ii) Above equation holds for $f_s = 2W$. This means if the samples are taken at the rate of 2W or higher, $x(t)$ is completely represented by its samples.

iii) First part of the sampling theorem is proved by above two comments.

Part II : Reconstruction of $x(t)$ from its samples

Step 1 : Take inverse Fourier transform of $X(f)$ which is in terms of $X_\delta(f)$.

Step 2 : Show that $x(t)$ is obtained back with the help of interpolation function.

Step 1 : The IFT of equation 5 becomes,

$$x(t) = \int_{-\infty}^{\infty} \left\{\frac{1}{f_s} \sum_{n=-\infty}^{\infty} x(nT_s) e^{-j2\pi f n T_s}\right\} e^{j2\pi f t}\, df$$

Here the integration can be taken from $-W \le f \le W$. Since $X(f) = \frac{1}{f_s} X_\delta(f)$ for $-W \le f \le W$. (See Fig. 6).

$$\therefore \quad x(t) = \int_{-W}^{W} \frac{1}{f_s} \sum_{n=-\infty}^{\infty} x(nT_s) e^{-j2\pi f n T_s} \cdot e^{j2\pi f t} \, df$$

Interchanging the order of summation and integration,

$$x(t) = \sum_{n=-\infty}^{\infty} x(nT_s) \frac{1}{f_s} \int_{-W}^{W} e^{j2\pi f(t-nT_s)} \, df$$

$$= \sum_{n=-\infty}^{\infty} x(nT_s) \cdot \frac{1}{f_s} \cdot \left[\frac{e^{j2\pi f(t-nT_s)}}{j2\pi(t-nT_s)} \right]_{-W}^{W}$$

$$= \sum_{n=-\infty}^{\infty} x(nT_s) \cdot \frac{1}{f_s} \left\{ \frac{e^{j2\pi W(t-nT_s)} - e^{-j2\pi W(t-nT_s)}}{j2\pi(t-nT_s)} \right\}$$

$$= \sum_{n=-\infty}^{\infty} x(nT_s) \cdot \frac{1}{f_s} \cdot \frac{\sin 2\pi W(t-nT_s)}{\pi(t-nT_s)}$$

$$= \sum_{n=-\infty}^{\infty} x(nT_s) \frac{\sin \pi(2Wt - 2WnT_s)}{\pi(f_s t - f_s nT_s)}$$

Here $f_s = 2W$, hence $T_s = \frac{1}{f_s} = \frac{1}{2W}$. Simplifying above equation,

$$x(t) = \sum_{n=-\infty}^{\infty} x(nT_s) \frac{\sin \pi(2Wt-n)}{\pi(2Wt-n)}$$

$$= \sum_{n=-\infty}^{\infty} x(nT_s) \,\text{sinc}(2Wt-n) \qquad \text{since } \frac{\sin \pi\theta}{\pi\theta} = \text{sinc}\,\theta \qquad \dots \quad 6$$

Step 2 : Let us interpret the above equation. Expanding we get,

$$x(t) = \cdots + x(-2T_s)\,\text{sinc}(2Wt+2) + x(-T_s)\,\text{sinc}(2Wt+1) + x(0)\,\text{sinc}(2Wt) + x(T_s)\,\text{sinc}(2Wt-1) + \cdots$$

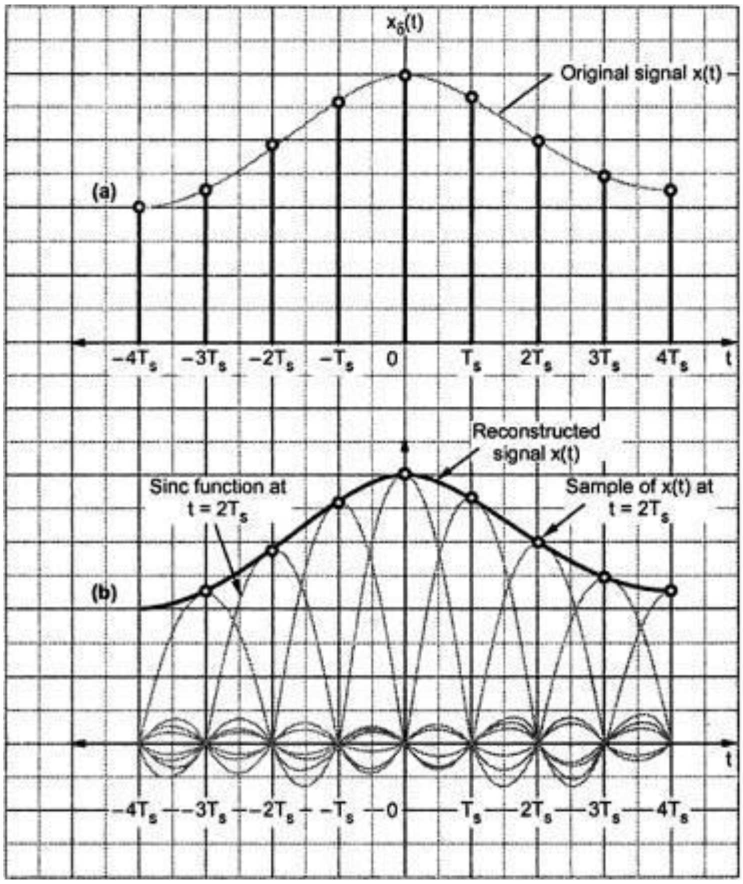


Fig. 7 (a) Sampled version of signal x(t)
(b) Reconstruction of x(t) from its samples

Comments :

i) The samples $x(nT_s)$ are weighted by sinc functions.

ii) The sinc function is the interpolating function. Fig. 7 shows, how $x(t)$ is interpolated.

Step 3 : Reconstruction of $x(t)$ by lowpass filter

When the interpolated signal of equation 6 is passed through the lowpass filter of bandwidth $-W \leq f \leq W$, then the reconstructed waveform shown in above Fig. 7 (b) is obtained. The individual sinc functions are interpolated to get smooth $x(t)$.

PCM Generator:

The pulse code modulator technique samples the input signal $x(t)$ at frequency $f_s \geq 2W$. This sampled 'Variable amplitude' pulse is then digitized by the analog to digital converter. The parallel bits obtained are converted to a serial bit stream. Fig. 8 shows the PCM generator.

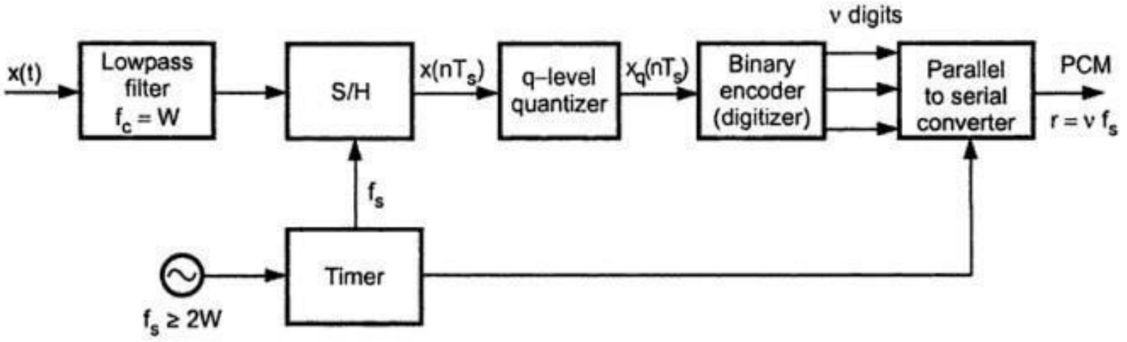


Fig. 8 PCM generator

In the PCM generator of above figure, the signal $x(t)$ is first passed through the lowpass filter of cutoff frequency 'W' Hz. This lowpass filter blocks all the frequency components above 'W' Hz. Thus $x(t)$ is bandlimited to 'W' Hz. The sample and hold circuit then samples this signal at the rate of f_s. Sampling frequency f_s is selected sufficiently above Nyquist rate to avoid aliasing i.e.,

$$f_s \geq 2W$$

In Fig. 8 output of sample and hold is called $x(nT_s)$. This $x(nT_s)$ is discrete in time and continuous in amplitude. A q-level quantizer compares input $x(nT_s)$ with its fixed digital levels. It assigns any one of the digital level to $x(nT_s)$ with its fixed digital levels. It then assigns any one of the digital level to $x(nT_s)$ which results in minimum distortion or error. This error is called *quantization error*. Thus output of quantizer is a digital level called $x_q(nT_s)$.

Now coming back to our discussion of PCM generation, the quantized signal level $x_q(nT_s)$ is given to binary encoder. This encoder converts input signal to 'v' digits binary word. Thus $x_q(nT_s)$ is converted to 'V' binary bits. The encoder is also called digitizer.

It is not possible to transmit each bit of the binary word separately on transmission line. Therefore '*v*' binary digits are converted to serial bit stream to generate single baseband signal. In a parallel to serial converter, normally a shift register does this job. The output of PCM generator is thus a single baseband signal of binary bits.

An oscillator generates the clocks for sample and hold an parallel to serial converter. In the pulse code modulation generator discussed above ; sample and hold, quantizer and encoder combinely form an analog to digital converter.

Transmission BW in PCM:

Let the quantizer use 'v' number of binary digits to represent each level. Then the number of levels that can be represented by 'v' digits will be,

$$q = 2^v \quad \dots \quad 1$$

Here 'q' represents total number of digital levels of q-level quantizer.

For example if $v = 3$ bits, then total number of levels will be,

$$q = 2^3 = 8 \text{ levels}$$

Each sample is converted to 'v' binary bits. i.e. Number of bits per sample = v

We know that, Number of samples per second = f_s

∴ Number of bits per second is given by,

(Number of bits per second) = (Number of bits per samples) × (Number of samples per second)

$$= v \text{ bits per sample} \times f_s \text{ samples per second} \quad \dots \quad 2$$

The number of bits per second is also called signaling rate of PCM and is denoted by 'r' i.e.,

$$\boxed{\text{Signaling rate in PCM : } r = v f_s} \quad \dots \quad 3$$

Here $f_s \geq 2W$.

Bandwidth needed for PCM transmission will be given by half of the signaling rate i.e.,

Transmission Bandwidth of PCM :

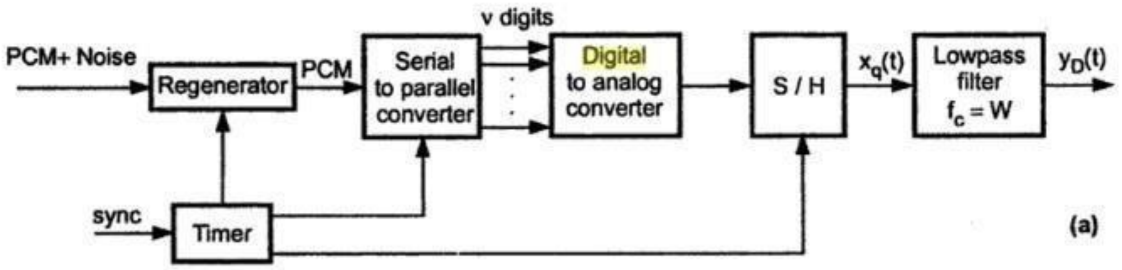
$$B_T \geq \frac{1}{2} r \quad \dots \quad 4$$

$$B_T \geq \frac{1}{2} v f_s \quad \text{Since } f_s \geq 2W \quad \dots \quad 5$$

$$B_T \geq v W \quad \dots \quad 6$$

PCM Receiver:

Fig. 9 (a) shows the block diagram of PCM receiver and Fig. 9 (b) shows the reconstructed signal. The regenerator at the start of PCM receiver reshapes the pulses and removes the noise. This signal is then converted to parallel digital words for each sample.



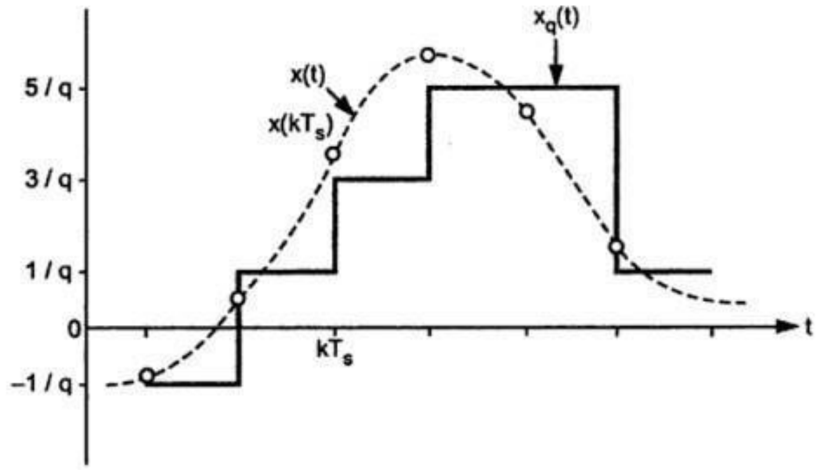


Fig. 9 (a) PCM receiver
(b) Reconstructed waveform

The digital word is converted to its analog value $x_q(t)$ along with sample and hold. This signal, at the output of S/H is passed through lowpass reconstruction filter to get $y_D(t)$. As shown in reconstructed signal of Fig. 9 (b), it is impossible to reconstruct exact original signal $x(t)$ because of permanent quantization error introduced during quantization at the transmitter. This quantization error can be reduced by increasing the binary levels. This is equivalent to increasing binary digits (bits) per sample. But increasing bits 'v' increases the signaling rate as well as transmission bandwidth as we have seen in equation 3 and equation 6. Therefore the choice of these parameters is made, such that noise due to quantization error (called as quantization noise) is in tolerable limits.

Quantization

- The quantizing of an analog signal is done by discretizing the signal with a number of quantization levels.

- **Quantization** is representing the sampled values of the amplitude by a finite set of levels, which means converting a continuous-amplitude sample into a discrete-time signal
- Both sampling and quantization result in the loss of information.
- The quality of a Quantizer output depends upon the number of quantization levels used.
- The discrete amplitudes of the quantized output are called as **representation levels** or **reconstruction levels**.
- The spacing between the two adjacent representation levels is called a **quantum** or **step-size**.
- There are two types of Quantization
    - Uniform Quantization
    - Non-uniform Quantization.
- The type of quantization in which the quantization levels are uniformly spaced is termed as a **Uniform Quantization**.
- The type of quantization in which the quantization levels are unequal and mostly the relation between them is logarithmic, is termed as a **Non-uniform Quantization.**

Uniform Quantization:

- There are two types of uniform quantization.
    - Mid-Rise type
    - Mid-Tread type.
- The following figures represent the two types of uniform quantization.

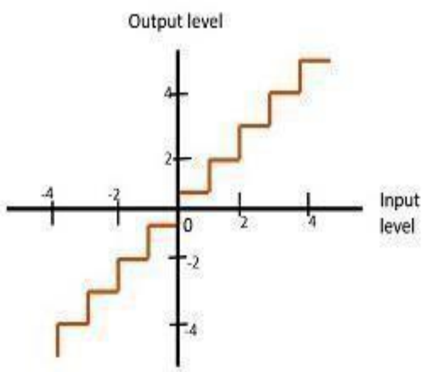


Fig 1 : Mid-Rise type Uniform Quantization

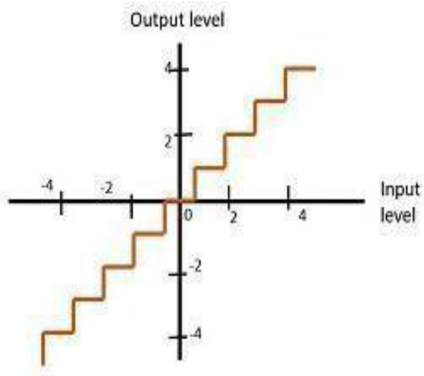


Fig 2 : Mid-Tread type Uniform Quantization

- The **Mid-Rise** type is so called because the origin lies in the middle of a raising part of the stair-case like graph. The quantization levels in this type are even in number.
- The **Mid-tread** type is so called because the origin lies in the middle of a tread of the stair-case like graph. The quantization levels in this type are odd in number.
- Both the mid-rise and mid-tread type of uniform quantizer is symmetric about the origin.

Quantization Noise and Signal to Noise ratio in PCM System:

Derivation of Quantization Error/Noise or Noise Power for Uniform (Linear) Quantization

Step 1 : Quantization Error

Because of quantization, inherent errors are introduced in the signal. This error is called *quantization error*. We have defined quantization error as,

$$\varepsilon = x_q(nT_s) - x(nT_s) \quad \text{.......................... (1)}$$

Step 2 : Step size

Let an input $x(nT_s)$ be of continuous amplitude in the range $-x_{max}$ to $+x_{max}$.

Therefore the total amplitude range becomes,

$$\text{Total amplitude range} = x_{max} - (-x_{max}) = 2x_{max} \quad \text{.......................(2)}$$

If this amplitude range is divided into 'q' levels of quantizer, then the step size 'δ' is given as,

$$\delta = \frac{x_{max} - (-x_{max})}{q}$$

$$= \frac{2x_{max}}{q} \quad \text{...................(3)}$$

If signal $x(t)$ is normalized to minimum and maximum values equal to 1, then

$$x_{max} = 1$$
$$-x_{max} = -1 \quad \text{..............................(4)}$$

Therefore step size will be,

$$\delta = \frac{2}{q} \quad \text{(for normalized signal)} \quad \text{...............................(5)}$$

Step 3 : Pdf of Quantization error

If step size 'δ' is sufficiently small, then it is reasonable to assume that the quantization error 'ε' will be uniformly distributed random variable. The maximum quantization error is given by

$$\varepsilon_{max} = \left|\frac{\delta}{2}\right| \quad \text{............................(6)}$$

i.e. $$-\frac{\delta}{2} \geq \varepsilon_{max} \geq \frac{\delta}{2} \quad \text{...............................(7)}$$

Thus over the interval $\left(-\frac{\delta}{2}, \frac{\delta}{2}\right)$ quantization error is uniformly distributed random variable.

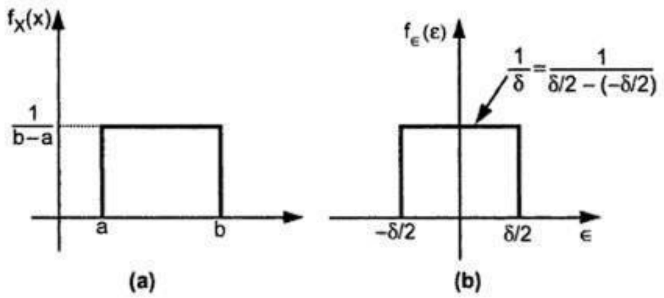


Fig. 10 (a) Uniform distribution
(b) Uniform distribution for quantization error

In above figure, a random variable is said to be uniformly distributed over an interval (a, b). Then PDF of 'X' is given by, (from equation of Uniform PDF).

$$f_X(x) = \begin{cases} 0 & \text{for} \quad x \le a \\ \frac{1}{b-a} & \text{for} \quad a < x \le b \\ 0 & \text{for} \quad x > b \end{cases} \qquad \text{.......................(8)}$$

Thus with the help of above equation we can define the probability density function for quantization error 'ε' as,

$$f_\varepsilon(\varepsilon) = \begin{cases} 0 & \text{for} \quad \varepsilon \le \frac{\delta}{2} \\ \frac{1}{\delta} & \text{for} \quad -\frac{\delta}{2} < \varepsilon \le \frac{\delta}{2} \\ 0 & \text{for} \quad \varepsilon > \frac{\delta}{2} \end{cases} \qquad \text{.................................(9)}$$

Step 4 : Noise Power

quantization error 'ε' has zero average value. That is mean 'm_ε' of the quantization error is zero.

The signal to quantization noise ratio of the quantizer is defined as,

$$\frac{S}{N} = \frac{\text{Signal power (normalized)}}{\text{Noise power (normalized)}} \quad \cdots 10$$

If type of signal at input i.e., $x(t)$ is known, then it is possible to calculate signal power.

The noise power is given as,

$$\text{Noise power} = \frac{V_{noise}^2}{R} \quad \cdots (11)$$

Here V_{noise}^2 is the mean square value of noise voltage. Since noise is defined by random variable 'ε' and PDF $f_\varepsilon(\varepsilon)$, its mean square value is given as,

$$\text{mean square value} = E[\varepsilon^2] = \bar{\varepsilon}^2 \quad \cdots (12)$$

The mean square value of a random variable 'X' is given as,

$$\bar{X}^2 = E[X^2] = \int_{-\infty}^{\infty} x^2 f_X(x)\, dx \quad \text{By definition} \quad \cdots (13)$$

Here $$E[\varepsilon^2] = \int_{-\infty}^{\infty} \varepsilon^2 f_\varepsilon(\varepsilon)\, d\varepsilon \quad \cdots (14)$$

From equation 9 we can write above equation as,

$$E[\varepsilon^2] = \int_{-\delta/2}^{\delta/2} \varepsilon^2 \times \frac{1}{\delta}\, d\varepsilon$$

$$= \frac{1}{\delta}\left[\frac{\varepsilon^3}{3}\right]_{-\delta/2}^{\delta/2} = \frac{1}{\delta}\left[\frac{(\delta/2)^3}{3} + \frac{(\delta/2)^3}{3}\right]$$

$$= \frac{1}{3\delta}\left[\frac{\delta^3}{8} + \frac{\delta^3}{8}\right] = \frac{\delta^2}{12} \quad \cdots (15)$$

∴ From equation 1.8.25, the mean square value of noise voltage is,

$$V_{noise}^2 = \text{mean square value} = \frac{\delta^2}{12}$$

When load resistance, $R = 1$ ohm, then the noise power is normalized i.e.,

$$\text{Noise power (normalized)} = \frac{V_{noise}^2}{1} \qquad \text{[with } R = 1 \text{ in equation 11]}$$

$$= \frac{\delta^2 / 12}{1} = \frac{\delta^2}{12}$$

Thus we have,

Normalized noise power
or Quantization noise power $= \frac{\delta^2}{12}$ **; For linear quantization.**
or Quantization error (in terms of power)

... (16)

Derivation of Maximum Signal to Quantization Noise Ratio for Linear Quantization:

signal to quantization noise ratio is given as,

$$\frac{S}{N} = \frac{\text{Normalized signal power}}{\text{Normalized noise power}}$$

$$= \frac{\text{Normalized signal power}}{(\delta^2 / 12)} \qquad \text{... (17)}$$

The number of bits 'v' and quantization levels 'q' are related as,

$$q = 2^v \qquad \text{... (18)}$$

Putting this value in equation (3) we have,

$$\delta = \frac{2\, x_{max}}{2^v} \qquad \text{... (19)}$$

Putting this value in equation 1.8.30 we get,

$$\frac{S}{N} = \frac{\text{Normalized signal power}}{\left(\frac{2\, x_{max}}{2^v}\right)^2 + 12}$$

Let normalized signal power be denoted as 'P'.

$$\frac{S}{N} = \frac{P}{\frac{4\, x_{max}^2}{2^{2v}} \times \frac{1}{12}} = \frac{3P}{x_{max}^2} \cdot 2^{2v}$$

This is the required relation for maximum signal to quantization noise ratio. Thus,

Maximum signal to quantization noise ratio : $\frac{S}{N} = \frac{3P}{x_{max}^2} \cdot 2^{2v}$ ... (20)

This equation shows that signal to noise power ratio of quantizer increases exponentially with increasing bits per sample.

If we assume that input $x(t)$ is normalized, i.e.,

$$x_{max} = 1 \quad \text{... (21)}$$

Then signal to quantization noise ratio will be,

$$\frac{S}{N} = 3 \times 2^{2v} \times P \quad \text{... (22)}$$

If the destination signal power 'P' is normalized, i.e.,

$$P \leq 1 \quad \text{... (23)}$$

Then the signal to noise ratio is given as,

$$\frac{S}{N} \leq 3 \times 2^{2v} \quad \text{... (24)}$$

Since $x_{max} = 1$ and $P \leq 1$, the signal to noise ratio given by above equation is normalized.

Expressing the signal to noise ratio in decibels,

$$\left(\frac{S}{N}\right) dB = 10 \log_{10} \left(\frac{S}{N}\right) dB \quad \text{since power ratio.}$$

$$\leq 10 \log_{10} [3 \times 2^{2v}]$$

$$\leq (4.8 + 6v)\, dB$$

Thus,

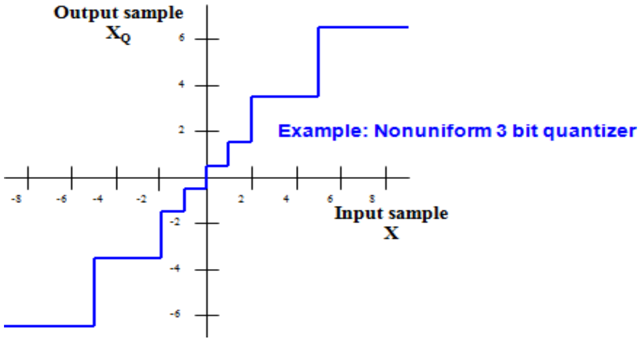
Signal to Quantization noise ratio

for normalized values of power : $\left(\frac{S}{N}\right) dB \leq (4.8 + 6v)\, dB$

'P' and amplitude of input $x(t)$ ... (25)

Non-Uniform Quantization:

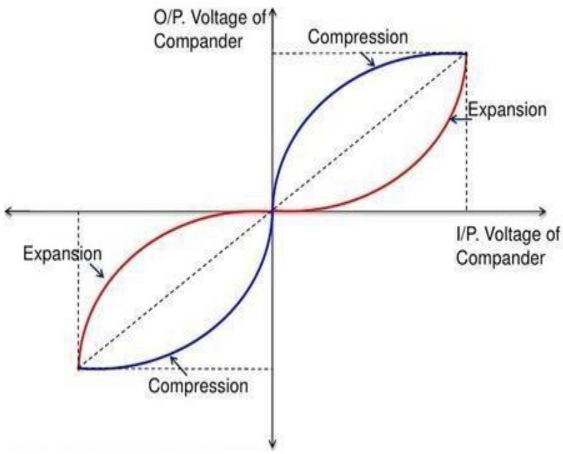
In non-uniform quantization, the step size is not fixed. It varies according to certain law or as per input signal amplitude. The following fig shows the characteristics of Non uniform quantizer.

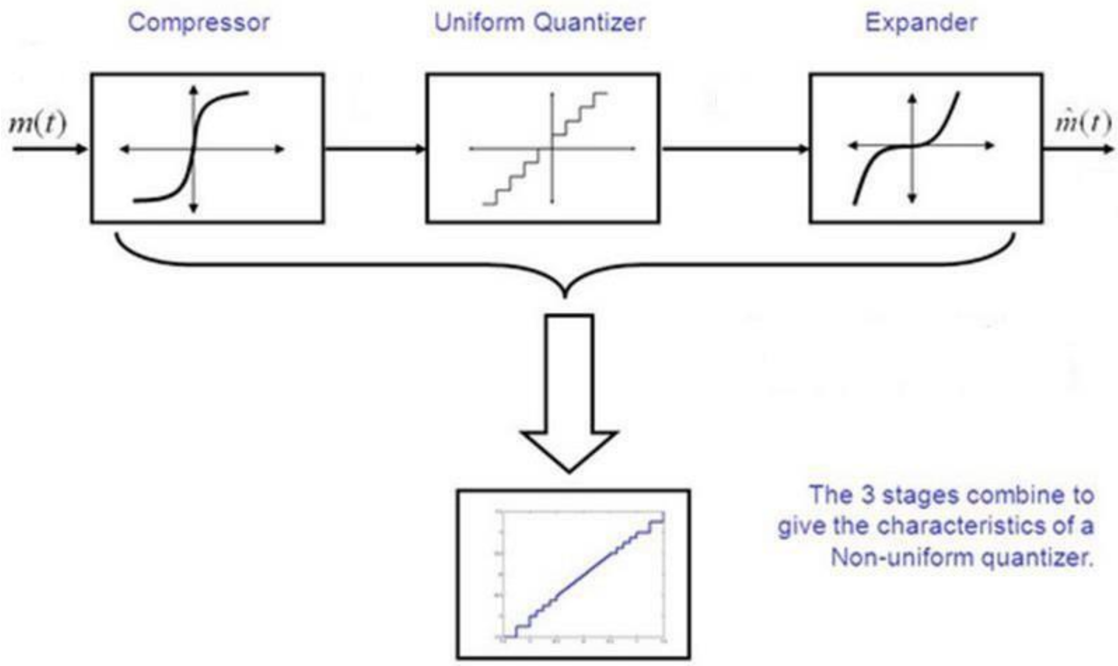


In this figure observe that step size is small at low input signal levels. Hence quantization error is also small at these inputs. Therefore signal to quantization noise power ratio is improved at low signal levels. Stepsize is higher at high input levels. Hence signal to noise power ratio remains almost same throughout the dynamic range of quantizer.

Companding PCM System:

- Non-uniform quantizers are difficult to make and expensive.
- An alternative is to first pass the speech signal through nonlinearity before quantizing with a uniform quantizer.
- The nonlinearity causes the signal amplitude to be ***compressed***.
  - The input to the quantizer will have a more uniform distribution.
- At the receiver, the signal is ***expanded*** by an inverse to the nonlinearity.
- The process of compressing and expanding is called ***Companding***.





μ - Law Companding for Speech Signals

Normally for speech and music signals a μ - law compression is used. This compression is defined by the following equation,

$$Z(x) = (\text{Sgn } x) \frac{ln(1+\mu|x|)}{ln(1+\mu)} \quad |x| \leq 1 \qquad \cdots (1)$$

Below Fig shows the variation of signal to noise ratio with respect to signal level without companding and with companding.

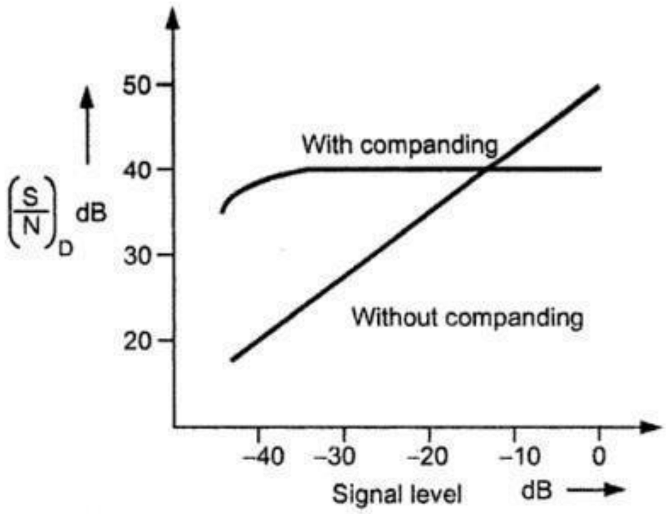


Fig. 11 PCM performance with μ - law companding

It can be observed from above figure that signal to noise ratio of PCM remains almost constant with companding.

A-Law for Companding

The A law provides piecewise compressor characteristic. It has linear segment for low level inputs and logarithmic segment for high level inputs. It is defined as,

$$Z(x) = \begin{cases} \dfrac{A|x|}{1+\ln A} & \text{for } 0 \le |x| \le \dfrac{1}{A} \\ \dfrac{1+\ln(A|x|)}{1+\ln A} & \text{for } \dfrac{1}{A} \le |x| \le 1 \end{cases} \quad \cdots (2)$$

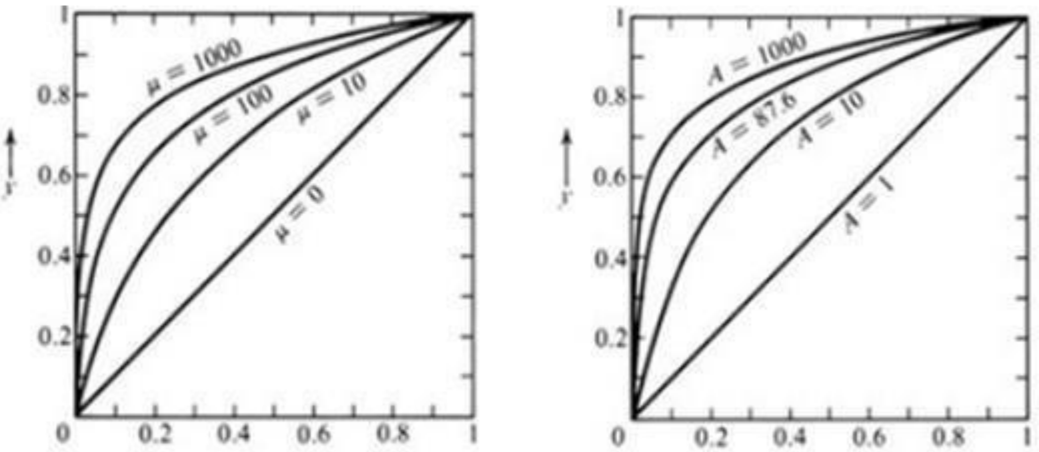
When A = 1, we get uniform quantization. The practical value for A is 87.56. Both A-law and μ-law companding is used for PCM telephone systems.

Signal to Noise Ratio of Companded PCM

The signal to noise ratio of companded PCM is given as,

$$\frac{S}{N} = \frac{3q^2}{[\ln(1+\mu)]^2} \quad \cdots (3)$$

Here $q = 2^v$ is number of quantization levels.



<u>Differential Pulse Code Modulation (DPCM):</u>

Redundant Information in PCM:

The samples of a signal are highly corrected with each other. This is because any signal does not change fast. That is its value from present sample to next sample does not differ by large amount. The adjacent samples of the signal carry the same information with little difference. When these samples are encoded by standard PCM system, the resulting encoded signal contains redundant information.

Fig. shows a continuous time signal $x(t)$ by dotted line. This signal is sampled by flat top sampling at intervals $T_s, 2T_s, 3T_s \dots nT_s$. The sampling frequency is selected to be higher than nyquist rate. The samples are encoded by using 3 bit (7 levels) PCM. The sample is quantized to the nearest digital level as shown by small

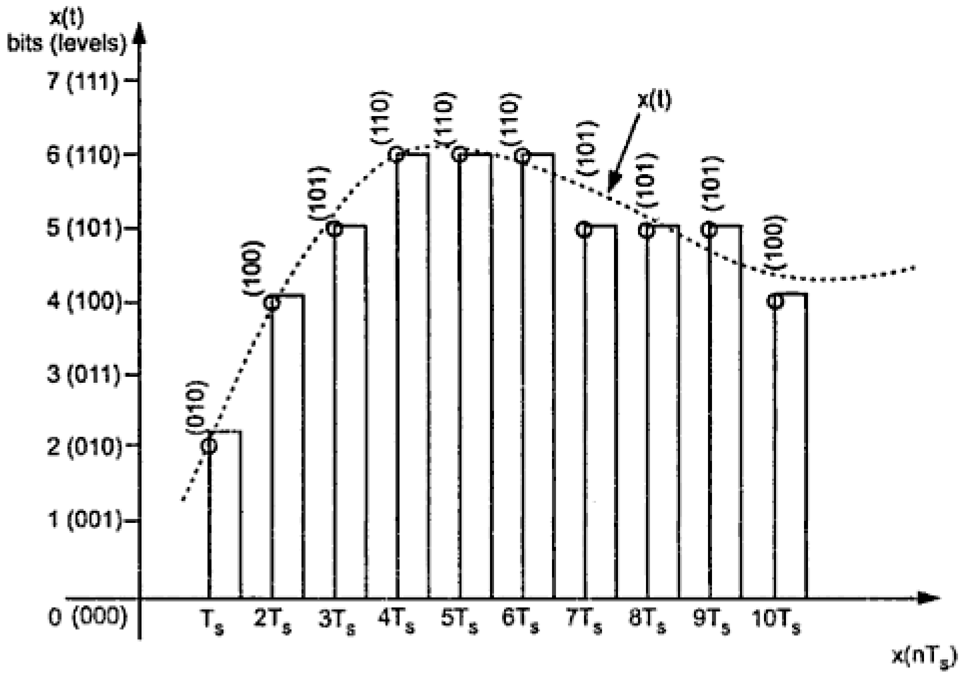


Fig. Redundant information in PCM

circles in the diagram. The encoded binary value of each sample is written on the top of the samples. We can see from Fig. that the samples taken at $4T_s, 5T_s$ and $6T_s$ are encoded to same value of (110). This information can be carried only by one sample. But three samples are carrying the same information means it is redundant. Consider another example of samples taken at $9T_s$ and $10T_s$. The difference between these samples is only due to last bit and first two bits are redundant, since they do not change.

Principle of DPCM

If this redundancy is reduced, then overall bit rate will decrease and number of bits required to transmit one sample will also be reduced. This type of digital pulse modulation scheme is called Differential Pulse Code Modulation.

DPCM Transmitter

The differential pulse code modulation works on the principle of prediction. The value of the present sample is predicted from the past samples. The prediction may not be exact but it is very close to the actual sample value. Fig. shows the transmitter of Differential Pulse Code Modulation (DPCM) system. The sampled signal is denoted by $x(nT_s)$ and the predicted signal is denoted by $\hat{x}(nT_s)$. The comparator finds out the difference between the actual sample value $x(nT_s)$ and predicted sample value $\hat{x}(nT_s)$. This is called error and it is denoted by $e(nT_s)$. It can be defined as,

$$e(nT_s) = x(nT_s) - \hat{x}(nT_s) \quad \text{..................(1)}$$

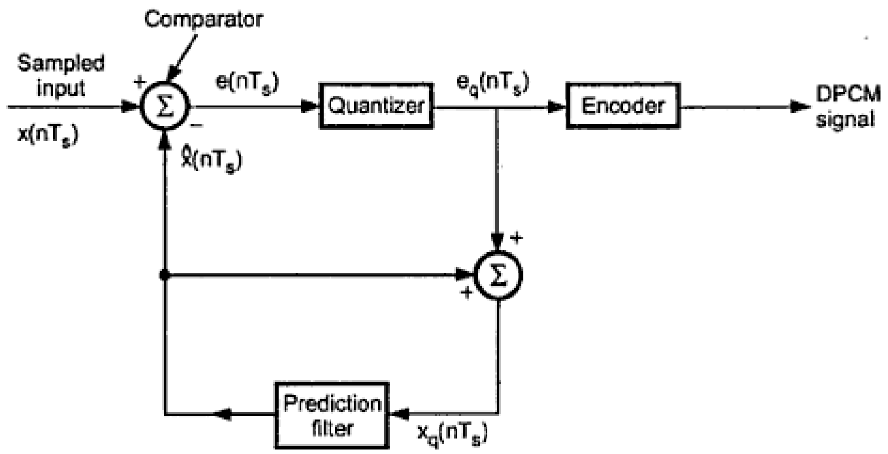


Fig. Differential pulse code modulation transmitter

Thus error is the difference between unquantized input sample $x(nT_s)$ and prediction of it $\hat{x}(nT_s)$. The predicted value is produced by using a prediction filter. The quantizer output signal $e_q(nT_s)$ and previous prediction is added and given as input to the prediction filter. This signal is called $x_q(nT_s)$. This makes the prediction more and more close to the actual sampled signal. We can see that the quantized error signal $e_q(nT_s)$ is very small and can be encoded by using small number of bits. Thus number of bits per sample are reduced in DPCM.

The quantizer output can be written as,

$$e_q(nT_s) = e(nT_s) + q(nT_s) \qquad \text{.................(2)}$$

Here $q(nT_s)$ is the quantization error. As shown in Fig. the prediction filter input $x_q(nT_s)$ is obtained by sum $\hat{x}(nT_s)$ and quantizer output i.e.,

$$x_q(nT_s) = \hat{x}(nT_s) + e_q(nT_s) \qquad \text{...................(3)}$$

Putting the value of $e_q(nT_s)$ from equation 2 in the above equation we get,

$$x_q(nT_s) = \hat{x}(nT_s) + e(nT_s) + q(nT_s) \qquad \text{................(4)}$$

Equation 1 is written as,

$$e(nT_s) = x(nT_s) - \hat{x}(nT_s)$$

$$\therefore \quad e(nT_s) + \hat{x}(nT_s) = x(nT_s) \qquad \text{...............(5)}$$

$\therefore$ Putting the value of $e(nT_s) + \hat{x}(nT_s)$ from above equation into equation 4 we get,

$$x_q(nT_s) = x(nT_s) + q(nT_s) \qquad \text{...................(6)}$$

Thus the quantized version of the signal $x_q(nT_s)$ is the sum of original sample value and quantization error $q(nT_s)$. The quantization error can be positive or negative. Thus equation 6 does not depend on the prediction filter characteristics.

Reconstruction of DPCM Signal

Fig. shows the block diagram of DPCM receiver.

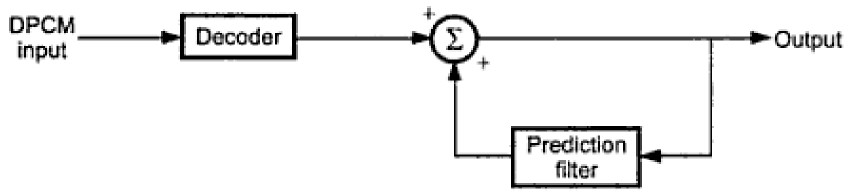


Fig. DPCM receiver

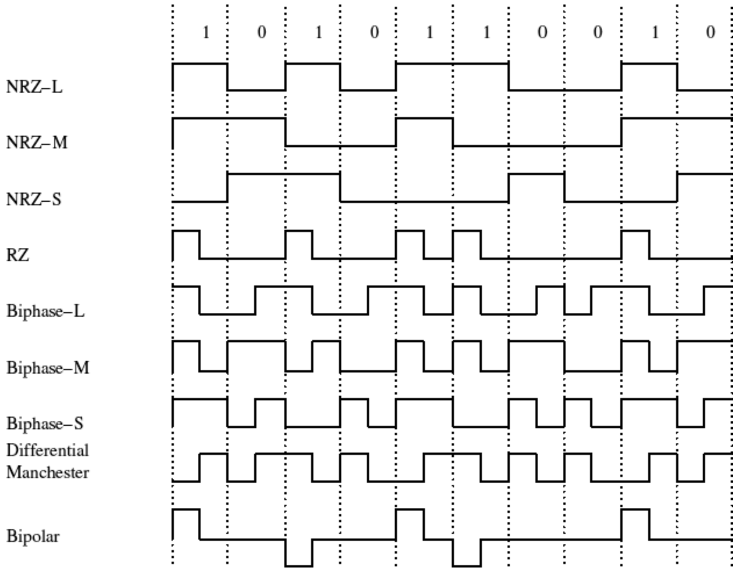
The decoder first reconstructs the quantized error signal from incoming binary signal. The prediction filter output and quantized error signals are summed up to give the quantized version of the original signal. Thus the signal at the receiver differs from actual signal by quantization error $q(nT_s)$, which is introduced permanently in the reconstructed signal.

Line Coding:

In telecommunication, a line code is a code chosen for use within a communications system for transmitting a digital signal down a transmission line. Line coding is often used for digital data transport.

The waveform pattern of voltage or current used to represent the 1s and 0s of a digital signal on a transmission link is called **line** encoding. The common types of

line encoding are unipolar, polar, bipolar and Manchester encoding. **Line codes** are used commonly in computer communication networks over short distances.



Signal	Comments
NRZ–L	Non-return to zero level. This is the standard positive logic signal format used in digital circuits. 1 forces a high level 0 forces a low level
NRZ–M	Non return to zero mark 1 forces a transition 0 does nothing
NRZ–S	Non return to zero space 1 does nothing 0 forces a transition
RZ	Return to zero 1 goes high for half the bit period 0 does nothing
Biphase–L	Manchester. Two consecutive bits of the same type force a transition at the beginning of a bit period. 1 forces a negative transition in the middle of the bit 0 forces a positive transition in the middle of the bit
Biphase–M	There is always a transition at the beginning of a bit period. 1 forces a transition in the middle of the bit 0 does nothing
Biphase–S	There is always a transition at the beginning of a bit period. 1 does nothing 0 forces a transition in the middle of the bit
Differential Manchester	There is always a transition in the middle of a bit period. 1 does nothing 0 forces a transition at the beginning of the bit
Bipolar	The positive and negative pulses alternate. 1 forces a positive or negative pulse for half the bit period 0 does nothing

Time Division Multiplexing:

The sampling theorem provides the basis for transmitting the information contained in a band-limited message signal $m(t)$ as a sequence of samples of $m(t)$ taken uniformly at a rate that is usually slightly higher than the Nyquist rate. An important feature of the sampling process is a *conservation of time*. That is, the transmission of the message samples engages the communication channel for only a fraction of the sampling interval on a periodic basis, and in this way some of the time interval between adjacent samples is cleared for use by other independent message sources on a time-shared basis. We thereby obtain a *time-division multiplex* (TDM) *system*, which enables the joint utilization of a common communication channel by a plurality of independent message sources without mutual interference among them.

The concept of TDM is illustrated by the block diagram shown in Figure Each input message signal is first restricted in bandwidth by a low-pass anti-aliasing filter to remove the frequencies that are nonessential to an adequate signal representation. The low-pass filter outputs are then applied to a *commutator*, which is usually implemented using electronic switching circuitry. The function of the commutator is twofold: (1) to take a narrow sample of each of the N input messages at a rate f_s that is slightly higher than $2W$, where W is the cutoff frequency of the anti-aliasing filter, and (2) to sequentially interleave these N samples inside the sampling interval T_s. Indeed, this latter function is the essence of the time-division multiplexing operation. Following the commutation process, the multiplexed signal is applied to a *pulse modulator*, the purpose of which is to transform the multiplexed signal into a form suitable for transmission over the common channel. It is clear that the use of time-division multiplexing introduces a bandwidth expansion factor N, because the scheme must squeeze N samples derived from N independent message sources into a time slot equal to one sampling interval. At the receiving end of the system, the received signal is applied to a *pulse demodulator*, which performs the reverse operation of the pulse modulator. The narrow samples produced at the pulse demodulator output are distributed to the appropriate low-pass reconstruction filters by means of a *decommutator*, which operates in *synchronism* with the commutator in the transmitter. This synchronization is essential for a satisfactory operation of the system.

The way this synchronization is implemented depends naturally on the method of pulse modulation used to transmit the multiplexed sequence of samples.

The TDM system is highly sensitive to dispersion in the common channel, that is, to variations of amplitude with frequency or lack of proportionality of phase with frequency. Accordingly, accurate equalization of both magnitude and phase responses of the channel is necessary to ensure a satisfactory operation of the system;

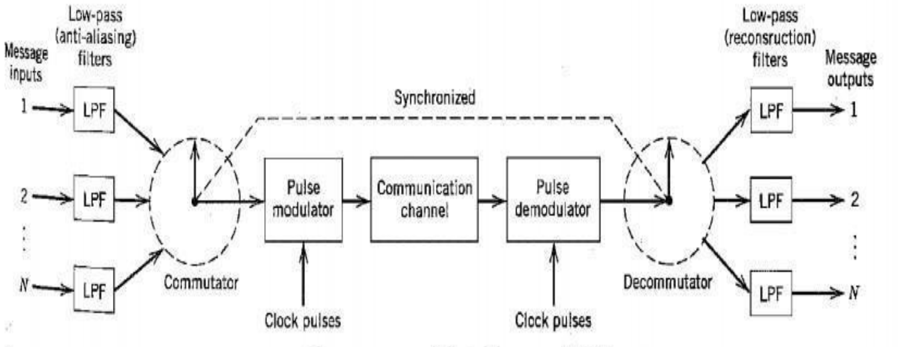


FIGURE Block diagram of TDM system.

TDM is immune to nonlinearities in the channel as a source of crosstalk. The reason for this behaviour is that different message signals are not simultaneously applied to the channel.

Introduction to Delta Modulation

PCM transmits all the bits which are used to code the sample. Hence signaling rate and transmission channel bandwidth are large in PCM. To overcome this problem Delta Modulation is used.

Delta Modulation

Operating Principle of DM

Delta modulation transmits only one bit per sample. That is the present sample value is compared with the previous sample value and the indication,whether the amplitude is increased or decreased is sent. Input signal $x(t)$ is approximated to step signal by the delta modulator. This step size is fixed . The difference between the input signal $x(t)$ and staircase approximated signal confined to two levels, i.e. $+\delta$ and $-\delta$. If the difference is positive, then approximated signal is increased by one step i.e. 'δ'. If the difference is negative, then approximated signal is reduced by 'δ'. When the step is reduced, '0' is transmitted and if the step is increased, '1' is transmitted. Thus for each sample, only one binary bit is transmitted. Fig. shows the analog signal $x(t)$ and its staircase approximated signal by the delta modulator.

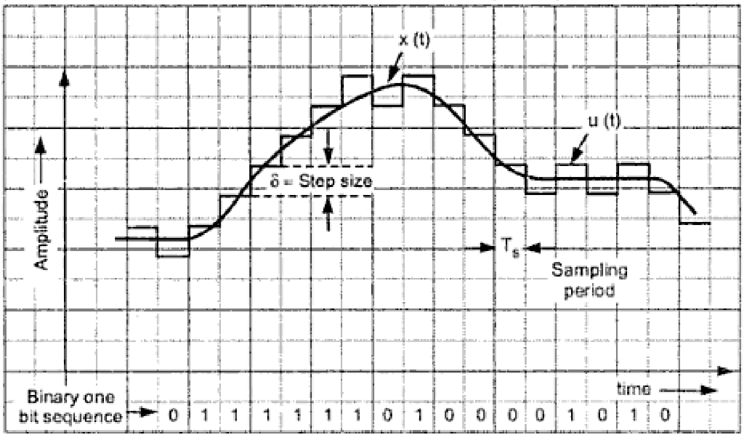


Fig. Delta modulation waveform

The principle of delta modulation can be explained by the following set of equations. The error between the sampled value of $x(t)$ and last approximated sample is given as,

$$e(nT_s) = x(nT_s) - \hat{x}(nT_s) \qquad \ldots (1)$$

Here, $e(nT_s)$ = Error at present sample

$x(nT_s)$ = Sampled signal of $x(t)$

$\hat{x}(nT_s)$ = Last sample approximation of the staircase waveform.

We can call $u(nT_s)$ as the present sample approximation of staircase output.

Then, $u[(n-1)T_s] = \hat{x}(nT_s) \qquad \ldots (2)$

= Last sample approximation of staircase waveform.

Let the quantity $b(nT_s)$ be defined as,

$$b(nT_s) = \delta\, sgn[e(nT_s)] \qquad \ldots (3)$$

That is depending on the sign of error $e(nT_s)$ the sign of step size δ will be decided. In other words,

$$b(nT_s) = +\delta \quad \text{if} \quad x(nT_s) \geq \hat{x}(nT_s)$$

$$= -\delta \quad \text{if} \quad x(nT_s) < \hat{x}(nT_s) \qquad \ldots (4)$$

If $b(nT_s) = +\delta$; binary '1' is transmitted

and if $b(nT_s) = -\delta$; binary '0' is transmitted.

T_s = Sampling interval.

DM Transmitter

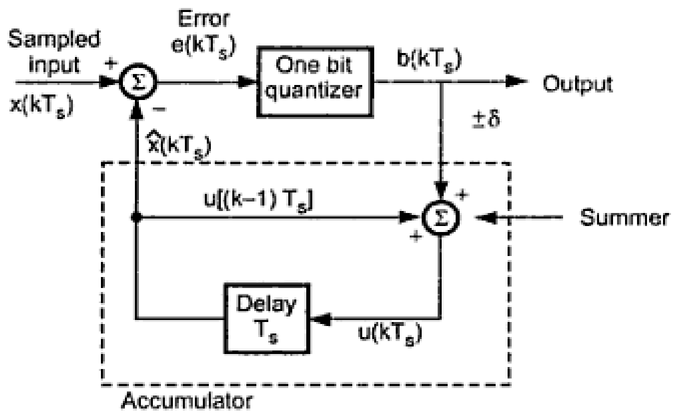
Fig. (a) shows the transmitter based on equations 3 to 5.

The summer in the accumulator adds quantizer output ($\pm\delta$) with the previous sample approximation. This gives present sample approximation. i.e.,

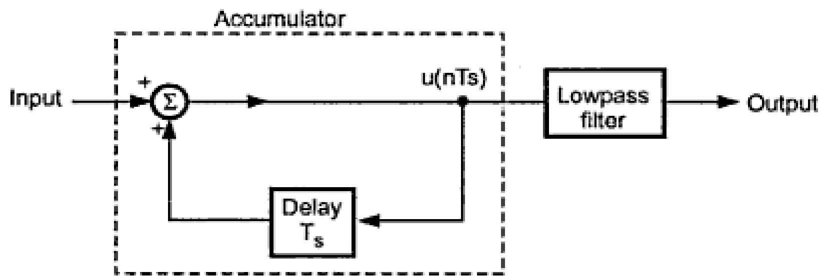
$$u(nT_s) = u(nT_s - T_s) + [\pm\delta] \quad \text{or}$$

$$= u[(n-1)T_s] + b(nT_s) \qquad \dots(5)$$

The previous sample approximation $u[(n-1)T_s]$ is restored by delaying one sample period T_s. The sampled input signal $x(nT_s)$ and staircase approximated signal $\hat{x}(nT_s)$ are subtracted to get error signal $e(nT_s)$.



(a)



(b)

Fig. (a) Delta modulation transmitter and (b) Delta modulation receiver

Depending on the sign of $e(nT_s)$ one bit quantizer produces an output step of $+\delta$ or $-\delta$. If the step size is $+\delta$, then binary '1' is transmitted and if it is $-\delta$, then binary '0' is transmitted.

DM Receiver

At the receiver shown in Fig. (b), the accumulator and low-pass filter are used. The accumulator generates the staircase approximated signal output and is delayed by one sampling period T_s. It is then added to the input signal. If input is binary '1' then it adds $+\delta$ step to the previous output (which is delayed). If input is binary '0' then one step 'δ' is subtracted from the delayed signal. The low-pass filter has the cutoff frequency equal to highest frequency in $x(t)$. This filter smoothen the staircase signal to reconstruct $x(t)$.

Advantages and Disadvantages of Delta Modulation

Advantages of Delta Modulation

The delta modulation has following advantages over PCM,

1. Delta modulation transmits only one bit for one sample. Thus the signaling rate and transmission channel bandwidth is quite small for delta modulation.
2. The transmitter and receiver implementation is very much simple for delta modulation. There is no analog to digital converter involved in delta modulation.

Disadvantages of Delta Modulation

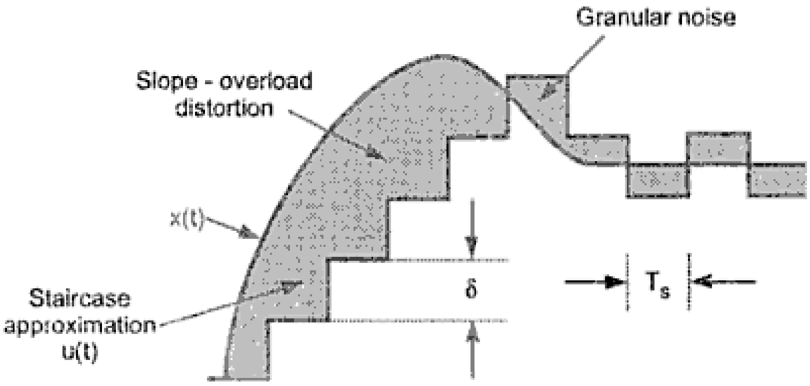


Fig. Quantization errors in delta modulation

The delta modulation has two drawbacks -

Slope Overload Distortion (Startup Error)

This distortion arises because of the large dynamic range of the input signal.

As can be seen from Fig. the rate of rise of input signal $x(t)$ is so high that the staircase signal cannot approximate it, the step size 'δ' becomes too small for staircase signal $u(t)$ to follow the steep segment of $x(t)$. Thus there is a large error between the staircase approximated signal and the original input signal $x(t)$. This error is called *slope overload distortion*. To reduce this error, the step size should be increased when slope of signal of $x(t)$ is high.

Since the step size of delta modulator remains fixed, its maximum or minimum slopes occur along straight lines. Therefore this modulator is also called Linear Delta Modulator (LDM).

Granular Noise (Hunting)

Granular noise occurs when the step size is too large compared to small variations in the input signal. That is for very small variations in the input signal, the staircase signal is changed by large amount (δ) because of large step size. Fig shows that when the input signal is almost flat, the staircase signal $u(t)$ keeps on oscillating by $\pm\delta$ around the signal. The error between the input and approximated signal is called *granular noise*. The solution to this problem is to make step size small.

Thus large step size is required to accommodate wide dynamic range of the input signal (to reduce slope overload distortion) and small steps are required to reduce granular noise. Adaptive delta modulation is the modification to overcome these errors.

Adaptive Delta Modulation

Operating Principle

To overcome the quantization errors due to slope overload and granular noise, the step size (δ) is made adaptive to variations in the input signal $x(t)$. Particularly in the steep segment of the signal $x(t)$, the step size is increased. When the input is varying slowly, the step size is reduced. Then the method is called *Adaptive Delta Modulation (ADM)*.

The adaptive delta modulators can take continuous changes in step size or discrete changes in step size.

Transmitter and Receiver

Fig. (a) shows the transmitter and (b) shows receiver of adaptive delta modulator. The logic for step size control is added in the diagram. The step size increases or decreases according to certain rule depending on one bit quantizer output.

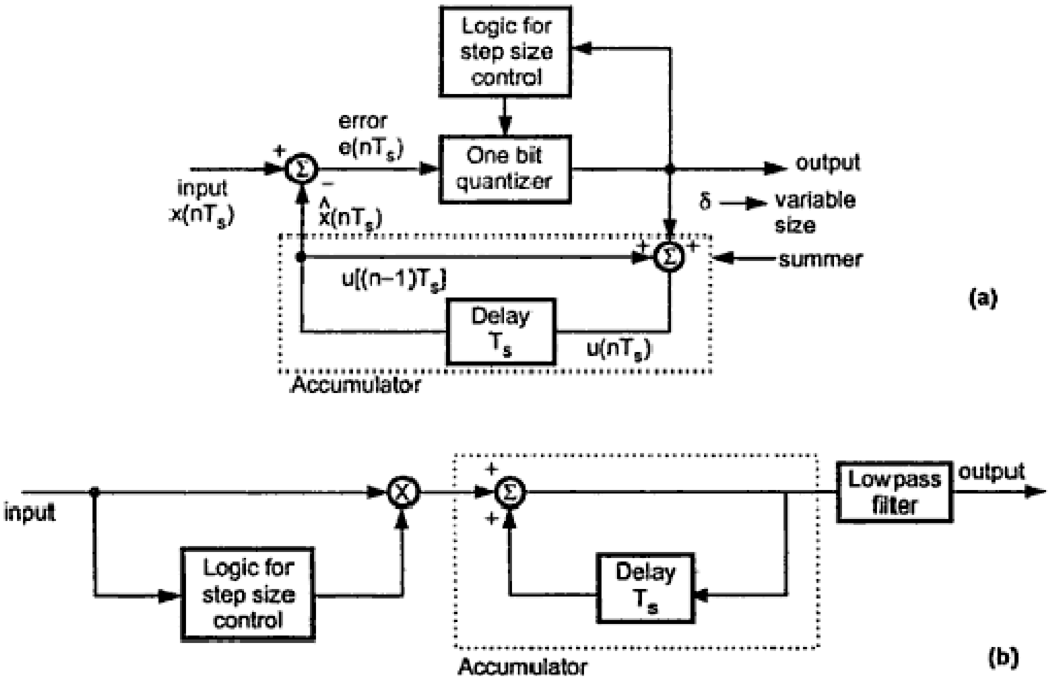


Fig. Adaptive delta modulator (a) Transmitter (b) Receiver

For example if one bit quantizer output is high (1), then step size may be doubled for next sample. If one bit quantizer output is low, then step size may be reduced by one step. Fig. shows the waveforms of adaptive delta modulator and sequence of bits transmitted.

In the receiver of adaptive delta modulator shown in Fig. (b) the first part generates the step size from each incoming bit. Exactly the same process is followed as that in transmitter. The previous input and present input decides the step size. It is then given to an accumulator which builds up staircase waveform. The low-pass filter then smoothens out the staircase waveform to reconstruct the smooth signal.

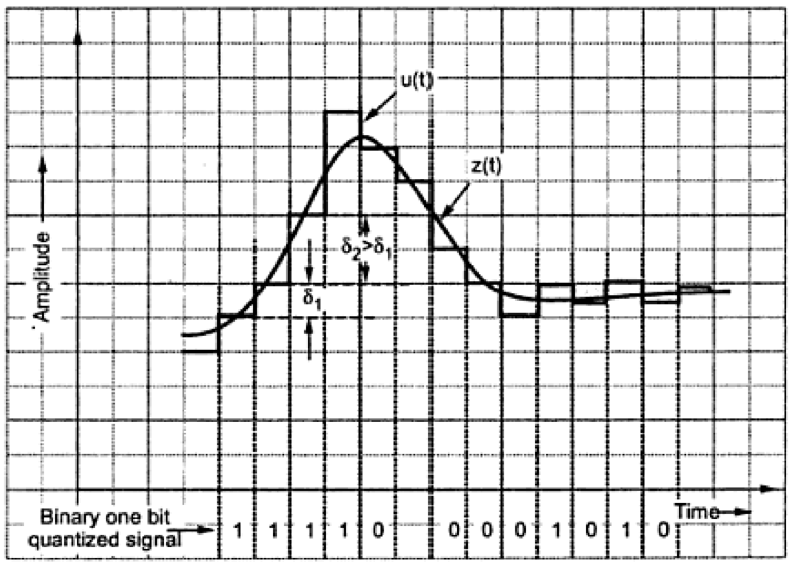


Fig. Waveforms of adaptive delta modulation

Advantages of Adaptive Delta Modulation

Adaptive delta modulation has certain advantages over delta modulation. i.e.,

1. The signal to noise ratio is better than ordinary delta modulation because of the reduction in slope overload distortion and granular noise.
2. Because of the variable step size, the dynamic range of ADM is wide.
3. Utilization of bandwidth is better than delta modulation.

Plus other advantages of delta modulation are, only one bit per sample is required and simplicity of implementation of transmitter and receiver.

Condition for Slope overload distortion occurrence:

Slope overload distortion will occur if

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

where T_s is the sampling period.

Let the sine wave be represented as,

$$x(t) = A_m \sin(2\pi f_m t)$$

Slope of $x(t)$ will be maximum when derivative of $x(t)$ with respect to 't' will be maximum. The maximum slope of delta modulator is given

$$\text{Max. slope} = \frac{\text{Step size}}{\text{Sampling period}}$$

$$= \frac{\delta}{T_s} \qquad \text{.............(1)}$$

Slope overload distortion will take place if slope of sine wave is greater than slope of delta modulator i.e.

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

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

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

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

or

$$\boxed{A_m > \frac{\delta}{2\pi f_m T_s}} \qquad \text{...............(2)}$$

Expression for Signal to Quantization Noise power ratio for Delta Modulation:

To obtain signal power :

slope overload distortion will not occur if

$$A_m \leq \frac{\delta}{2\pi f_m T_s}$$

Here A_m is peak amplitude of sinusoided signal

δ is the step size

f_m is the signal frequency and

T_s is the sampling period.

From above equation, the maximum signal amplitude will be,

$$A_m = \frac{\delta}{2\pi f_m T_s} \quad \text{..................(1)}$$

Signal power is given as,

$$P = \frac{V^2}{R}$$

Here V is the rms value of the signal. Here $V = \frac{A_m}{\sqrt{2}}$. Hence above equation becomes,

$$P = \left(\frac{A_m}{\sqrt{2}}\right)^2 / R$$

Normalized signal power is obtained by taking R = 1. Hence,

$$P = \frac{A_m^2}{2}$$

Putting for A_m from equation 1

$$P = \frac{\delta^2}{8\pi^2 f_m^2 T_s^2} \quad \text{.................(2)}$$

This is an expression for signal power in delta modulation.

(ii) To obtain noise power

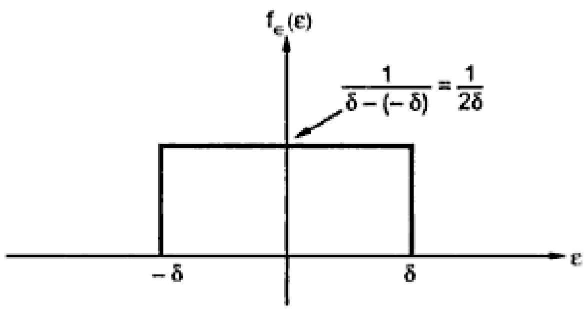


Fig. Uniform distribution of quantization error

We know that the maximum quantization error in delta modulation is equal to step size 'δ'. Let the quantization error be uniformly distributed over an interval $[-\delta, \delta]$. This is shown in Fig. From this figure the PDF of quantization error can be expressed as,

$$f_\epsilon(\varepsilon) = \begin{cases} 0 & for \quad \varepsilon < \delta \\ \frac{1}{2\delta} & for \quad -\delta < \varepsilon < \delta \\ 0 & for \quad \varepsilon > \delta \end{cases} \qquad \ldots\ldots\ldots\ldots\ldots(3)$$

The noise power is given as,

$$\text{Noise power} = \frac{V_{noise}^2}{R}$$

Here V_{noise}^2 is the mean square value of noise voltage. Since noise is defined by random variable 'ε' and PDF $f_\epsilon(\varepsilon)$, its mean square value is given as,

$$\text{mean square value} = E[\varepsilon^2] = \overline{\varepsilon^2}$$

mean square value is given as,

$$E[\varepsilon^2] = \int_{-\infty}^{\infty} \varepsilon^2 f_\epsilon(\varepsilon)\, d\varepsilon$$

From equation 3

$$E[\varepsilon^2] = \int_{-\delta}^{\delta} \varepsilon^2 \cdot \frac{1}{2\delta} d\varepsilon$$

$$= \frac{1}{2\delta}\left[\frac{\varepsilon^3}{3}\right]_{-\delta}^{\delta}$$

$$= \frac{1}{2\delta}\left[\frac{\delta^3}{3} + \frac{\delta^3}{3}\right] = \frac{\delta^2}{3} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots(4)$$

Hence noise power will be,

$$\text{noise power} = \left(\frac{\delta^2}{3}\right) / R$$

Normalized noise power can be obtained with R = 1. Hence,

$$\text{noise power} = \frac{\delta^2}{3} \qquad \ldots\ldots\ldots\ldots\ldots(5)$$

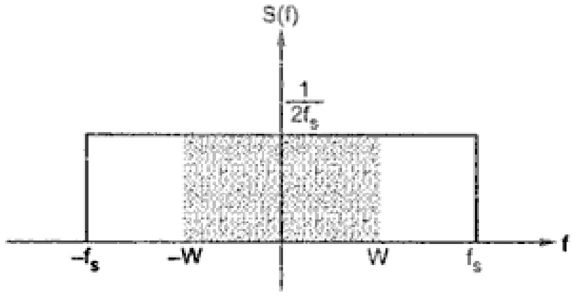


Fig. PSD of noise

This noise power is uniformly distributed over $-f_s$ to f_s range. This is illustrated in Fig. At the output of delta modulator receiver there is lowpass reconstruction filter whose cutoff frequency is 'W'. This cutoff frequency is equal to highest signal frequency. The reconstruction filter passes part of the noise power at the output as Fig. From the geometry of Fig. , output noise power will be,

$$\text{Output noise power} = \frac{W}{f_s} \times \text{noise power} = \frac{W}{f_s} \times \frac{\delta^2}{3}$$

We know that $f_s = \frac{1}{T_s}$, hence above equation becomes,

$$\text{Output noise power} = \frac{WT_s\delta^2}{3} \qquad \text{.................(6)}$$

(iii) To obtain signal to noise power ratio

Signal to noise power ratio at the output of delta modulation receiver is given as,

$$\frac{S}{N} = \frac{\text{Normalized signal power}}{\text{Normalized noise power}}$$

From equation 2. and equation 6

$$\frac{S}{N} = \frac{\frac{\delta^2}{8\pi^2 f_m^2 T_s^2}}{\frac{WT_s\delta^2}{3}}$$

$$\boxed{\frac{S}{N} = \frac{3}{8\pi^2 W f_m^2 T_s^3}} \qquad \text{..............(7)}$$

This is an expression for signal to noise power ratio in delta modulation.

comparison of all modulations:

S. No.	Parameter of comparison	Pulse Code Modulation (PCM)	Delta modulation (DM)	Adaptive Delta Modulation (ADM)	Differential Pulse Code Modulation (DPCM)
1.	Number of bits.	It can use 4, 8 or 16 bits per sample.	It uses only one bit for one sample.	Only one bit is used to encode one sample.	Bits can be more than one but are less than PCM.
2.	Levels and step size	The number of levels depend on number of bits. Level size is kept fixed.	Step size is kept fixed and cannot be varied.	According to the signal variation, step size varies (*i.e.* Adapted).	Here, Fixed number of levels are used.
3.	Quantization error and distortion	Quantization error depends on number of levels used.	Slope overload distor-tion and granular noise are present.	Quantization noise is present but other errors are absent.	Slope overload distor-tion and quantization noise is present.
4.	Transmission bandwidth	Highest bandwidth is required since number of bits are high	Lowest bandwidth is required.	Lowest bandwidth is required.	Bandwidth required is lower than PCM.
5.	Feedback	There is no feedback in transmitter or receiver.	Feedback exists in transmitter.	Feedback exists.	Here, Feedback exists.
6.	Complexity of implementation	System complex.	Simple.	Simple.	Simple

UNIT2
BASEBAND PULSE TRANSMISSION

Introduction

Consider that a binary encoded signal consists of a time sequence of voltage levels +Vor–V. If there is a guard interval between the bits, the signal forms a sequence of positive andnegative pulses, in either case there is no particular interest in preserving the waveform of the signal after reception .we are interested only in knowing within each bit interval whether the transmitted voltage was +V or –V. With noise present, the receives signal and noise togetherwill yield sample values generally different from ±V. In this case, what deduction shall we make from the sample value concerning the transmitted bit.

- Suppose that the noise is gaussian and therefore the noise voltage has a probability density which is entirely symmetrical with respect to zero volts. Then the probability that the noise has increased the sample value is the same as the probability that the noise has decreased the sample value. It then seems entirely reasonable that we can do no better than to assume that if the sample value is positive the transmitted level was $+V$, and if the sample value is negative the transmitted level was $-V$. It is, of course, possible that at the sampling time the noise voltage may be of magnitude larger than V and of a polarity opposite to the polarity assigned to the transmitted bit. In this case an error will be made as indicated in Fig. 11.1-1. Here the transmitted bit is represented by the voltage $+V$ which is sustained over an interval T from t_1 to t_2. Noise has been superimposed on the level $+V$ so that the voltage v represents the received signal and noise. If now the sampling should happen to take place at a time $t = t_1 + \Delta t$, an error will have been made.

We can reduce the probability of error by processing the received signal plus noise in such a manner that we are then able to find a sample time where the sample voltage due to the signal is emphasized relative to the sample voltage due to the noise. Such a processer (receiver) is shown in Fig. 11.1-2. The signal input during a bit interval is indicated. As a matter of convenience we have set $t = 0$ at the beginning of the interval. The waveform of the signal $s(t)$ before $t = 0$ and after $t = T$ has not been indicated since, as will appear, the operation of the receiver during each bit interval is independent of the waveform during past and future bit intervals.

The signal $s(t)$ with added white gaussian noise $n(t)$ of power spectral density $\eta/2$ is presented to an integrator. At time $t = 0+$ we require that capacitor C be uncharged. Such a discharged condition may be ensured by a brief closing of switch SW_1 at time $t = 0-$, thus relieving C of any charge it may have acquired during the previous interval. The sample is taken at the output of the integrator by closing this sampling switch SW_2. This sample is taken at the end of the bit interval, at $t = T$. The signal processing indicated in Fig. 11.1-2 is described by the phrase *integrate and dump*, the term *dump* referring to the abrupt discharge of the capacitor after each sampling.

The probability of error p_e, as given in eq.(11.2-3),is plotted in fig.11.2-2.note that p_e decreases rapidly as E_s/η increases. The maximum value of p_e is ½.thus ,even if the signal is entirely lost in the noise so that any determination of the receiver is a sheer guess, the receiver cannot bi wrong more than half the time on the average.

THE OPTIMUM FILTER:

In the receiver system of Fig 11.1-2, the signal was passed through a filter(integrator),so that at the sampling time the signal voltage might be emphasized in comparison with the noise voltage. We are naturally led to risk whether the integrator is the optimum filter for the purpose of minimizing the probability of error. We shall find that the received signal contemplated in system of fig 11.1-2 the integrator is indeed the optimum filter. However, before returning specifically to the integrator receiver.

We assume that the received signal is a binary waveform. One binary digit is represented by a signal waveformS_1 (t) which persists for time T, while the4 other bit is represented by the waveform $S_2(t)$ which also lasts for an interval T. For example, in the transmission at baseband, as shown in fig 11.1-2 $S_1(t)=+V$; for other modulation systems, different waveforms are transmitted. for example for PSK signaling , $S_1(t)=A\cos w_0t$ and $S_2(t)=-A\cos w_0t$;while for FSK, $S_1(t)=A\cos(w_{0+\Omega})t$.

As shown in Fig. 11.3-1 the input, which is $s_1(t)$ or $s_2(t)$, is corrupted by the addition of noise $n(t)$. The noise is gaussian and has a spectral density $G(f)$. [In most cases of interest the noise is white, so that $G(f) = \eta/2$. However, we shall assume the more general possibility, since it introduces no complication to do so.] The signal and noise are filtered and then sampled at the end of each bit interval. The output sample is either $v_o(T) = s_{o1}(T) + n_o(T)$ or $v_o(T) = s_{o2}(T) + n_o(T)$. We assume that immediately after each sample, every energy-storing element in the filter has been discharged.

We have already considered in Sec. 2.22, the matter of signal determination in the presence of noise. Thus, we note that in the absence of noise the output sample would be $v_o(T) = s_{o1}(T)$ or $s_{o2}(T)$. When noise is present we have shown that to minimize the probability of error one should assume that $s_1(t)$ has been transmitted if $v_o(T)$ is *closer* to $s_{o1}(T)$ than to $s_{o2}(T)$. Similarly, we assume $s_2(t)$ has been transmitted if $v_o(T)$ is *closer* to $s_{o2}(T)$. The decision boundary is therefore midway between $s_{o1}(T)$ and $s_{o2}(T)$. For example, in the baseband system of Fig. 11.1-2, where $s_{o1}(T) = VT/\tau$ and $s_{o2}(T) = -VT/\tau$, the decision boundary is $v_o(T) = 0$. In general, we shall take the decision boundary to be

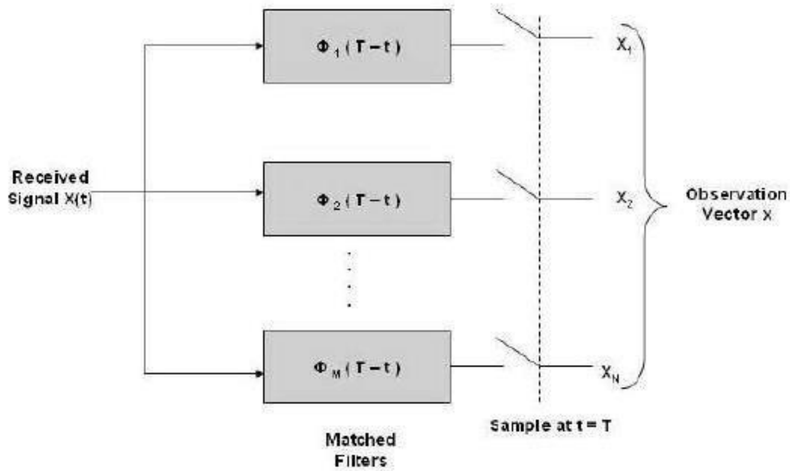
$$v_o(T) = \frac{s_{o1}(T) + s_{o2}(T)}{2} \qquad (11.3\text{-}1)$$

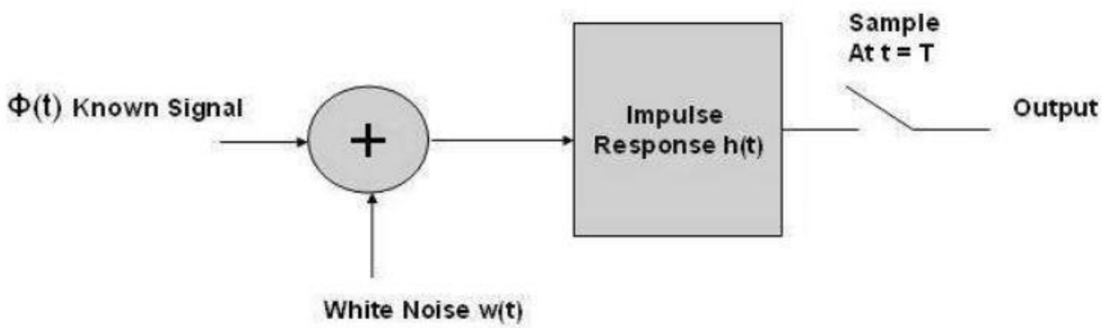
The probability of error for this general case may be deduced as an extension of the considerations used in the baseband case. Suppose that $s_{o1}(T) > s_{o2}(T)$ and that $s_2(t)$ was transmitted. If, at the sampling time, the noise $n_o(T)$ is positive and larger in magnitude than the voltage difference $\frac{1}{2}[s_{o1}(T) + s_{o2}(T)] - s_{o2}(T)$, an error will have been made. That is, an error [we decide that $s_1(t)$ is transmitted rather than $s_2(t)$] will result if

$$n_o(T) \geq \frac{s_{o1}(T) - s_{o2}(T)}{2} \qquad (11.3\text{-}2)$$

Matchedfilter

A filter whose impulse response is time-reversed and delayed version of the input signal is saidto be matched. Correspondingly , the optimum receiver based on this is referred as the **matched filterreceiver**.





Φ(t)=inputsignal

h(t)=impulseresponse

W(t)=white noise

Theimpulseresponseofthematchedfilteristime-reversedanddelayedversionofthe

inputsignal.
Properties of Matchedfilter

PROPERTY1:The spectrum of the output signal of a matched filter with the matched signal as inputis,except fo r a time delayfactor,proportional to the energy spectral density of the input signal.
PROPERTY2:The output signal of a Matched Filter is proportional to a shifted version of the autocorrelation function of the input signal to which the filter is matched.
PROPERTY3:The output Signal to Noise Ratio of a Matched filter depends onlyon the ratio of the signal energy to the power spectral density of the white noise at the filter input.
PROPERTY 4: The Matched Filtering operation may be separated into two matching conditions ;namely spectral phase matching that produces the desired output peak at time T,and the spectral amplitude matching that gives this peak value its optimum signal to noise density ratio.

THEOPTIMUMFILTER:

In the receiver system of Fig 11.1-2, the signal was passed through a filter(integrator),so that at thesamplingtimethesignalvoltagemightbeemphasizedincomparisonwiththe noisevoltage.Wearenaturally led to risk whether the integrator is the optimum filter for the purpose of minimizing theprobability of error. We shall find that the received signal contemplated in system of fig 11.1-2 theintegrator is indeed the optimum filter. However, before returning specifically to the integratorreceiver.

We assume that the received signal is a binary waveform. One binary digit is representedby a signal waveformS_1 (t) which persists for time T, while the4 other bit is represented by thewaveformS_2(t)whichalso lasts foranintervalT.Forexample,inthetransmissionatbaseband,asshown in fig 11.1-2 S_1(t)=+V; for other modulation systems, different waveforms are transmitted.forexample forPSKsignaling,S_1(t)=Acosw$_0$tandS_2(t)=-Acosw$_0$t;whileforFSK,S_1(t)=Acos(w$_{0+\Omega)}$t.

$$v_o(T) = \frac{s_{o1}(T) + s_{o2}(T)}{2} \qquad (11.3\text{-}1)$$

Henceprobabilityofferroris

$$P_e = \int_{[s_{o1}(T)-s_{o2}(T)]/2}^{\infty} \frac{e^{-n_o^2(T)/2\sigma_o^2}}{\sqrt{2\pi\sigma_o^2}}\, dn_o(T) \qquad (11.3\text{-}3)$$

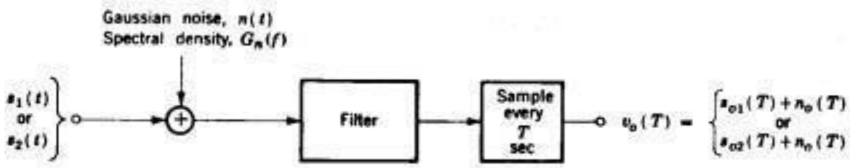


Figure 11.3-1 A receiver for binary coded signalling.

If we make the substitution $x \equiv n_o(T)/\sqrt{2}\sigma_o$, Eq. (11.3-3) becomes

$$P_e = \frac{1}{2}\frac{2}{\sqrt{\pi}} \int_{[s_{o1}(T)-s_{o2}(T)]/2\sqrt{2}\sigma_o}^{\infty} e^{-x^2}\, dx \qquad (11.3\text{-}4a)$$

$$P_e = \frac{1}{2}\,\text{erfc}\left[\frac{s_{o1}(T)-s_{o2}(T)}{2\sqrt{2}\sigma_o}\right] \qquad (11.3\text{-}4b)$$

Note that for the case $s_{o1}(T) = VT/\tau$ and $s_{o2}(T) = -VT/\tau$, and, using Eq. (11.1-4), Eq. (11.3-4b) reduces to Eq. (11.2-3) as expected.

The complementary error function is a monotonically decreasing function of its argument. (See Fig. 11.2-2.) Hence, as is to be anticipated, P_e decreases as the difference $s_{o1}(T) - s_{o2}(T)$ becomes larger and as the rms noise voltage σ_o becomes smaller. The optimum filter, then, is the filter which maximizes the ratio

$$\gamma = \frac{s_{o1}(T)-s_{o2}(T)}{\sigma_o} \qquad (11.3\text{-}5)$$

We now calculate the transfer function $H(f)$ of this optimum filter. As a matter of mathematical convenience we shall actually maximize γ^2 rather than γ.

Calculation of the Optimum-Filter Transfer Function $H(f)$

The fundamental requirement we make of a binary encoded data receiver is that it distinguishes the voltages $s_1(t) + n(t)$ and $s_2(t) + n(t)$. We have seen that the ability of the receiver to do so depends on how large a particular receiver can make γ. It is important to note that γ is proportional not to $s_1(t)$ nor to $s_2(t)$, but rather to the *difference* between them. For example, in the baseband system we represented the signals by voltage levels $+V$ and $-V$. But clearly, if our only interest was in distinguishing levels, we would do just as well to use $+2$ volts and 0 volt, or $+8$ volts and $+6$ volts, etc. (The $+V$ and $-V$ levels, however, have the advantage of requiring the least average power to be transmitted.) Hence, while $s_1(t)$ or $s_2(t)$ is the received signal, the signal which is to be compared with the noise, i.e., the signal which is relevant in all our error-probability calculations, is the difference signal

$$p(t) \equiv s_1(t) - s_2(t) \qquad (11.3\text{-}6)$$

Thus, for the purpose of calculating the minimum error probability, we shall assume that the input signal to the optimum filter is $p(t)$. The corresponding *output signal* of the filter is then

$$p_o(t) \equiv s_{o1}(t) - s_{o2}(t) \qquad (11.3\text{-}7)$$

We shall let $P(f)$ and $P_o(f)$ be the Fourier transforms, respectively, of $p(t)$ and $p_o(t)$.

If $H(f)$ is the transfer function of the filter,

$$P_o(f) = H(f)P(f) \tag{11.3-8}$$

and

$$p_o(T) = \int_{-\infty}^{\infty} P_o(f)e^{j2\pi fT}\,df = \int_{-\infty}^{\infty} H(f)P(f)e^{j2\pi fT}\,df \tag{11.3-9}$$

The input noise to the optimum filter is $n(t)$. The output noise is $n_o(t)$ which has a power spectral density $G_{n_o}(f)$ and is related to the power spectral density of the input noise $G_n(f)$ by

$$G_{n_o}(f) = |H(f)|^2 G_n(f) \tag{11.3-10}$$

Using Parseval's theorem (Eq. 1.13-5), we find that the normalized output noise power, i.e., the noise variance σ_o^2, is

$$\sigma_o^2 = \int_{-\infty}^{\infty} G_{n_o}(f)\,df = \int_{-\infty}^{\infty} |H(f)|^2 G_n(f)\,df \tag{11.3-11}$$

From Eqs. (11.3-9) and (11.3-11) we now find that

$$\gamma^2 = \frac{p_o^2(T)}{\sigma_o^2} = \frac{|\int_{-\infty}^{\infty} H(f)P(f)e^{j2\pi Tf}\,df|^2}{\int_{-\infty}^{\infty} |H(f)|^2 G_n(f)\,df} \tag{11.3-12}$$

Equation (11.3-12) is unaltered by the inclusion or deletion of the absolute value sign in the numerator since the quantity within the magnitude sign $p_o(T)$ is a positive real number. The sign has been included, however, in order to allow further development of the equation through the use of the *Schwarz inequality*.

The *Schwarz inequality* states that given arbitrary complex functions $X(f)$ and $Y(f)$ of a common variable f, then

$$\left|\int_{-\infty}^{\infty} X(f)Y(f)\,df\right|^2 \le \int_{-\infty}^{\infty} |X(f)|^2\,df \int_{-\infty}^{\infty} |Y(f)|^2\,df \tag{11.3-13}$$

The equal sign applies when

$$X(f) = KY^*(f) \tag{11.3-14}$$

where K is an arbitrary constant and $Y^*(f)$ is the complex conjugate of $Y(f)$.

We now apply the Schwarz inequality to Eq. (11.3-12) by making the identification

$$X(f) \equiv \sqrt{G_n(f)}\,H(f) \tag{11.3-15}$$

and

$$Y(f) \equiv \frac{1}{\sqrt{G_n(f)}} P(f)e^{j2\pi Tf} \tag{11.3-16}$$

Using Eqs. (11.3-15) and (11.3-16) and using the Schwarz inequality, Eq. (11.3-13), we may rewrite Eq. (11.3-12) as

$$\frac{p_o^2(T)}{\sigma_o^2} = \frac{|\int_{-\infty}^{\infty} X(f)Y(f)\,df|^2}{\int_{-\infty}^{\infty} |X(f)|^2\,df} \le \int_{-\infty}^{\infty} |Y(f)|^2\,df \tag{11.3-17}$$

or, using Eq. (11.3-16),

$$\frac{p_o^2(T)}{\sigma_o^2} \le \int_{-\infty}^{\infty} |Y(f)|^2 \, df = \int_{-\infty}^{\infty} \frac{|P(f)|^2}{G_n(f)} \, df \qquad (11.3\text{-}18)$$

The ratio $p_o^2(T)/\sigma_o^2$ will attain its maximum value when the equal sign in Eq. (11.3-18) may be employed as is the case when $X(f) = KY^*(f)$. We then find from Eqs. (11.3-15) and (11.3-16) that the optimum filter which yields such a maximum ratio $p_o^2(T)/\sigma_o^2$ has a transfer function

$$H(f) = K \frac{P^*(f)}{G_n(f)} e^{-j2\pi fT} \qquad (11.3\text{-}19)$$

Correspondingly, the maximum ratio is, from Eq. (11.3-18),

$$\left[\frac{p_o^2(T)}{\sigma_o^2}\right]_{\max} = \int_{-\infty}^{\infty} \frac{|P(f)|^2}{G_n(f)} \, df \qquad (11.3\text{-}20)$$

In succeeding sections we shall have occasion to apply Eqs. (11.13-19) and (11.13-20) to a number of cases of interest.

11.4 WHITE NOISE: THE MATCHED FILTER

An optimum filter which yields a maximum ratio $p_o^2(T)/\sigma_o^2$ is called a *matched filter* when the input noise is *white*. In this case $G_n(f) = \eta/2$, and Eq. (11.3-19) becomes

$$H(f) = K \frac{P^*(f)}{\eta/2} e^{-j2\pi fT} \qquad (11.4\text{-}1)$$

The impulsive response of this filter, i.e., the response of the filter to a unit strength impulse applied at $t = 0$, is

$$h(t) = \mathscr{F}^{-1}[H(f)] = \frac{2K}{\eta} \int_{-\infty}^{\infty} P^*(f) e^{-j2\pi fT} e^{j2\pi ft} \, df \qquad (11.4\text{-}2a)$$

$$= \frac{2K}{\eta} \int_{-\infty}^{\infty} P^*(f) e^{j2\pi f(t-T)} \, df \qquad (11.4\text{-}2b)$$

A physically realizable filter will have an impulse response which is real, i.e., not complex. Therefore $h(t) = h^*(t)$. Replacing the right-hand member of Eq. (11.4-2*b*) by its complex conjugate, an operation which leaves the equation unaltered, we have

$$h(t) = \frac{2K}{\eta} \int_{-\infty}^{\infty} P(f) e^{j2\pi f(T-t)} \, df \qquad (11.4\text{-}3a)$$

$$= \frac{2K}{\eta} p(T-t) \qquad (11.4\text{-}3b)$$

Finally, since $p(t) \equiv s_1(t) - s_2(t)$ [see Eq. (11.3-6)], we have

$$h(t) = \frac{2K}{\eta} [s_1(T-t) - s_2(T-t)] \qquad (11.4\text{-}4)$$

The significance of these results for the matched filter may be more readily appreciated by applying them to a specific example. Consider then, as in Fig. 11.4-1*a*, that $s_1(t)$ is a triangular waveform of duration T, while $s_2(t)$, as shown in Fig. 11.4-1*b*, is of identical form except of reversed polarity. Then $p(t)$ is as shown in Fig. 11.4-1*c*, and $p(-t)$ appears in Fig. 11.4-1*d*. The waveform $p(-t)$ is the waveform $p(t)$ rotated around the axis $t = 0$. Finally, the waveform $p(T - t)$ called for as the impulse response of the filter in Eq. (11.4-3*b*) is this rotated waveform $p(-t)$ translated in the positive t direction by amount T. This last translation ensures that $h(t) = 0$ for $t < 0$ as is required for a *causal* filter.

In general, the impulsive response of the matched filter consists of p(t) rotated about t=0andthendelayedlongenough(i.e.,atimeT)tomakethefilterrealizable.Wemaynoteinpassing,that any additional delay that a filter might introduce would in no way interfere with theperformance of the filter ,for both signal and noise would be delayed by the same amount, and atthe sampling time (which would need similarity to be delayed)the ratio of signal to noise wouldremainunaltered.

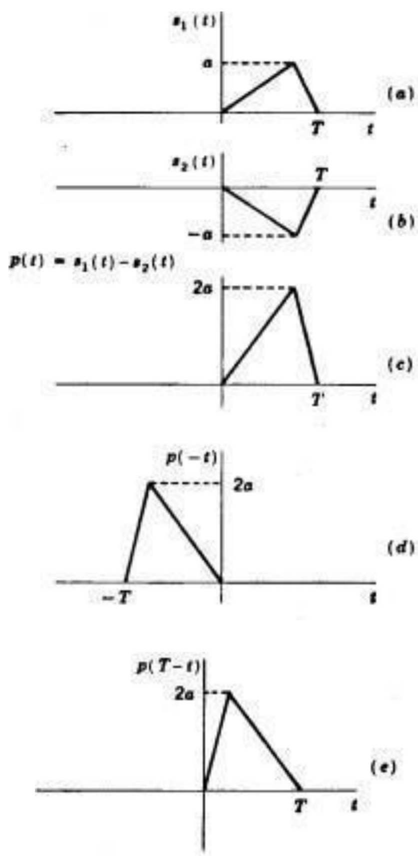


Figure 11.4-1 The signals (*a*) $s_1(t)$, (*b*) $s_2(t)$, and (*c*) $p(t) = s_1(t) - s_2(t)$. (*d*) $p(t)$ rotated about the axis $t = 0$. (*e*) The waveform in (*d*) translated to the right by amount T.

11.5 PROBABILITY OF ERROR OF THE MATCHED FILTER

The probability of error which results when employing a matched filter, may be found by evaluating the maximum signal-to-noise ratio $[p_o^2(T)/\sigma_o^2]_{max}$ given by Eq. (11.3-20). With $G_n(f) = \eta/2$, Eq. (11.3-20) becomes

$$\left[\frac{p_o^2(T)}{\sigma_o^2}\right]_{max} = \frac{2}{\eta}\int_{-\infty}^{\infty} |P(f)|^2\, df \qquad (11.5\text{-}1)$$

From parseval's theorem we have

$$\int_{-\infty}^{\infty} |P(f)|^2\, df = \int_{-\infty}^{\infty} p^2(t)\, dt = \int_0^T p^2(t)\, dt \qquad (11.5\text{-}2)$$

In the last integral in Eq. (11.5-2), the limits take account of the fact that $p(t)$ persists for only a time T. With $p(t) = s_1(t) - s_2(t)$, and using Eq. (11.5-2), we may write Eq. (11.5-1) as

$$\left[\frac{p_o^2(T)}{\sigma_o^2}\right]_{max} = \frac{2}{\eta}\int_0^T [s_1(t) - s_2(t)]^2\, dt \qquad (11.5\text{-}3a)$$

$$= \frac{2}{\eta}\left[\int_0^T s_1^2(t)\, dt + \int_0^T s_2^2(t)\, dt - 2\int_0^T s_1(t)s_2(t)\, dt\right. \qquad (11.5\text{-}3b)$$

$$= \frac{2}{\eta}(E_{s1} + E_{s2} - 2E_{s12}) \qquad (11.5\text{-}3c)$$

Here E_{s1} and E_{s2} are the energies, respectively, in $s_1(t)$ and $s_2(t)$, while E_{s12} is the energy due to the correlation between $s_1(t)$ and $s_2(t)$.

Suppose that we have selected $s_1(t)$, and let $s_1(t)$ have an energy E_{s1}. Then it can be shown that if $s_2(t)$ is to have the *same energy*, the optimum choice of $s_2(t)$ is

$$s_2(t) = -s_1(t) \qquad (11.5\text{-}4)$$

The choice is optimum in that it yields a maximum output signal $p_o^2(T)$ for a given signal energy. Letting $s_2(t) = -s_1(t)$, we find

$$E_{s1} = E_{s2} = -E_{s12} \equiv E_s$$

and Eq. (11.5-3c) becomes

$$\left[\frac{p_o^2(T)}{\sigma_o^2}\right]_{max} = \frac{8E_s}{\eta} \qquad (11.5\text{-}5)$$

Rewriting Eq. (11.3-4b) using $p_o(T) = s_{o1}(T) - s_{o2}(T)$, we have

$$P_e = \frac{1}{2}\,\text{erfc}\left[\frac{p_o(T)}{2\sqrt{2}\,\sigma_o}\right] = \frac{1}{2}\,\text{erfc}\left[\frac{p_o^2(T)}{8\sigma_o^2}\right]^{1/2} \qquad (11.5\text{-}6)$$

Combining Eq. (11.5-6) with (11.5-5), we find that the minimum error probability $(P_e)_{min}$ corresponding to a maximum value of $p_o^2(T)/\sigma_o^2$ is

$$(P_e)_{min} = \frac{1}{2}\,\text{erfc}\left\{\frac{1}{8}\left[\frac{p_o^2(T)}{\sigma_o^2}\right]_{max}\right\}^{1/2} \qquad (11.5\text{-}7)$$

$$= \frac{1}{2}\,\text{erfc}\left(\frac{E_s}{\eta}\right)^{1/2} \qquad (11.5\text{-}8)$$

We note that Eq. (11.5-8) establishes more generally the idea that the error probability depends only on the signal energy and not on the signal waveshape. Previously we had established this point only for signals which had constant voltage levels.

We note also that Eq. (11.5-8) gives $(P_e)_{min}$ for the case of the matched filter and when $s_1(t) = -s_2(t)$. In Sec. 11.2 we considered the case when $s_1(t) = +V$ and $s_2(t) = -V$ and the filter employed was an integrator. There we found [Eq. (11.2-3)] that the result for P_e was identical with $(P_e)_{min}$ given in Eq. (11.5-8). This agreement leads us to suspect that for an input signal where $s_1(t) = +V$ and $s_2(t) = -V$, the integrator is the matched filter. Such is indeed the case. For when we have

$$s_1(t) = V \qquad 0 \le t \le T \qquad (11.5\text{-}9a)$$

$$s_2(t) = \qquad (11.5\text{-}9b)$$

the impulse r $s_o(T) = \frac{1}{\tau}\int_0^T s_i(t)[s_1(t) - s_2(t)]\,dt$ rom Eq. (11.4-4), *(11.6-1)*

$$h(t) = \frac{2K}{\eta}[s_1(T-t) - s_2(T-t)] \qquad (11.5\text{-}10)$$

The quantity $s_1(T-t) - s_2(T-t)$ is a pulse of amplitude $2V$ extending from $t = 0$ to $t = T$ and may be rewritten, with $u(t)$ the unit step,

$$h(t) = \frac{2K}{\eta}(2V)[u(t) - u(t-T)] \qquad (11.5\text{-}11)$$

The constant factor of proportionality $4KV/\eta$ in the expression for $h(t)$ (that is, the gain of the filter) has no effect on the probability of error since the gain affects signal and noise alike. We may therefore select the coefficient K in Eq. (11.5-11) so that $4KV/\eta = 1$. Then the inverse transform of $h(t)$, that is, the transfer function of the filter, becomes, with s the Laplace transform variable,

$$H(s) = \frac{1}{s} - \frac{e^{-sT}}{s} \qquad (11.5\text{-}12)$$

The first term in Eq. (11.5-12) represents an integration beginning at $t = 0$, while the second term represents an integration with reversed polarity beginning at $t = T$. The overall response of the matched filter is an integration from $t = 0$ to $t = T$ and a zero response thereafter. In a physical system, as already described, we achieve the effect of a zero response after $t = T$ by sampling at $t = T$, so that so far as the determination of one bit is concerned we ignore the response after $t = T$.

COHERENT RECEPTION:CORRELATION:

We discuss now an alternative type of receiving system which, as we shall see, is identical in performance with the matched filter receiver. Again, as shown in Fig. 11.6-1, the input is a binary data waveform $s_1(t)$ or $s_2(t)$ corrupted by noise $n(t)$. The bit length is T. The received signal plus noise $v_i(t)$ is multiplied by a locally generated waveform $s_1(t) - s_2(t)$. The output of the multiplier is passed through an integrator whose output is sampled at $t = T$. As before, immediately after each sampling, at the beginning of each new bit interval, all energy-storing elements in the integrator are discharged. This type of receiver is called a *correlator*, since we are *correlating* the received signal and noise with the waveform $s_1(t) - s_2(t)$.

The output signal and noise of the correlator shown in Fig. 11.6-1 are

$$n_o(T) = \frac{1}{\tau}\int_0^T n(t)[s_1(t) - s_2(t)]\, dt$$

(11.6-2)

Where $s_1(t)$ is either $s_1(t)$or$s_2(t)$,and wthere π is the constant of the integrator(i.e.,the integratoroutput is 1/π times the integral of its input).we now compare these outputs with the matchedfilter outputs.

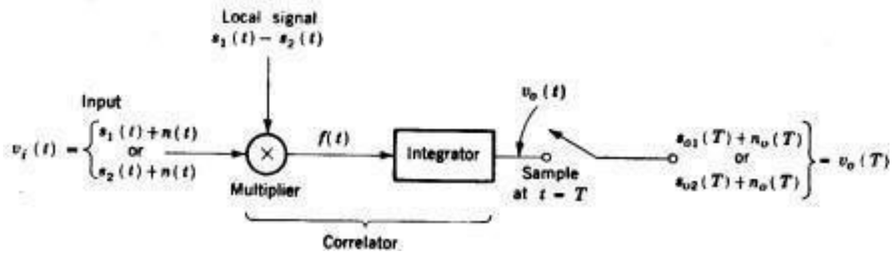


Fig:11.6-1Coherentsystemofsignalreception

Ifh(t)isthe impulsiveresponseofthematched filter,thenthe outputofthe matchedfilter*$v_0(t)$* canbefoundusingtheconvolutionintegral.wehave

$$v_o(t) = \int_{-\infty}^{\infty} v_i(\lambda)h(t-\lambda)\, d\lambda = \int_0^T v_i(\lambda)h(t-\lambda)\, d\lambda$$

(11.6-3)

The limits on the integral have been charged to 0 and T since we are interested in the filterresponse to a bit which extends only over that interval. Using Eq.(11.4-4) which gives h(t) for thematchedfilter,wehave

$$h(t) = \frac{2K}{\eta}[s_1(T-t) - s_2(T-t)]$$

(11.6-4)

so that
$$h(t-\lambda) = \frac{2K}{\eta}[s_1(T-t+\lambda) - s_2(T-t+\lambda)]$$

(11.6-5)

sub11.6-5in11.6-3

$$v_o(t) = \frac{2K}{\eta}\int_0^T v_i(\lambda)[s_1(T-t+\lambda) - s_2(T-t+\lambda)]\, d\lambda$$

Since $v_i(\lambda) = s_i(\lambda) + n(\lambda)$, and $v_o(t) = s_o(t) + n_o(t)$, setting $t = T$ yields

$$s_o(T) = \frac{2K}{\eta}\int_0^T s_i(\lambda)[s_1(\lambda) - s_2(\lambda)]\, d\lambda$$

where $s_i(\lambda)$ is equal to $s_1(\lambda)$ or $s_2(\lambda)$. Similarly we find that

$$n_o(T) = \frac{2K}{\eta}\int_0^T n(\lambda)[s_1(\lambda) - s_2(\lambda)]\, d\lambda$$

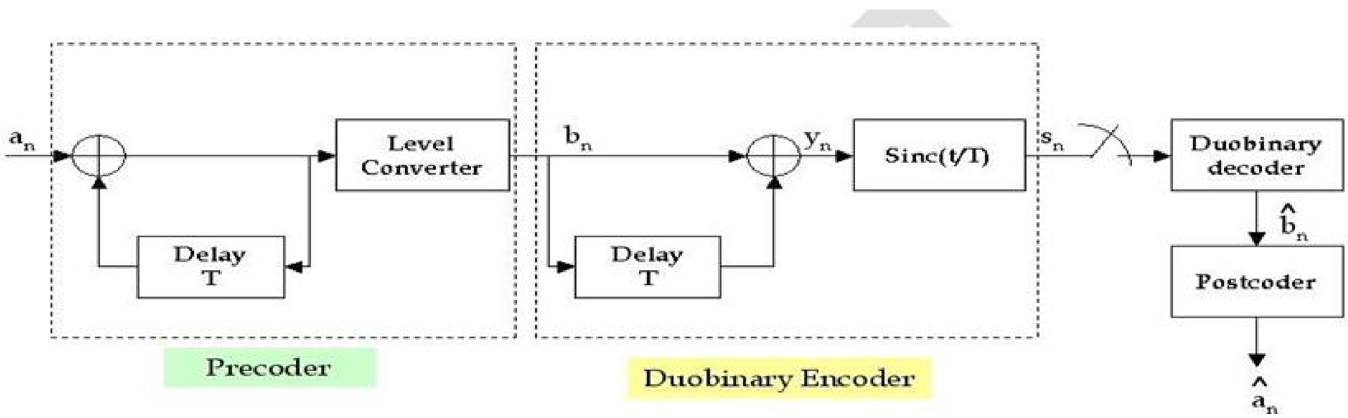
Nyquist's criterion for distortionless baseband binary transmission:

Raised cosine response meets the Nyquist ISI criterion. Consecutive raised-cosine impulses demonstrate the zero ISI property between transmitted symbols at the sampling instants. Att=0 the middle pulse is at its maximum and the sum of other impulsesis zero.

In communications, the Nyquist ISI criterion describes the conditions which, when satisfied by acommunication channel (including responses of transmit and receive filters), result in no inter symbol interference or ISI. It provides a method for constructing band-limited functions to overcome the effects of inter symbol interference.

Correlative coding-duo binary

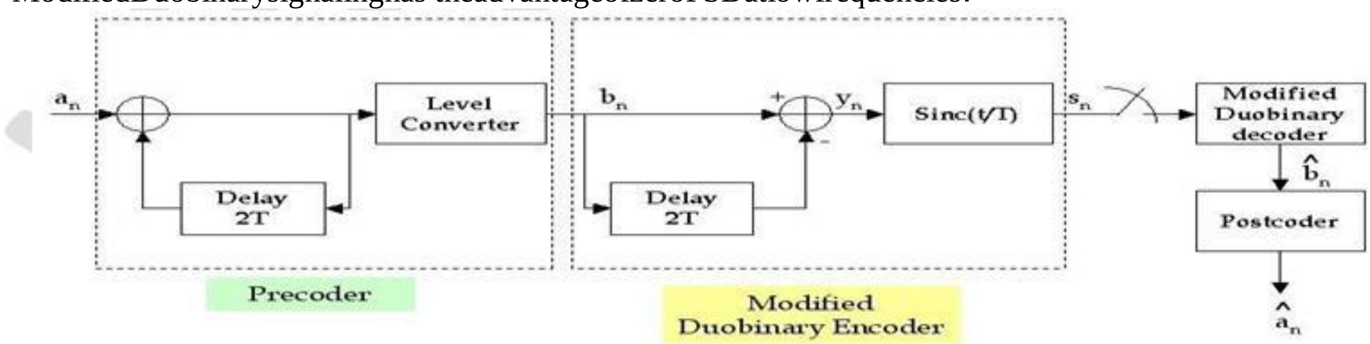
The following figure shows the duo binary signaling scheme



The receiver consists of duo binary decoder and a post coder(invert of pre coder)

Modified duo binary signaling scheme:

Modified Duo binary Signaling is an extension of duo binary signaling
ModifiedDuobinarysignalinghas theadvantageofzeroPSDatlowfrequencies.



Modified DuobinarySignaling is an extension of duobinarysignaling. It has the advantage of zero PSD at low frequencies (especially at DC) that is suitable for channels with poor DC response. It correlates two symbols that are 2T time instants apart, whereas in duobinarysignaling, symbols that are 1T apart are correlated.
The general condition to achieve zero ISI is given by

$$p(nT) = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases}$$

As discussed in a previous article, in correlative coding , the requirement of zero ISI condition is relaxed as a controlled amount of ISI is introduced in the transmitted signal and is counteracted in the receiver side

In the case of modified duobinarysignaling, the above equation is modified as

$$p(nT) = \begin{cases} 1, & n = 0, 2 \\ 0, & otherwise \end{cases}$$

which states that the ISI is limited to two alternate samples. Here a controlled or "deterministic" amount of ISI is introduced and hence its effect can be removed upon signal detection at the receiver.

Encoding Process:

1) a_n = binary input bit; $a_n \in \{0,1\}$.

2) b_n = NRZ polar output of Level converter in the precoder and is given by,

$$b_n = \begin{cases} -d, & if\ a_k = 0 \\ +d, & if\ a_k = 1 \end{cases}$$

wherea_k is the precoded output (before level converter).

3) y_n can be represented as

Note that the samples b_n are uncorrelated (i.e either +d for "1" or -d for "0" input). On the other-hand,the samples y_n are correlated (i.e. there are three possible values +2d,0,-2d depending on ak and a_{k-2}). Meaning that the modified duobinary encoding correlates present sample a_k and the previous input sample a_{k-2}.

4) From the diagram,impulse response of the modified duobinary encoder is computed as

$$h(t) = sinc\left(\frac{t}{T}\right) - sinc\left(\frac{t-2T}{T}\right)$$

Decoding Process:

5) The receiver consists of a modified duobinary decoder and a postcoder (inverse of precoder). The decoder implements the following equation (which can be deduced from the equation given under step 3 (see above))

$$\hat{b}_n = y_n - \hat{b}_{n-2}$$

This equation indicates that the decoding process is prone to error propagation as the estimate of present sample relies on the estimate of previous sample. This error propagation is avoided by using a precoder before modified-duobinary encoder at the transmitter and a postcoder after the modified-duobinary decoder. The precoder ties the present sample and the sample that precedes the previous sample (correlates these two samples) and the postcoder does the reverse process.

The entire process of modified-duobinary decoding and the postcoding can be combined together as one algorithm. The following decision rule is used for detecting the original modified-duobinary signal samples $\{a_n\}$ from $\{y_n\}$

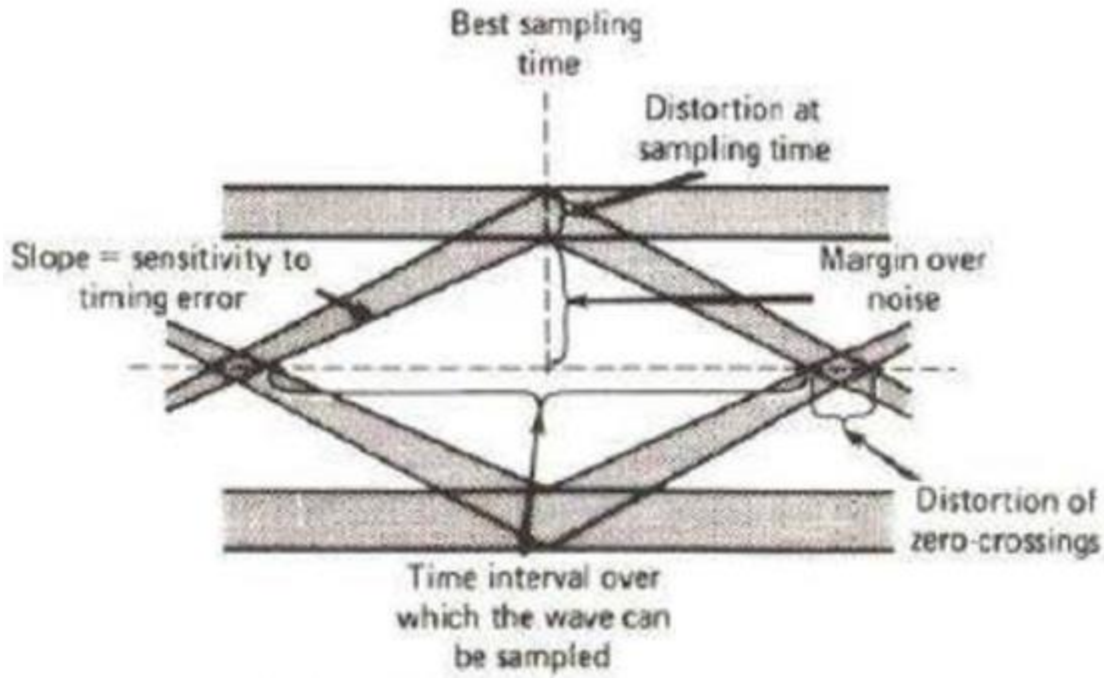
Partial response signaling

Partial response signalling(PRS), also known as correlative coding. From a practical point of view,the back ground of this technique is related to the Nyquist criterion.

BasebandM-arrayPAMtransmission:
In a baseband Mary PAM system,the pulse amplitude modulator produces M possible amplitude levels with M>2. In an M-ary system, the information source emits a sequence of symbols from an alphabet that consists o f M symbols.

Eyediagrams

The quality of digital transmission systems are evaluated using the bit error rate. Degradation ofqualityoccursineachprocessmodulation,transmission,anddetection.Theeyepatternisexperimentalmetho d that contains all the information concerning the degradation of quality. Therefore, carefulanalysisof the eye pattern is important in analyzing and degradation mechanism.
Eye patterns can be observed using an oscilloscope. The received wave is applied to the vertical deflection plates of an oscilloscope and the sawtooth wave at a rate equal to transmitted symbol rate is applied to the horizontal deflection plates, resulting display is eye pattern as it resembles humaneye.



Interpretation of eye pattern

UNIT –III
SIGNAL SPACE ANALYSIS

Introduction

space analysis provides a mathematically elegant and highly insightful tool for the study of digital signal transmission. Signal space analysis permits a general geometric framework for the interpretation of digital signaling that includes both baseband and bandpass signaling schemes.

The transmitter takes the message source output *mi* and codes it into a distinct signal *si(t)* suitable for transmission over the communications channel. The transmission channel is perturbed by zero-mean additive white Gaussian noise (AWGN).

The AWGN channel is one of the simplest mathematical models for various physical communications channels.

The received signal *r(t)* is given by

$$r(t)=s_i(t)+n(t) \quad \text{for } 0<t<T$$

The receiver has the task of observing the received signal *r(t)* for a duration of *T* seconds and making the best estimate of the transmitted signal *si(t)*. However, owing to the presence of channel noise, the decision making process is statistical in nature withthe result that the receiver will make occasional errors

The key to analyzing and understanding the performance of digital transmission is the realization that signals used in communications can be expressed and visualized graphically. Thus, we need to understand signal space concepts as applied to digital communications

We consider the following model of a generic transmission system (*digital source*):

- A message source transmits 1 symbol every T sec
- Symbols belong to an alphabet M (m_1, m_2, …m_M)
  - Binary – symbols are 0s and 1s
  - Quaternary PCM – symbols are 00, 01, 10, 11

Transmitter takes the *symbol (data)* m_i (digital message source output) and encodes it into a *distinct signal* $s_i(t)$.

The *signal* $s_i(t)$ occupies the whole *slot T* allotted to *symbol* m_i.

$s_i(t)$ is a real valued energy signal.

$$E_i = \int_0^T s_i^2(t)\,dt, \quad i=1,2,\ldots,M \qquad (5.2)$$

Linear, wide enough to accommodate the signal $s_i(t)$ with no or negligible distortion

Channel noise, is w(t) is a zero-mean white Gaussian noise process – AWGN

- additive noise
- received signal may be expressed as:

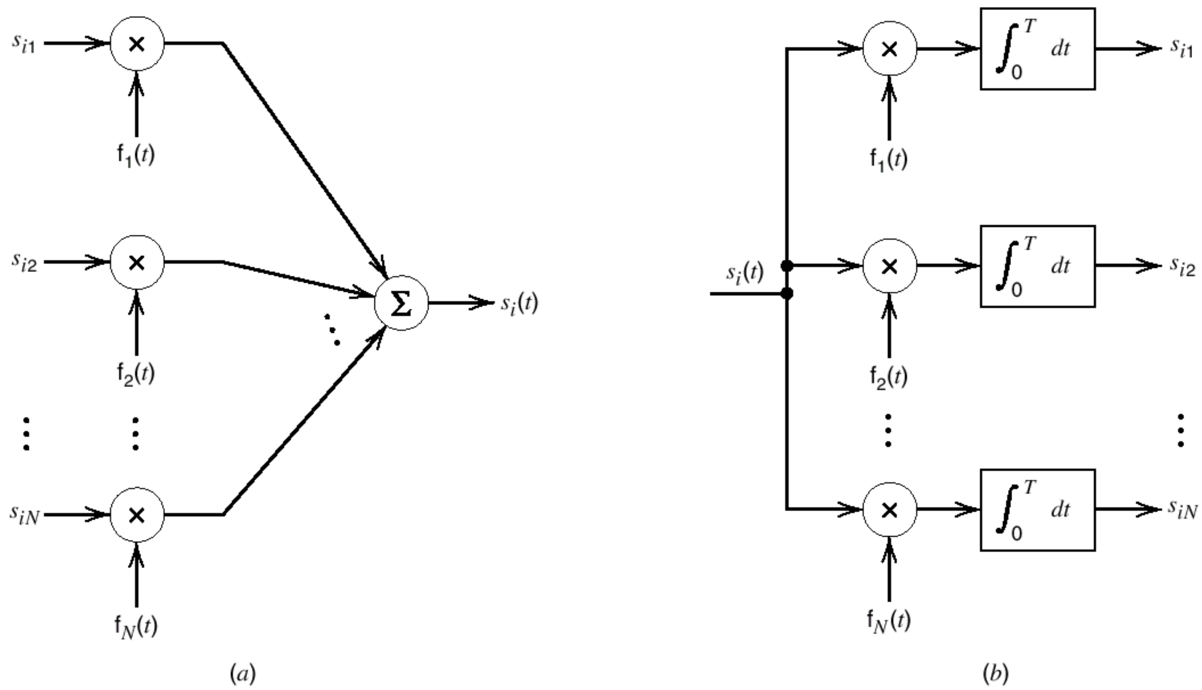
$$x(t) = s_i(t) + w(t), \qquad \begin{Bmatrix} 0 \le t \le T \\ i=1,2,....,M \end{Bmatrix} \quad (5.3)$$

GEOMETRIC REPRESENTATION OF SIGNALS:

- To represent any set of *M energy signals* $\{s_i(t)\}$ as linear combinations of *N orthogonal basis functions*, where N ≤ M
- Real value energy signals $s_1(t), s_2(t),..s_M(t)$, each of duration T sec

$$s_i(t) = \sum_{j=1}^{N} s_{ij}\phi_j(t), \qquad \begin{Bmatrix} 0 \le t \le T \\ i==1,2,....,M \end{Bmatrix} \quad (5.5)$$

- The set of coefficients can be viewed as a N-dimensional vector, denoted by s_i
- Bears a one-to-one relationship with the transmitted signal $s_i(t)$



where,

a) Synthesizer for generating the signal $s_i(t)$.

b) Analyzer for generating the set of signal vectors $\{s_i\}$.

- The signal vector s_i concept can be extended to 2D, 3D etc. N-dimensional Euclidian space
- Provides mathematical basis for the geometric representation of energy signals that is used in noise analysis
- Allows definition of
  - Length of vectors (absolute value)
  - Angles between vectors
  - Squared value (inner product of s_i with itself)

$$\|\mathbf{s_i}\|^2 = s_i^T \mathbf{s_i}$$
$$= \sum_{j=1}^{N} s_{ij}^2, \qquad i = 1,2,....,M \qquad (5.9)$$

Each signal in the set $s_i(t)$ is completely determined by the vector of its coefficients

$$\mathbf{s}_i = \begin{bmatrix} s_{i1} \\ s_{i2} \\ \cdot \\ \cdot \\ \cdot \\ s_{iN} \end{bmatrix}, \qquad i = 1,2,\ldots,M \qquad (5.8)$$

SCHWARTZ INEQUALITY

$$\left(\int_{-\infty}^{\infty} s_1(t)s_2(t)dt\right)^2 = \left(\int_{-\infty}^{\infty} s_1^2(t)dt\right)\left(\int_{-\infty}^{\infty} s_2^2(t)dt\right) \quad (5.16)$$

Gram- Schmidt orthogonalization procedure:

Suppose we are given a signal set

$\{s_1(t), s_2(t)\ldots\ldots\ldots s_M(t)\})$

Find the orthogonal basis functions for this signal set

$\{\phi_1(t), \phi_2(t)\ldots\ldots\ldots \phi_K(t)\})$

Where K<M

Step 1: Construct the First Basis Function

Compute the energy in signal 1:

$$E_1 = \int_{-\infty}^{\infty} s2(t)dt$$

$$\emptyset(t) = 1/\sqrt{E1}\, s_1(t)$$

$$S_1(t) = S_{11}\phi_1(t) = \sqrt{E_1}\,\phi_1(t)$$

$$S_{11} = \int S_1(t)\,\phi_1(t) = \sqrt{E}$$

Step 2: Construct the Second Basis Function

Compute correlation between signal 2 and basic function 1

$$S_{21}= \int_{-\infty}^{\infty} s2(t)\phi1(t)dt$$

subtract off the correlation portion

$$g_2(t)= S_2(t)-S_{21}(t)\phi_1(t)$$

compute the energy in the remaining portion

$$E_{g2}= \int_{-\infty}^{\infty} g(t)^2 dt$$

Normalize th remaining portion

$$\phi_2(t)=(1/\sqrt{E_{g2}})g_2(t)$$

$$S_{22}= \int_{-\infty}^{\infty} s2(t)\phi2(t)dt =\sqrt{E_{g2}}$$

Step 3: Construct Successive Basis Functions

For signal $S_k(t)$, compute

$$S_{ki}=\int_{-\infty}^{\infty} sk(t)\phi i(t)dt$$

Energy of the K^{th} function

$$E_{gk}= \int_{-\infty}^{\infty} gk(t)^2 dt$$

$$\phi_k (t) = (1/\sqrt{E_{gk}})g_k(t)$$

$$S_{kk}(t) = \int_{-\infty}^{\infty} sk(t)\phi k(t)dt \quad =\sqrt{E_{gk}}$$

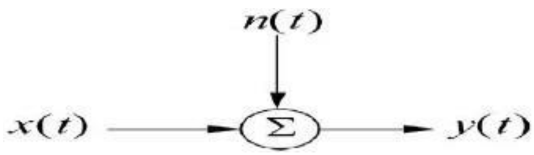
Conversion of the Continuous AWGN channel into a vector channel:

Most analyzed, digital communication channel is the AWGN channel.This channel passes the sum of the modulated signal $x(t)$ and an uncorrelated Gaussian noise $n(t)$ to the output. The Gaussian noise is assumed to be uncorrelated with itself (or "white")

- Suppose that the $s_i(t)$ is not any signal, but specifically the signal at the receiver side, defined in accordance with an AWGN channel:
- So the output of the correlator can be defined as:

$$x(t) = s_i(t) + w(t), \quad \begin{Bmatrix} 0 \le t \le T \\ i = 1, 2, \ldots, M \end{Bmatrix} \qquad (5.28)$$

$$\mathbf{x_i} = \int_0^T x(t)\phi_j(t)\,dt = s_{ij} + w_i, \quad j = 1, 2, \ldots, N \qquad (5.29)$$



The analysis can thus convert the continuous channel

$$y(t) = x(t) + n(t)$$

to a discrete vector channel model,

$$y = x + n$$

Coherent detection of signals in noise

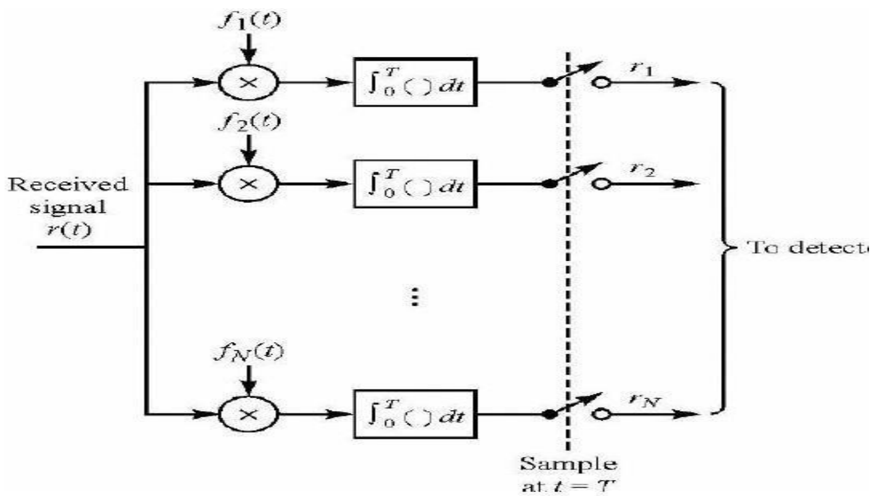
Correlation receiver

The principle on which the cross-correlation receiver operates is that, for two random time-varying signals, $V1(t)$ and $V2(t)$, the cross-correlation function

$$r(\tau)=\int_{-\infty}^{\infty} V1(t)V2(t-\tau)dt$$

$V1(t)$ and $V2(t)$ are the voltages

Correlation Demodulator



Noise components

$$r_k = \int_0^T r(t)f_k(t)dt$$

$$= \int_0^T Sm(t)f_k(t)dt + \int_0^T n(t)f_k(t)dt$$

$$= S_{mk}+n_K$$

Correlator outputs

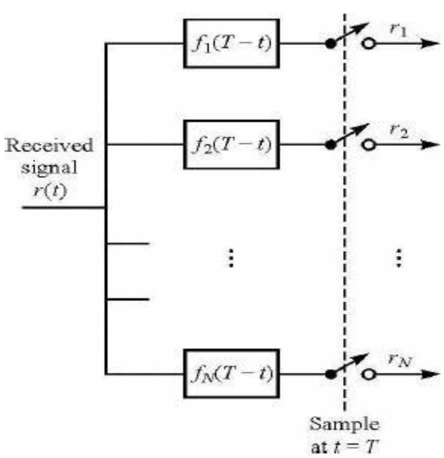
$$E(n_k n_m) = \int_0^T \int E\,[n(t)\,n(\tau)]\; f_k(t)\; f_m(\tau)\,dt\,d\tau$$

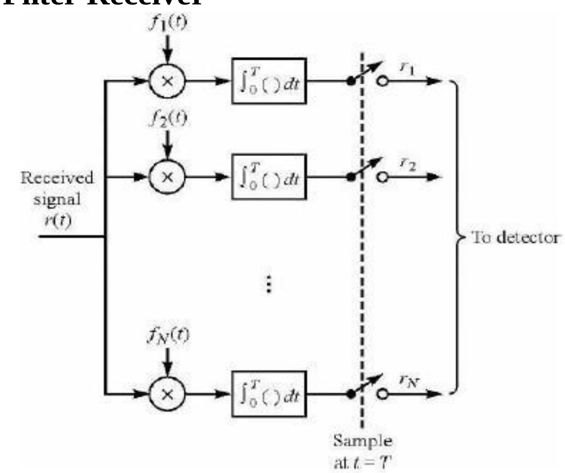
$$= 1/2 N_o \int_0^T fk(t) fm(\tau) d\tau$$

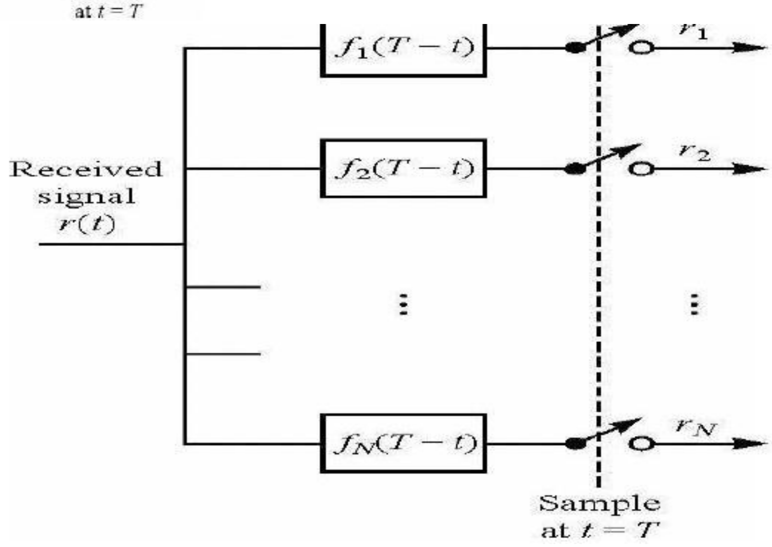
$= 1/2\ N_o \quad m = k$

$= 0 \qquad m \neq k$

Eqivalance of Correlation and Matched Filter Receiver







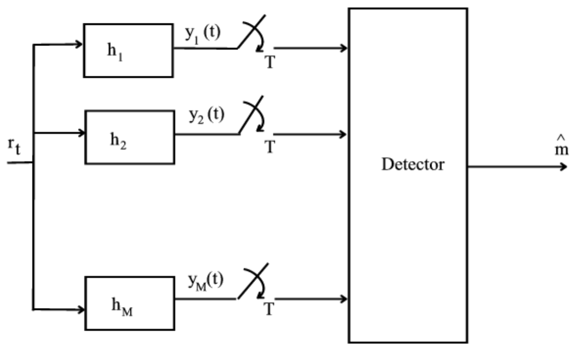
$$h_j(t) = \Phi_j(T-t)$$

$$y_j(t) = \int_{-\infty}^{\infty} x(\tau)\Phi_j(T-t+\tau)d\tau$$

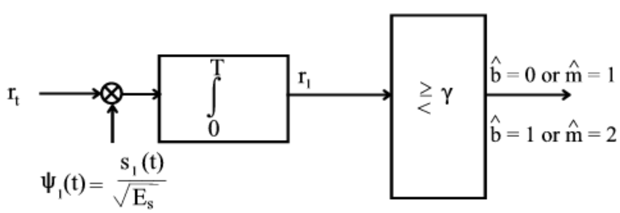
$$y_j(t) = \int_{-\infty}^{\infty} x(\tau)\Phi_j(\tau)d\tau$$

- From the definition of the matched filter, we can incorporate the impulse $h_j(t)$ and the input signal $\varphi_j(t)$ so that:
- Then, the output becomes:
- Sampling at t = T, we get:

- So we can see that the detector part of the receiver may be implemented using either matched filters or correlators. The output of each correlator is equivalent to the output of a corresponding matched filter when sampled at t = T.



Matched filters



Correlators

PROBABILITY OF ERROR:

Consider the received signal waveform for the bit transmitted between time 0 and time T . Due to the presence of noise the actual waveform $y.t/$ at the receiver is

$$y(t) = f(t) + n(t),$$

where $f(t)$ is the ideal noise-free signal.

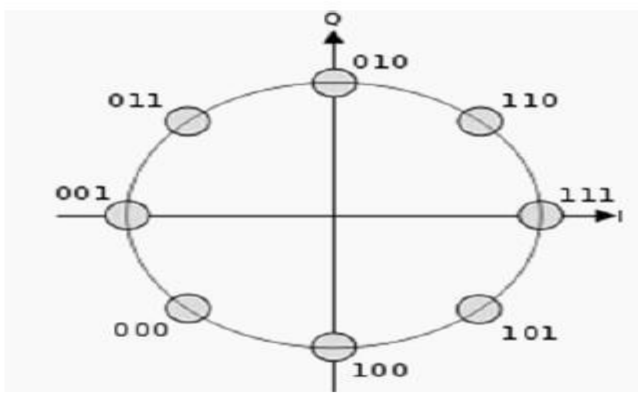
In the case described the signal f (t) is

f (t) = 0 symbol 0 transmitted (signal absent)

1 symbol 1 transmitted (signal present).

Signal constellation diagram

A constellation diagram is a representation of a signal modulated by a digital modulation scheme such as quadrature amplitude modulation or phase-shift keying. It displays the signal as a two-dimensional X-Y plane scatter diagram in the complex plane at symbol sampling instants

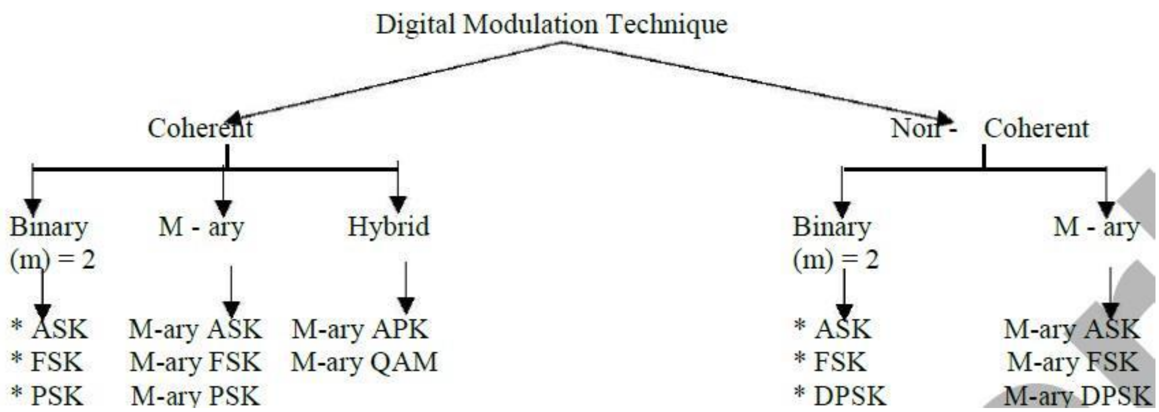


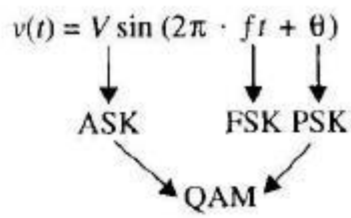
A constellation diagram for Gray encoded 8-PSK

UNIT 4
PASSBAND DATA TRANSMISSION

Introduction

Types of Digital /modulation Techniques





Referring to the above equation

If **the information signal is digital** and the **amplitude of the carrier is varied** proportional to the information signal, a digitally modulated signal called ***amplitude shift keying*** (ASK)

If the **frequency (f) is varied** proportional to the information signal, ***frequency shift keying*** (FSK) is produced and

If the **phase of the carrier (θ) is varied** proportional to the information signal, ***phase shift keying (PSK)*** is produced.

If both the **amplitude and the phase are varied** proportional to the information signal, ***quadrature amplitude modulation***(QAM) results.

ASK, FSK, PSK, and QAM are all forms of digital modulation

Pass band transmission model

The **incoming data stream is modulated onto a carrier with fixed frequency** and then **transmitted over a band-pass channel** is called pass band Transmission

There are three basic signaling schemes used in pass band data transmissiom

Amplitude-shift keying (ASK)
Frequency-shift keying (FSK)
Phase-shift keying (PSK)

AMPLITUDE-SHIFT KEYING

The simplest digital modulation technique is *amplitude-shift keying* (ASK), where a binary information signal directly modulates the amplitude of an analog carrier.

ASK is similar to standard amplitude modulation except there are **only two output amplitudes possible**. Amplitude-shift keying is sometimes called ***digital amplitude modulation*** **(DAM).**

Mathematically, amplitude-shift keying is

$$V_{ASK}(t) = \{1+V_m(t)\}[A/2 \cos(\omega_c t)]$$

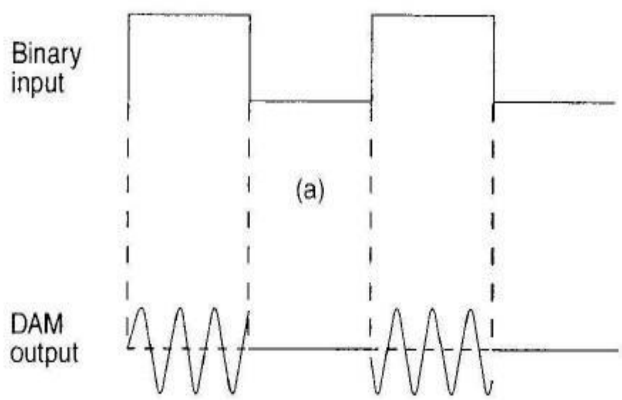
where

$V_{ASK}(t)$ = amplitude-shift keying wave

$V_m(t$ = digital information (modulating) signal (volts) A/2 =unmodulated carrier amplitude (v)

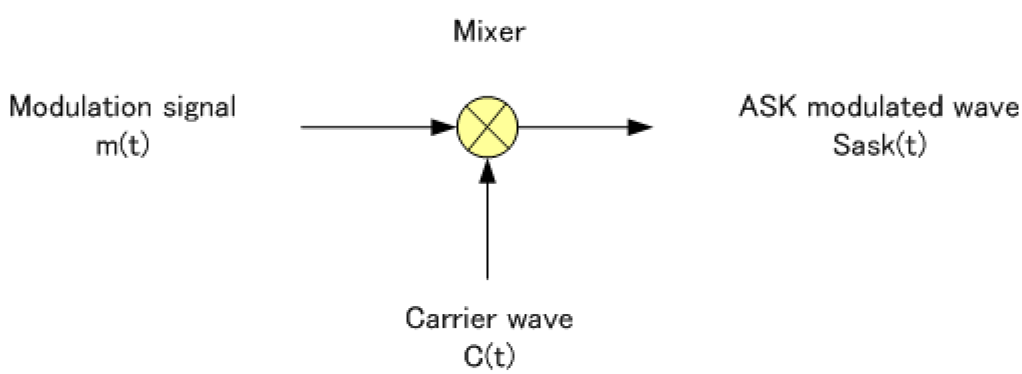
ω_c= analog carrier radian frequency (radians per second, 2πfct)

[vm(t)] is a normalized binary waveform, where + 1 V = logic 1 and -1 V = logic 0
Thus, the modulated wave $v_{ask}(t)$, is either A cos(ω_c t) or 0. Hence, the carrier is either *"on"*or *"off,"* That's why amplitude-shift keying is referred to as *on-off keying*(OOK)



The rate of change of the ASK waveform (baud) is the same as the rate of change of the binary input (bps).

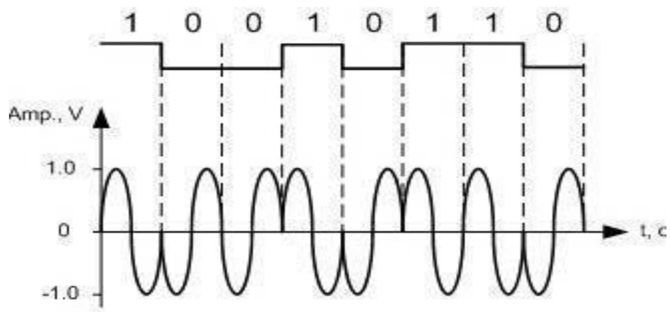
ASK TRANSMITTER:



The input binary sequence is applied to the product modulator. The product modulator amplitude modulates the sinusoidal carrier .it passes the carrier when input bit is ‘1’ .it blocks the carrier when input bit is ‘0.’

PHASE SHIFT KEYING:

The phase of the output signal gets shifted depending upon the input. These are mainly of two types, namely BPSK and QPSK, according to the number of phase shifts. The other one is DPSK



Phase shift keying (PSK)

which changes the phase according to the previous value.

Phase Shift Keying (PSK) is the digital modulation technique in which the phase of the carrier signal is changed by varying the sine and cosine inputs at a particular time. PSK technique is widely used for wireless LANs, bio-metric, contactless operations, along with RFID and Bluetooth communications.

PSK is of two types, depending upon the phases the signal gets shifted. They are −

Binary Phase Shift Keying (BPSK)

This is also called as **2-phase PSK** (or) **Phase Reversal Keying**. In this technique, the sine wave carrier takes two phase reversals such as 0° and 180°.

BPSK is basically a DSB-SC (Double Sideband Suppressed Carrier) modulation scheme, for message being the digital information.

Following is the image of BPSK Modulated output wave along with its in

Binary Phase-Shift Keying

The simplest form of PSK is *binary phase-shift keying* (BPSK), where N = 1 and *M = 2.* Therefore, with BPSK, two phases ($2^1 = 2$) are possible for the carrier. One phase represents a logic 1, and the other phase represents a logic 0. As the input digital signal changes state (i.e., from a 1 to a 0 or from a 0 to a 1), the phase of the output carrier shifts between two angles that are separated by 180°.

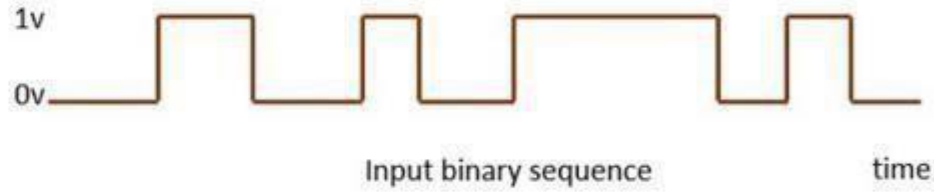
Hence, other names for BPSK are *phase reversal keying* (PRK) and *biphase modulation.* BPSK is a form of square-wave modulation of a *continuous wave (CW)* signal.

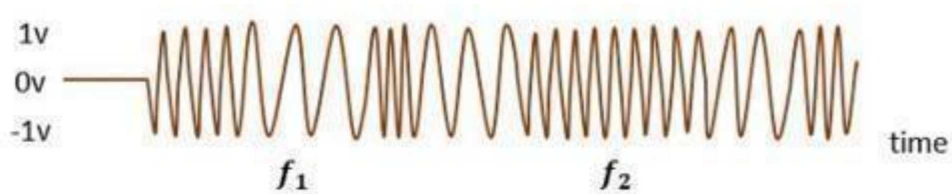
FREQUENCY SHIFT KEYING

The frequency of the output signal will be either high or low, depending upon the input data applied.

Frequency Shift Keying (FSK) is the digital modulation technique in which the frequency of the carrier signal varies according to the discrete digital changes. FSK is a scheme of frequency modulation.

Following is the diagram for FSK modulated waveform along with its input.





FSK Modulated output wave

The output of a FSK modulated wave is high in frequency for a binary HIGH input and is low in frequency for a binary LOW input. The binary 1s and 0s are called **Mark** and **Space frequencies**.

FSK is a form of constant-amplitude angle modulation similar to standard frequency modulation (FM) except the modulating signal is a binary signal that varies between two discrete voltage levels rather than a continuously changing analog waveform.Consequently, FSK is sometimes called *binary FSK* (BFSK). The general expression for FSK is

$$v_{fsk}(t) = V_c \cos\{2\pi[f_c + v_m(t)\,\Delta f]t\}$$

where

$v_{fsk}(t)$ = binary FSK waveform

V_c = peak analog carrier amplitude (volts)

f_c = analog carrier center frequency(hertz)

f=peak change (shift)in the analog carrier frequency(hertz)

$v_m(t)$ = binary input (modulating) signal (volts)

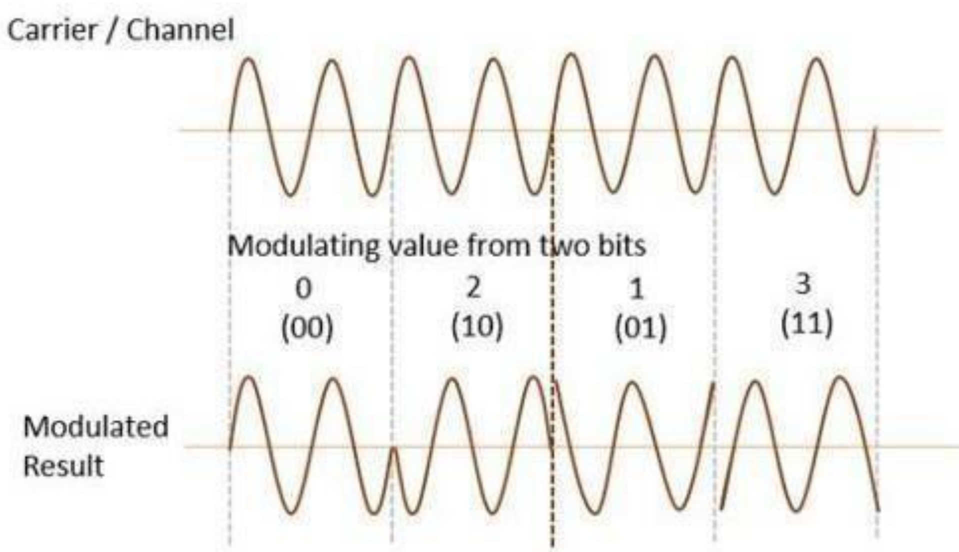
The modulating signal is a normalized binary waveform where a logic 1 = + 1 V and a logic 0 = -1 V. Thus, for a logic l input, $v_m(t)$ a logic 0 input, $v_m(t)$ = -1, Equation becomes

With binary FSK, the carrier center frequency (f_c) is shifted (deviated) up and down in the frequency domain by the binary input signal

QUADRATURE PHASE SHIFT KEYING (QPSK):

This is the phase shift keying technique, in which the sine wave carrier takes four phase reversals such as 0°, 90°, 180°, and 270°.

If this kind of techniques are further extended, PSK can be done by eight or sixteen values also, depending upon the requirement. The following figure represents the QPSK waveform for two bits input, which shows the modulated result for different instances of binary inputs.



QPSK is a variation of BPSK, and it is also a DSB-SC (Double Sideband Suppressed Carrier) modulation scheme, which sends two bits of digital information at a time, called as **bigits**.

Instead of the conversion of digital bits into a series of digital stream, it converts them into bit-pairs. This decreases the data bit rate to half, which allows space for the other users.

QPSK transmitter.

A block diagram of a QPSK modulator is shown in Figure 2-17Two bits (a dibit) are clocked into the bit splitter. After both bits have been serially inputted, they are simultaneously parallel outputted.

The I bit modulates a carrier that is in phase with the reference oscillator (hence the name

"I" for "in phase" channel), and theQ bit modulate, a carrier that is 90° out of phase.

For a logic 1 = + 1 V and a logic 0= - 1 V, two phases are possible at the output of the I balanced modulator (+sin $\omega_c t$ and - sin $\omega_c t$), and two phases are possible at the output of the Q balanced modulator (+cos $\omega_c t$), and (-cos $\omega_c t$).

When the linear summer combines the two quadrature (90° out of phase) signals, there are four possible resultant phasors given by these expressions: + sin $\omega_c t$ + cos $\omega_c t$, + sin $\omega_c t$ - cos $\omega_c t$, -sin $\omega_c t$ + cos $\omega_c t$, and -sin $\omega_c t$ - cos $\omega_c t$.

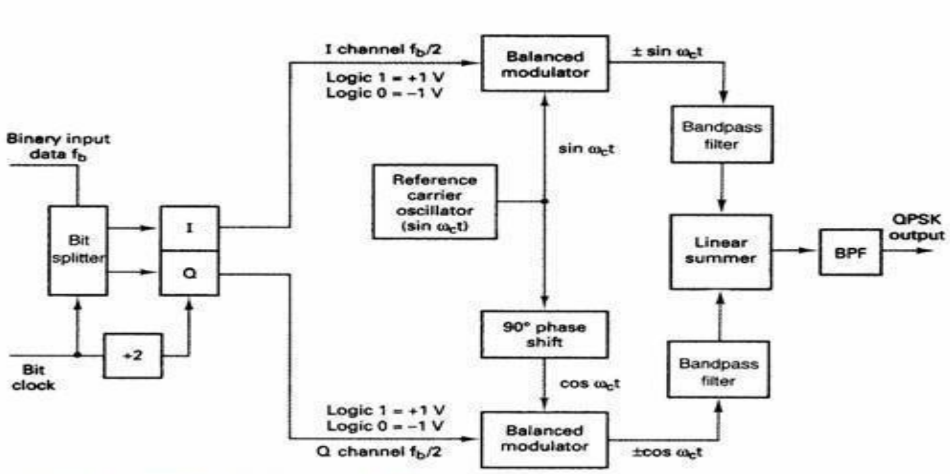


FIGURE 2-17 QPSK modulator

QPSK RECEIVER:

The block diagram of a QPSK receiver is shown in Figure 2-21

The power splitter directs the input QPSK signal to the I and Q product detectors and the carrier recovery circuit. The carrier recovery circuit reproduces the original transmit carrier oscillator signal. The recovered carrier must be frequency and phase coherent with the transmit reference carrier. The QPSK signal is demodulated in the I and Q product detectors, which generate the original I and Q data bits. The outputs of the product detectors are fed to the bit combining circuit, where they are converted from parallel I and Q data channels to a single binary output data stream. The incoming QPSK signal may be any one of the four possible output phases shown in Figure 2-
18. To illustrate the demodulation process, let the incoming QPSK signal be -sin $\omega_c t$ + cos $\omega_c t$. Mathematically, the demodulation process is as follows.

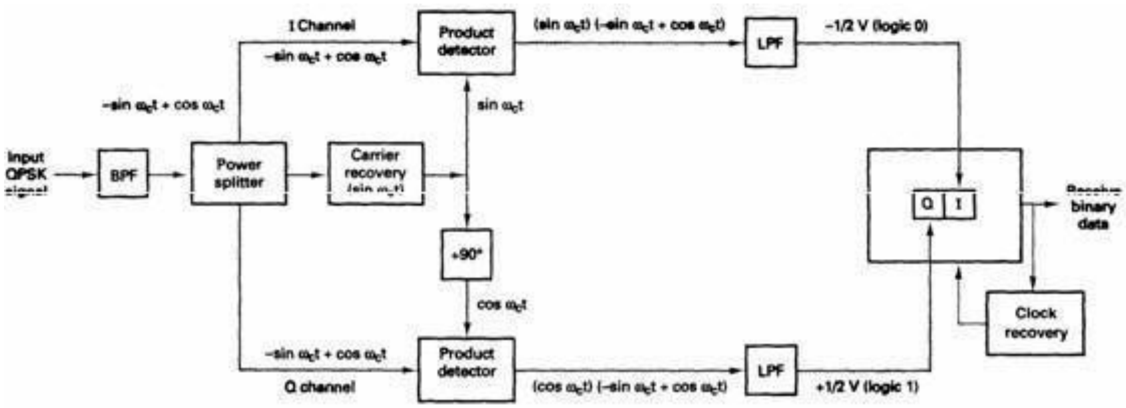
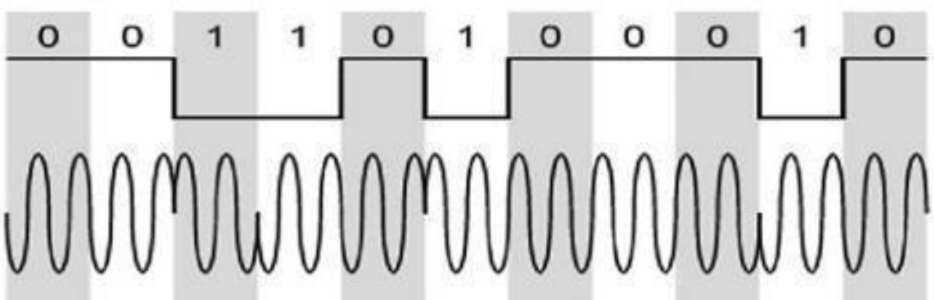


FIGURE 2-21 QPSK receiver

DIFFERENTIAL PHASE SHIFT KEYING (DPSK):

In DPSK (Differential Phase Shift Keying) the phase of the modulated signal is shifted relative to the previous signal element. No reference signal is considered here. The signal phase follows the high or low state of the previous element. This DPSK technique doesn't need a reference oscillator.

The following figure represents the model waveform of DPSK.



It is seen from the above figure that, if the data bit is LOW i.e., 0, then the phase of the signal is not reversed, but is continued as it was. If the data is HIGH i.e., 1, then the phase of the signal is reversed, as with NRZI, invert on 1 (a form of differential encoding).

If we observe the above waveform, we can say that the HIGH state represents an **M** in the modulating signal and the LOW state represents a **W** in the modulating signal.

The word binary represents two-bits. **M** simply represents a digit that corresponds to the number of conditions, levels, or combinations possible for a given number of binary variables.

This is the type of digital modulation technique used for data transmission in which instead of one-bit, two or **more bits are transmitted at a time**. As a single signal is used for multiple bit transmission, the channel bandwidth is reduced.

DBPSK TRANSMITTER.:

Figure 2-37a shows a simplified block diagram of a *differential binary phase-shift keying* (DBPSK) transmitter. An incoming information bit is XNORed with the preceding bit prior to entering the BPSK modulator (balanced modulator).

For the first data bit, there is no preceding bit with which to compare it. Therefore, an initial reference bit is assumed. Figure 2-37b shows the relationship between the input data, the XNOR output data, and the phase at the output of the balanced modulator. If the initial reference bit is assumed a logic 1, the output from the XNOR circuit is simply the complement of that shown.

In Figure 2-37b, the first data bit is XNORed with the reference bit. If they are the same, the XNOR output is a logic 1; if they are different, the XNOR output is a logic 0. The balanced modulator operates the same as a conventional BPSK modulator; a logic I produces +sin $\omega_c t$ at the output, and A logic 0 produces –sin $\omega_c t$ at the output.

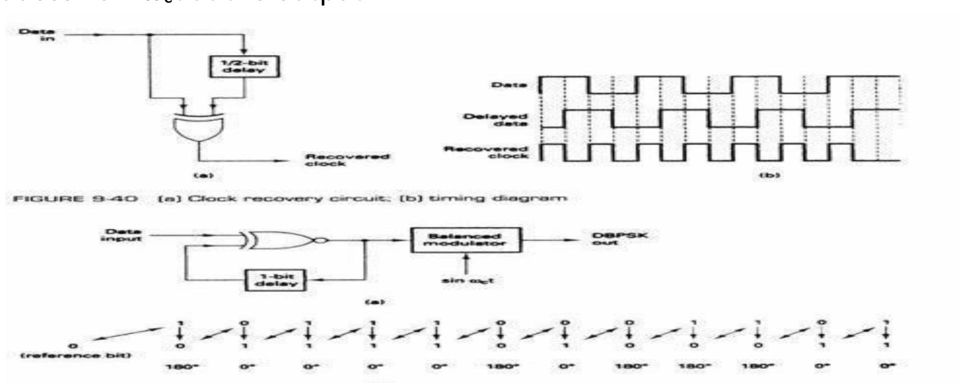


FIGURE 2-37 DBPSK modulator (a) block diagram (b) timing diagram

BPSK RECEIVER:

Figure 9-38 shows the block diagram and timing sequence for a DBPSK receiver. The received signal is delayed by one bit time, then compared with the next signaling element in the balanced modulator. If they are the same. J logic 1(+ voltage) is generated. If they are different, a logic 0 (- voltage) is generated. [f the reference phase is incorrectly assumed, only the first demodulated bit is in error. Differential encoding can be implemented with higher-than-binary digital modulation schemes, although the differential algorithms are much more complicated than for DBPS K.

The primary advantage of DBPSK is the simplicity with which it can be implemented. With DBPSK, no carrier recovery circuit is needed. A disadvantage of DBPSK is, that it requires between 1 dB and 3 dB more signal-to-noise ratio to achieve the same bit error rate as that of absolute PSK

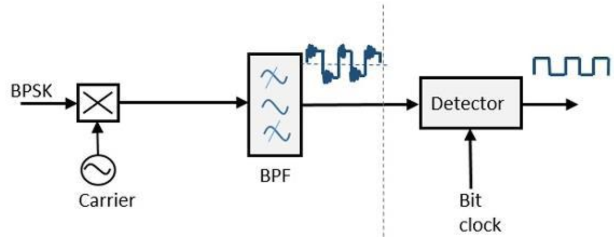


FIGURE 2-BPSK demodulator: (a) block diagram

COHERENT RECEPTION OF FSK:

The coherent demodulator for the coherent FSK signal falls in the general form of coherent demodulators described in Appendix B. The demodulator can be implemented with two correlators as shown in Figure 3.5, where the two reference signals are cos(27r f t) and cos(27r fit). They must be synchronized with the received signal. The receiver is optimum in the sense that it minimizes the error probability for equally likely binary signals. Even though the receiver is rigorously derived in Appendix B, some heuristic explanation here may help understand its operation. When s 1 (t) is transmitted, the upper correlator yields a signal 1 with a positive signal component and a noise component. However, the lower correlator output 12, due to the signals' orthogonality, has only a noise component. Thus the output of the summer is most likely above zero, and the threshold detector will most likely produce a 1. When s2(t) is transmitted, opposite things happen to the two correlators and the threshold detector will most likely produce a 0. However, due to the noise nature that its values range from -00 to m, occasionally the noise amplitude might overpower the signal amplitude, and then detection errors will happen. An alternative to Figure 3.5 is to use just one correlator with the reference signal cos (27r f t) - cos(2s f2t) (Figure 3.6). The correlator in Figure

can be replaced by a matched filter that matches cos(27r fit) - cos(27r f2t) (Figure 3.7). All

implementations are equivalent in terms of error performance (see Appendix B). Assuming an AWGN channel, the received signal is

$$r(t) = s_i(t) + n(t), \qquad i = 1.2$$

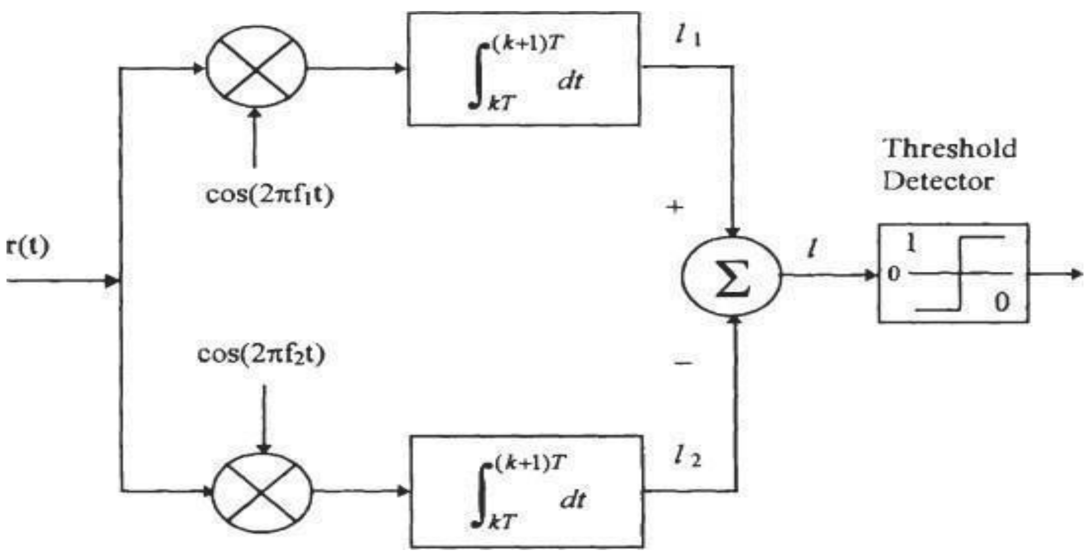
where n(t) is the additive white Gaussian noise with zero mean and a two-sided power spectral density A',/2. From (B.33) the bit error probability for any equally likely binary signals is

$$P_b = Q\left(\sqrt{\frac{E_1 + E_2 - 2\rho_{12}\sqrt{E_1 E_2}}{2N_o}}\right)$$

where No/2 is the two-sided power spectral density of the additive white Gaussian noise. For Sunde's FSK signals El = Ez = Eb, pl2 = 0 (orthogonal). thus the error probability is

$$P_b = Q\left(\sqrt{\frac{E_b}{N_o}}\right)$$

where Eb = A2T/2 is the average bit energy of the FSK signal. The above Pb is plotted in Figure 3.8 where Pb of noncoherently demodulated FSK, whose expression will be given shortly, is also



plotted for comparison.

5.13.1 Error Probability of ASK

In Amplitude Shift Keying (ASK), some number of carrier cycles are transmitted to send '1' and no signal is transmitted for binary '0'. Thus,

Binary '1' $\Rightarrow$ $x_1(t) = \sqrt{2P_s}\cos(2\pi f_0 t)$ and

Binary '0' $\Rightarrow$ $x_2(t) = 0$ (i.e. no signal) ... (5.13.1)

Here P_s is the normalized power of the signal in 1Ω load. i.e. power $P_s = \frac{A^2}{2}$. Hence $A = \sqrt{2P_s}$. Therefore in above equation for $x_1(t)$ amplitude 'A' is replaced by $\sqrt{2P_s}$.

We know that the probability of error of the optimum filter is given as,

$$P_e = \frac{1}{2}\, erfc\left\{\frac{x_{01}(T) - x_{02}(T)}{2\sqrt{2}\,\sigma}\right\} \quad \text{... (5.13.2)}$$

Here $\left[\frac{x_{01}(T) - x_{02}(T)}{\sigma}\right]^2_{max} = \int_{-\infty}^{\infty} \frac{|X(f)|^2}{S_{ni}(f)}\, df$

The above equations can be applied to matched filter when we consider white Gaussian noise. The power spectral density of white Gaussian noise is given as,

$$S_{ni}(f) = \frac{N_0}{2}$$

Putting this value of $S_{ni}(f)$ in above equations we get,

$$\left[\frac{x_{01}(T) - x_{02}(T)}{\sigma}\right]^2_{max} = \int_{-\infty}^{\infty} \frac{|X(f)|^2}{\frac{N_0}{2}}\, df$$

$$= \frac{2}{N_0}\int_{-\infty}^{\infty} |X(f)|^2\, df \quad \text{... (5.13.3)}$$

Parseval's power theorem states that,

$$\int_{-\infty}^{\infty} |X(f)|^2\, df = \int_{-\infty}^{\infty} x^2(t)\, dt$$

Hence equation 5.13.3 becomes,

$$\left[\frac{x_{01}(T) - x_{02}(T)}{\sigma}\right]^2_{max} = \frac{2}{N_0}\int_{-\infty}^{\infty} x^2(t)\, dt$$

We know that $x(t)$ is present from 0 to T. Hence limits in above equation can be changed as follows :

$$\left[\frac{x_{01}(T) - x_{02}(T)}{\sigma}\right]^2_{max} = \frac{2}{N_0}\int_0^T x^2(t)\,dt \qquad \text{... (5.13.4)}$$

We know that $x(t) = x_1(t) - x_2(t)$. For ASK $x_2(t)$ is zero, hence $x(t) = x_1(t)$. Hence above equation becomes,

$$\left[\frac{x_{01}(T) - x_{02}(T)}{\sigma}\right]^2_{max} = \frac{2}{N_0}\int_0^T x_1^2(t)\,dt$$

Putting equation of $x_1(t)$ from equation 5.13.1 in above equation we get,

$$\left[\frac{x_{01}(T) - x_{02}(T)}{\sigma}\right]^2_{max} = \frac{2}{N_0}\int_0^T \left[\sqrt{2P_s}\cos(2\pi f_0 t)\right]^2 dt$$

$$= \frac{4P_s}{N_0}\int_0^T \cos^2(2\pi f_0 t)\,dt$$

We know that $\cos^2\theta = \frac{1+\cos 2\theta}{2}$. Here applying this formula to $\cos^2(2\pi f_0 t)$ we get,

$$\left[\frac{x_{01}(T) - x_{02}(T)}{\sigma}\right]^2_{max} = \frac{4P_s}{N_0}\int_0^T \frac{1+\cos 4\pi f_0 t}{2}\,dt$$

$$= \frac{4P_s}{N_0}\cdot\frac{1}{2}\left\{\int_0^T dt + \int_0^T \cos 4\pi f_0 t\,dt\right\}$$

$$= \frac{2P_s}{N_0}\left\{[t]_0^T + \left[\frac{\sin 4\pi f_0 t}{4\pi f_0}\right]_0^T\right\}$$

$$= \frac{2P_s}{N_0}\left\{T + \frac{\sin 4\pi f_0 T}{4\pi f_0}\right\} \qquad \text{... (5.13.5)}$$

We know that T is the bit period and in this one bit period, the carrier has integer number of cycles. Thus the product $f_0 T$ is an integer. This is illustrated in Fig. 5.13.1

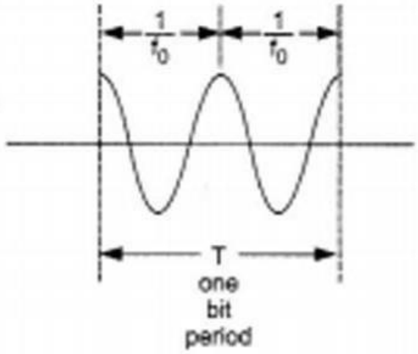


Fig. 5.13.1 In one bit period T, the carrier completes its two cycles. The carrier has frequency f_0. From figure we can write,

$$T = \frac{1}{f_0} + \frac{1}{f_0}$$

i.e. $$T = \frac{2}{f_0}$$

$\therefore \quad f_0 T = 2 \quad$ (integer no. of cycles)

As shown in above figure, the carrier completes two cycles in one bit duration. Hence

$$f_0 T = 2$$

Therefore, in general if carrier completes 'k' number of cycles, then,

$$f_0 T = k \qquad \text{(Here } k \text{ is an integer)}$$

Therefore the sine term in equation 5.13.5 becomes, $\sin 4\pi k$ and k is integer.

For all integer values of k, $\sin 4\pi k = 0$. Hence equation 5.13.5 becomes,

$$\left[\frac{x_{01}(T) - x_{02}(T)}{\sigma}\right]^2_{max} = \frac{2P_s T}{N_0} \qquad \text{... (5.13.6)}$$

$$\therefore \quad \left[\frac{x_{01}(T) - x_{02}(T)}{\sigma}\right]_{max} = \sqrt{\frac{2P_s T}{N_0}} \qquad \text{... (5.13.7)}$$

Putting this value in equation 5.13.2 we get error probability of ASK using matched filter detection as,

$$P_e = \frac{1}{2}\, erfc\left\{\frac{1}{2\sqrt{2}} \cdot \sqrt{\frac{2P_s T}{N_0}}\right\} = \frac{1}{2}\, erfc\sqrt{\frac{P_s T}{4N_0}}$$

Here $P_s T = E$, i.e. energy of one bit hence above equation becomes,

$$\textbf{Error probability of ASK : } P_e = \frac{1}{2}\, erfc\sqrt{\frac{E}{4N_0}} \qquad \text{... (5.13.8)}$$

This is the expression for error probability of ASK using matched filter detection.

Error Probability of Binary FSK

The observation vector $\mathbf{x}$ has two elements x_1 and x_2 that are defined by, respectively,

$$x_1 = \int_0^{T_b} x(t)\phi_1(t)\, dt \tag{6.92}$$

$$x_2 = \int_0^{T_b} x(t)\phi_2(t)\, dt \tag{6.93}$$

where $x(t)$ is the received signal, the form of which depends on which symbol was transmitted. Given that symbol 1 was transmitted, $x(t)$ equals $s_1(t) + w(t)$, where $w(t)$ is the sample function of a white Gaussian noise process of zero mean and power spectral density $N_0/2$. If, on the other hand, symbol 0 was transmitted, $x(t)$ equals $s_2(t) + w(t)$.

Now, applying the decision rule of Equation (5.59), we find that the observation space is partitioned into two decision regions, labeled Z_1 and Z_2 in Figure 6.25. The decision boundary, separating region Z_1 from region Z_2 is the perpendicular bisector of the line joining the two message points. The receiver decides in favor of symbol 1 if the received signal point represented by the observation vector $\mathbf{x}$ falls inside region Z_1. This occurs when $x_1 > x_2$. If, on the other hand, we have $x_1 < x_2$, the received signal point falls inside region Z_2, and the receiver decides in favor of symbol 0. On the decision boundary, we have $x_1 = x_2$, in which case the receiver makes a random guess in favor of symbol 1 or 0.

Define a new Gaussian random variable Y whose sample value y is equal to the difference between x_1 and x_2; that is,

$$y = x_1 - x_2 \tag{6.94}$$

The mean value of the random variable Y depends on which binary symbol was transmitted. Given that symbol 1 was transmitted, the Gaussian random variables X_1 and X_2, whose sample values are denoted by x_1 and x_2, have mean values equal to $\sqrt{E_b}$ and zero, respectively. Correspondingly, the conditional mean of the random variable Y, given that symbol 1 was transmitted, is

$$\begin{aligned} E[Y|1] &= E[X_1|1] - E[X_2|1] \\ &= +\sqrt{E_b} \end{aligned} \tag{6.95}$$

On the other hand, given that symbol 0 was transmitted, the random variables X_1 and X_2 have mean values equal to zero and $\sqrt{E_b}$, respectively. Correspondingly, the conditional mean of the random variable Y, given that symbol 0 was transmitted, is

$$E[Y|0] = E[X_1|0] - E[X_2|0] = -\sqrt{E_b} \tag{6.96}$$

The variance of the random variable Y is independent of which binary symbol was transmitted. Since the random variables X_1 and X_2 are statistically independent, each with a variance equal to $N_0/2$, it follows that

$$\text{var}[Y] = \text{var}[X_1] + \text{var}[X_2] = N_0 \tag{6.97}$$

Suppose we know that symbol 0 was transmitted. The conditional probability density function of the random variable Y is then given by

$$f_Y(y|0) = \frac{1}{\sqrt{2\pi N_0}} \exp\left[-\frac{(y + \sqrt{E_b})^2}{2N_0}\right] \tag{6.98}$$

Since the condition $x_1 > x_2$, or equivalently, $y > 0$, corresponds to the receiver making a decision in favor of symbol 1, we deduce that the conditional probability of error, given that symbol 0 was transmitted, is

$$\begin{aligned} p_{10} &= P(y > 0|\text{symbol 0 was sent}) \\ &= \int_0^\infty f_Y(y|0)\, dy \\ &= \frac{1}{\sqrt{2\pi N_0}} \int_0^\infty \exp\left[-\frac{(y + \sqrt{E_b})^2}{2N_0}\right] dy \end{aligned} \tag{6.99}$$

$$\frac{y + \sqrt{E_b}}{\sqrt{2N_0}} = z \tag{6.100}$$

Then, changing the variable of integration from y to z, we may rewrite Equation (6.99) as follows:

$$\begin{aligned} p_{10} &= \frac{1}{\sqrt{\pi}} \int_{\sqrt{E_b/2N_0}}^\infty \exp(-z^2)\, dz \\ &= \frac{1}{2} \text{erfc}\left(\sqrt{\frac{E_b}{2N_0}}\right) \end{aligned} \tag{6.101}$$

Similarly, we may show the p_{01}, the conditional probability of error given that symbol 1 was transmitted, has the same value as in Equation (6.101). Accordingly, averaging p_{10} and p_{01}, we find that the *average probability of bit error* or, equivalently, the *bit error rate for coherent binary FSK* is (assuming equiprobable symbols)

$$P_e = \frac{1}{2}\,\text{erfc}\left(\sqrt{\frac{E_b}{2N_0}}\right) \tag{6.102}$$

Comparing Equations (6.20) and (6.102), we see that, in a coherent binary FSK system, we have to double the *bit energy-to-noise density ratio*, E_b/N_0, to maintain the same bit error rate as in a coherent binary PSK system. This result is in perfect accord with the signal-space diagrams of Figures 6.3 and 6.25, where we see that in a binary PSK system the Euclidean distance between the two message points is equal to $2\sqrt{E_b}$, whereas in a binary FSK system the corresponding distance is $\sqrt{2E_b}$. For a prescribed E_b, the minimum distance $d_{\min}$ in binary PSK is therefore $\sqrt{2}$ times that in binary FSK. Recall from Chapter 5 that the probability of error decreases exponentially as $d_{\min}^2$, hence the difference between the formulas of Equations (6.20) and (6.102).

Error Probability of QPSK

In a coherent QPSK system, the received signal $x(t)$ is defined by

$$x(t) = s_i(t) + w(t), \qquad \begin{cases} 0 \le t \le T \\ i = 1, 2, 3, 4 \end{cases} \tag{6.28}$$

where $w(t)$ is the sample function of a white Gaussian noise process of zero mean and power spectral density $N_0/2$. Correspondingly, the observation vector $\mathbf{x}$ has two elements, x_1 and x_2, defined by

$$\begin{aligned} x_1 &= \int_0^T x(t)\phi_1(t)\,dt \\ &= \sqrt{E}\cos\left[(2i-1)\frac{\pi}{4}\right] + w_1 \\ &= \pm\sqrt{\frac{E}{2}} + w_1 \end{aligned} \tag{6.29}$$

$$
\begin{aligned}
x_2 &= \int_0^T x(t)\phi_2(t)\,dt \\
&= -\sqrt{E}\sin\left[(2i-1)\frac{\pi}{4}\right] + w_2 \\
&= \mp\sqrt{\frac{E}{2}} + w_2
\end{aligned} \tag{6.30}
$$

Thus the observable elements x_1 and x_2 are sample values of independent Gaussian random variables with mean values equal to $\pm\sqrt{E/2}$ and $\mp\sqrt{E/2}$, respectively, and with a common variance equal to $N_0/2$.

The decision rule is now simply to decide that $s_1(t)$ was transmitted if the received signal point associated with the observation vector **x** falls inside region Z_1, decide that $s_2(t)$ was transmitted if the received signal point falls inside region Z_2, and so on. An erroneous decision will be made if, for example, signal $s_4(t)$ is transmitted but the noise $w(t)$ is such that the received signal point falls outside region Z_4.

To calculate the average probability of symbol error, we note from Equation (6.24) that a coherent QPSK system is in fact equivalent to two coherent binary PSK systems working in parallel and using two carriers that are in phase quadrature; this is merely a statement of the quadrature-carrier multiplexing property of coherent QPSK. The in-phase channel output x_1 and the quadrature channel output x_2 (i.e., the two elements of the observation vector **x**) may be viewed as the individual outputs of the two coherent binary PSK systems. Thus, according to Equations (6.29) and (6.30), these two binary PSK systems may be characterized as follows:

- The signal energy per bit is $E/2$.
- The noise spectral density is $N_0/2$.

Hence, using Equation (6.20) for the average probability of bit error of a coherent binary PSK system, we may now state that the average probability of bit error in *each* channel of the coherent QPSK system is

$$
\begin{aligned}
P' &= \frac{1}{2}\operatorname{erfc}\left(\sqrt{\frac{E/2}{N_0}}\right) \\
&= \frac{1}{2}\operatorname{erfc}\left(\sqrt{\frac{E}{2N_0}}\right)
\end{aligned} \tag{6.31}
$$

$$
\begin{aligned}
P_c &= (1 - P')^2 \\
&= \left[1 - \frac{1}{2}\,\mathrm{erfc}\left(\sqrt{\frac{E}{2N_0}}\right)\right]^2 \qquad (6.32)\\
&= 1 - \mathrm{erfc}\left(\sqrt{\frac{E}{2N_0}}\right) + \frac{1}{4}\,\mathrm{erfc}^2\left(\sqrt{\frac{E}{2N_0}}\right)
\end{aligned}
$$

The average probability of symbol error for coherent QPSK is therefore

$$
\begin{aligned}
P_e &= 1 - P_c \\
&= \mathrm{erfc}\left(\sqrt{\frac{E}{2N_0}}\right) - \frac{1}{4}\,\mathrm{erfc}^2\left(\sqrt{\frac{E}{2N_0}}\right) \qquad (6.33)
\end{aligned}
$$

In the region where $(E/2N_0) \gg 1$, we may ignore the quadratic term on the right-hand side of Equation (6.33), so we approximate the formula for the average probability of symbol error for coherent QPSK as

$$
P_e \simeq \mathrm{erfc}\left(\sqrt{\frac{E}{2N_0}}\right) \qquad (6.34)
$$

The formula of Equation (6.34) may also be derived in another insightful way, using the signal-space diagram of Figure 6.6. Since the four message points of this diagram are circularly symmetric with respect to the origin, we may apply Equation (5.92), reproduced here in the form

$$
P_e \le \frac{1}{2} \sum_{\substack{k=1 \\ k \ne i}}^{4} \mathrm{erfc}\left(\frac{d_{ik}}{2\sqrt{N_0}}\right) \qquad \text{for all } i \qquad (6.35)
$$

Consider, for example, message point m_1 (corresponding to dibit 10) chosen as the transmitted message point. The message points m_2 and m_4 (corresponding to dibits 00 and 11) are the *closest* to m_1. From Figure 6.6 we readily find that m_1 is equidistant from m_2 and m_4 in a Euclidean sense, as shown by

$$
d_{12} = d_{14} = \sqrt{2E}
$$

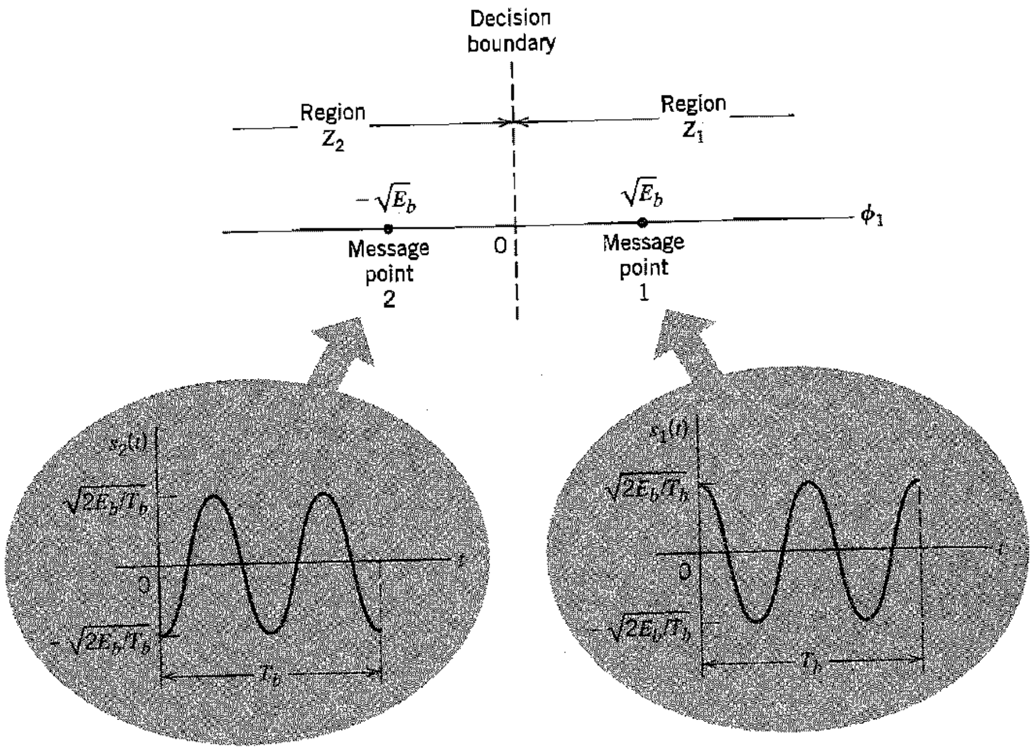


FIGURE 6.3 Signal-space diagram for coherent binary PSK system. The waveforms depicting the transmitted signals $s_1(t)$ and $s_2(t)$, displayed in the inserts, assume $n_c = 2$.

Another important point to note is that the bit errors in the in-phase and quadrature channels of the coherent QPSK system are statistically independent. The in-phase channel makes a decision on one of the two bits constituting a symbol (dibit) of the QPSK signal, and the quadrature channel takes care of the other bit. Accordingly, the *average probability of a correct decision* resulting from the combined action of the two channels working together is

Signal Space of BPSK:

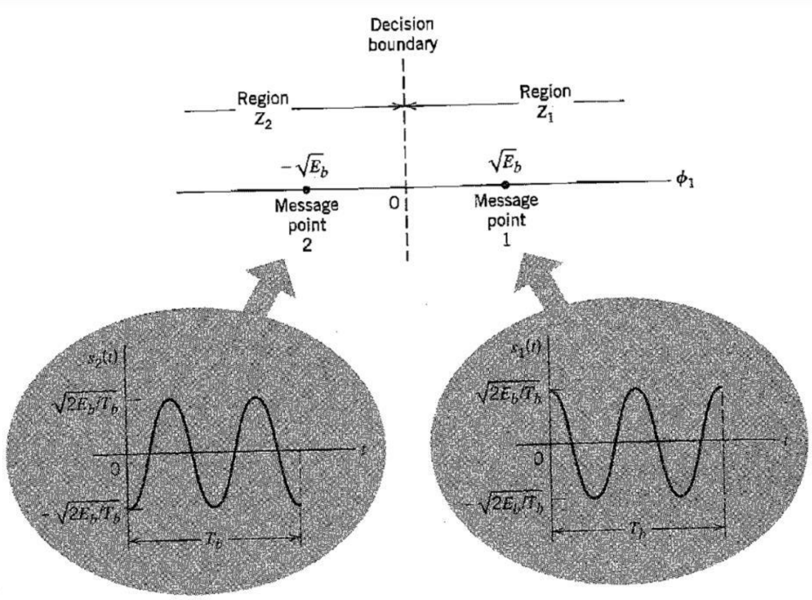


FIGURE 6.3 Signal-space diagram for coherent binary PSK system. The waveforms depicting the transmitted signals $s_1(t)$ and $s_2(t)$, displayed in the inserts, assume $n_c = 2$.

Power spectrum of BPSK:

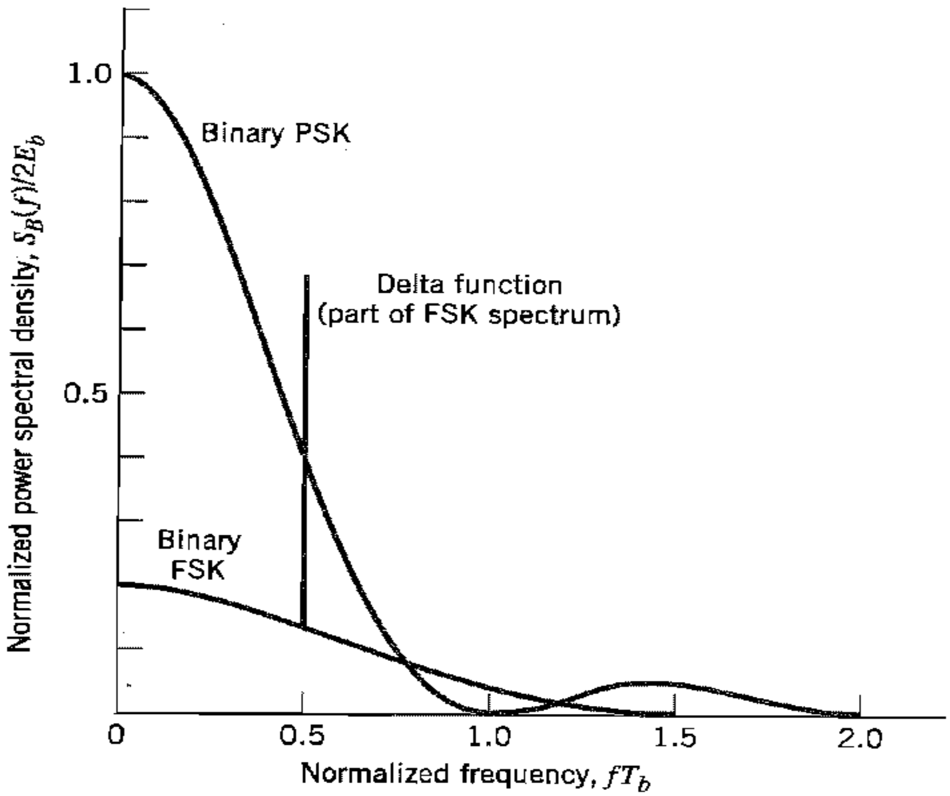
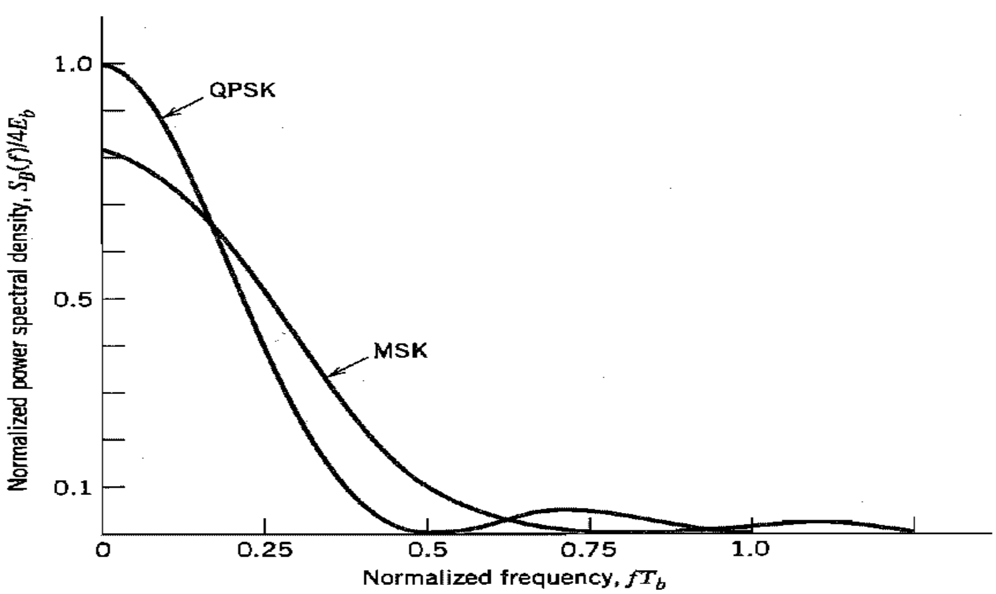


FIGURE 6.5 Power spectra of binary PSK and FSK signals.

Power spectrum of QPSK:



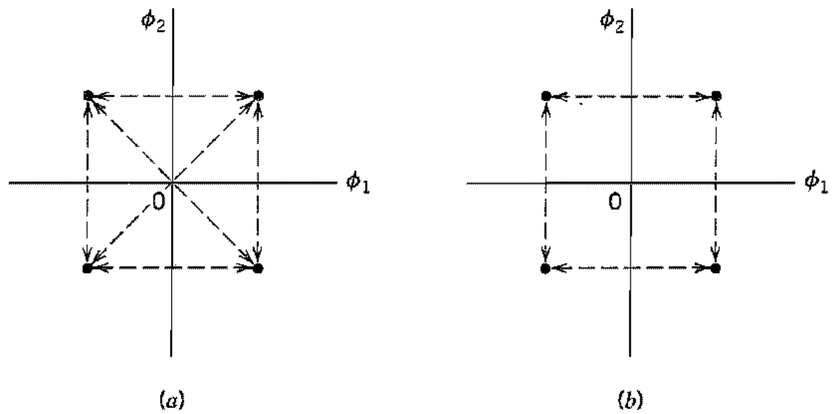


FIGURE 6.10 Possible paths for switching between the message points in (*a*) QPSK and (*b*) offset QPSK.

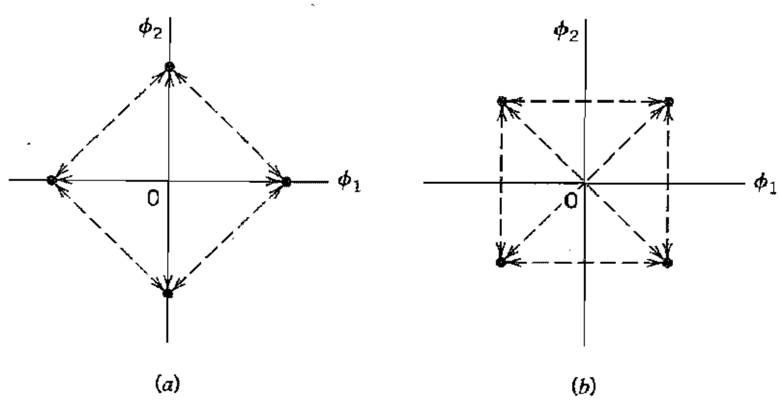


FIGURE 6.11 Two commonly used signal constellations for QPSK; the arrows indicate the paths along which the QPSK modulator can change its state.

Comparisons between Modulation Techniques:

Sr. No.	Parameter	BPSK	DPSK	QPSK	M-ary PSK	QASK	BFSK	M-ary FSK	MSK	ASK
1	Modulation of	Phase	Phase	Phase	Phase	Amplitude and phase	frequency	frequency	frequency	amplitude
2	Equation of the transmitted signal $s(t)$	$s(t) = b(t) \sqrt{2P_s} \cos(2\pi f_0 t)$	$s(t) = b(t)\sqrt{2P_s} \cos(2\pi f_0 t)$ $b(t)$ differentially coded	$s(t) = \sqrt{2P_s} \cos \left[2\pi f_0 t + (2m+1)\frac{\pi}{4}\right]$ $m = 0, 1, 2, 3$	$s(t) = \sqrt{2P_s} \cos(2\pi f_0 t + \phi_m)$ $\phi_m = (2m+1)\frac{\pi}{M}$ $m = 0, 1, 2, \ldots M-1$	$s(t) = k_1\sqrt{0.2P_s} \cos(2\pi f_0 t) + k_2\sqrt{0.2P_s} \sin(2\pi f_0 t)$ $k_1, k_2 = \pm 1$ or ± 3 for $M = 16$	$s(t) = \sqrt{2P_s} \cos[2(\pi f_0 + d(t)\Omega) t]$ Ω is frequency shift.	$s(t) = \sqrt{2P_s} \cos(2\pi f_i t)$ $i = 1, 2, \ldots M$	$s(t) = b_o(t)\sqrt{2P_s} \sin 2\pi \left[f_0 + b_e(t)\, b_o(t) \frac{f_b}{4}\right] t$ $b_e(t), b_o(t)$ = odd/even sequence	$s(t) = 2\sqrt{2P_s} \cos(2\pi f_0 t)$ for symbol '1' = 0 for symbol '0'
3	Bits per symbol	One	One	Two	N	N	One	N	Two	One
4	Number of possible symbols $M = 2^N$	Two	Two	Four	$M = 2^N$	$M = 2^N$	Two	$M = 2^N$	Four	Two
5	Detection method	Coherent	Non-Coherent	Coherent	Coherent	Coherent	Non-coherent	Non-coherent	Coherent	Coherent
6	Minimum Euclidean distance	$2\sqrt{E_b}$		$2\sqrt{E_b}$	$2\sqrt{E_s} \sin\frac{\pi}{M}$	$\sqrt{0.4E_s}$ for $M = 16$	$\sqrt{2E_b}$	$\sqrt{2NE_b}$	$2\sqrt{E_b}$	$\sqrt{E_b}$
7	Minimum bandwidth (BW)	$2f_b$	f_b	f_b	$\frac{2f_b}{N}$	$\frac{2f_b}{N}$	$4f_b$	$\frac{2^{N+1}}{N} f_b$	$1.5 f_b$	$2f_b$
8	Symbol duration (T_s)	T_b	$2T_b$	$2T_b$	NT_b	NT_b	T_b	NT_b	$2T_b$	T_b

Channel Coding

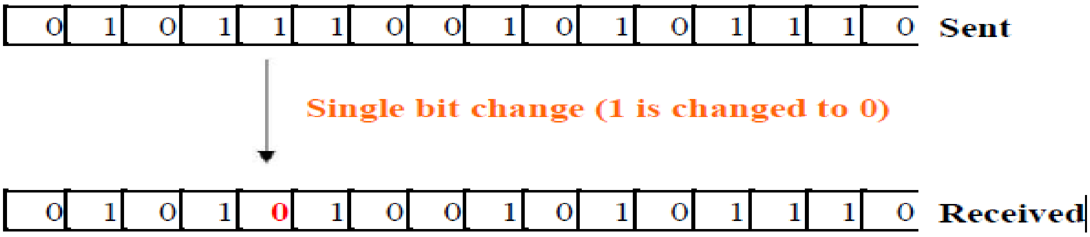
Error Detection & Correction

Environmental interference and physical defects in the communication medium can cause random bit errors during data transmission. Error coding is a method of detecting and correcting the errors

Types of Errors

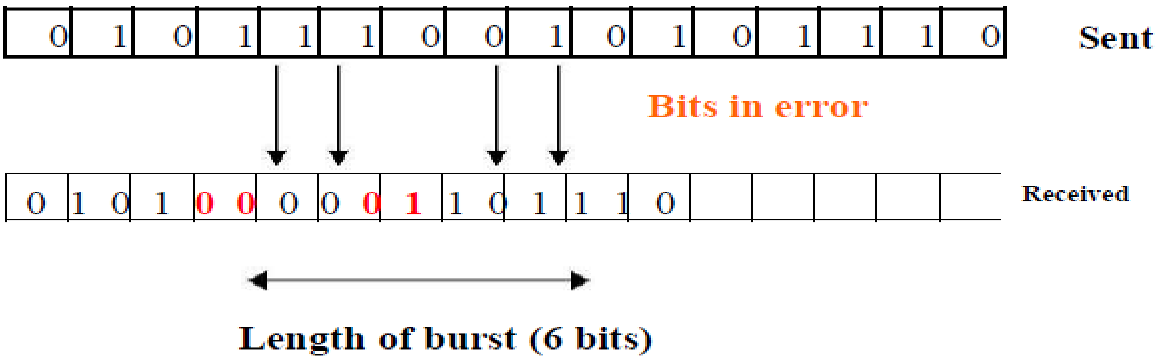
1. Single-bit error

The term single- bit error means that only one bit of given data unit (such as a byte, character, or data unit) is changed from 1 to 0 or from 0 to 1 as shown in Fig. 3.2.1.



2. Burst error

The term burst error means that two or more bits in the data unit have changed from 0 to 1 or vice-versa. Burst error doesn't necessary means that error occurs in consecutive bits. The length of the burst error is measured from the first corrupted bit to the last corrupted bit. Some bits in between may not be corrupted.



Error Detecting Codes

Basic approach used for error detection is the use of redundancy, where additional bits are added to facilitate detection and correction of errors.

Popular techniques are:

- Simple Parity check
- Two-dimensional Parity check
- Checksum
- Cyclic redundancy check

Error Correcting Codes

Error Correction can be handled in two ways.

1. One is when an error is discovered; the receiver can have the sender retransmit the entire data unit. This is known as **backward error correction**.

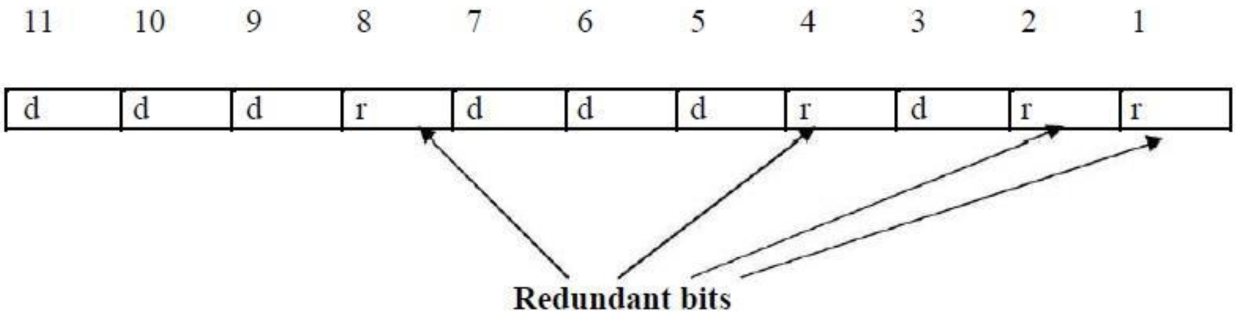
2. In the other, receiver can use an error-correcting code, which automatically corrects certain errors. This is known as **forward error correction**

Hamming Code

$2^r >= d+r+1$(to find redundant bits)

Example

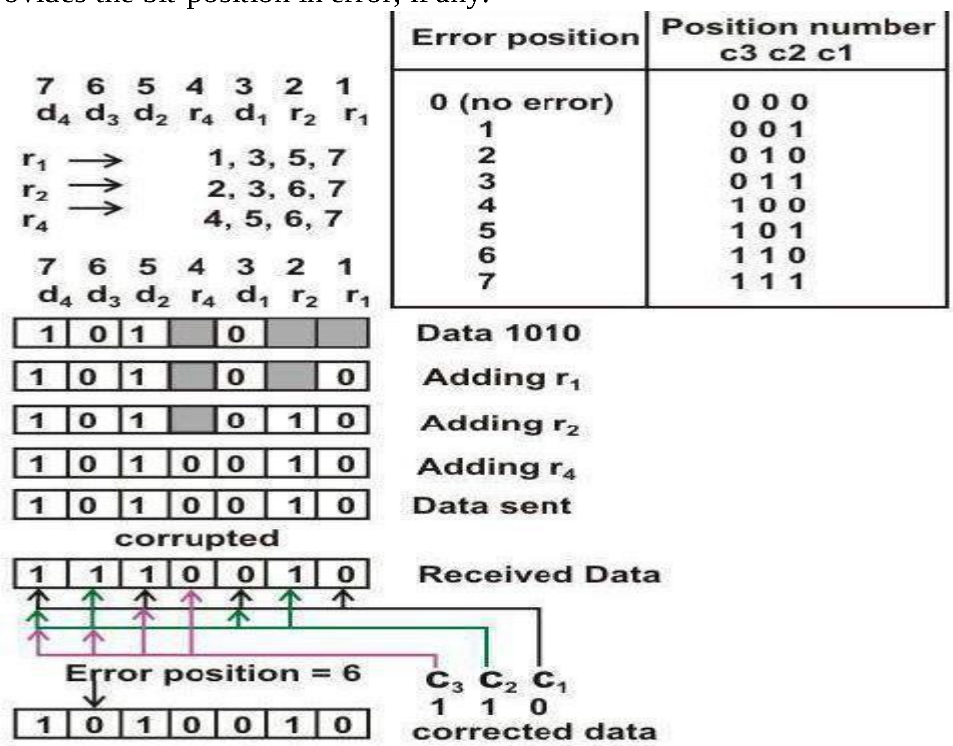
If d is 7, then the smallest value of r that satisfies the above relation is 4. So the total bits, which are to be transmitted is 11 bits (d+r = 7+4 =11).



Positions of redundancy bits in hamming code

Basic approach for error detection by using Hamming code is as follows:

- To each group of m information bits k parity bits are added to form (m+k) bit code as shown in above Fig
- Location of each of the (m+k) digits is assigned a decimal value.
- The k parity bits are placed in positions 1, 2, …, 2k-1 positions.–K parity checks are performed on selected digits of each codeword.
- At the receiving end the parity bits are recalculated. The decimal value of the k parity bits provides the bit-position in error, if any.



Repetition & Parity Check Codes

Repetition code is one of the most basic error-correcting codes. In order to transmit a message over a noisy channel that may corrupt the transmission in a few places, the idea of the repetition code is to just repeat the message several times.

Parity check is a simple way to add redundancy bits to the packets such that the total number of 1's is even (or odd).Single parity check: a single bit is appended to the end of each frame, the bit is 1 if the data portion of the frame has odd number of 1's. Otherwise, it is 0

Interleaving

Interleaving is a technique for making forward error correction more robust with respect to burst errors

Hamming distance

The error correction capability of a block code is directly related to the "Hamming distance" between each of the codewords. The Hamming distance between n-bit codewords v1 and v2 is defined

$$d(v_1,v_2) = \sum_{t=0}^{n-1} XOR(\,v1(l), v2(l))$$

This is simply the number of bits in which v1 and v2 are different.

Example: v1 = 011011 and v2 = 110001. An XOR of these codewords gives

XOR(v1, v2) = 101010. Hence the Hamming distance d(v1, v2) = 3.

Forward Error Correction (FEC) Systems

Forward error correction (**FEC**) or **channel coding** is a technique used for controlling errors in data transmission over unreliable or noisy communication channels.

Automatic Retransmission Query (ARQ) Systems

Automatic repeat request (**ARQ**), also known as **automatic repeat query**, is an error-control method for data transmission that uses **acknowledgements** (messages sent by the receiver indicating that it has correctly received a packet) and timeouts (specified periods of time allowed to elapse before an acknowledgment is to be received) to achieve reliable data transmission over an unreliable service.

If the sender does not receive an acknowledgment before the timeout, it usually re-transmits the packet until the sender receives an acknowledgment or exceeds a predefined number of retransmissions.

The types of ARQ protocols include

Stop-and-wait ARQ

Go-Back-N ARQ and

Selective Repeat ARQ/Selective Reject ARQ.

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 spectral density 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, the error 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.

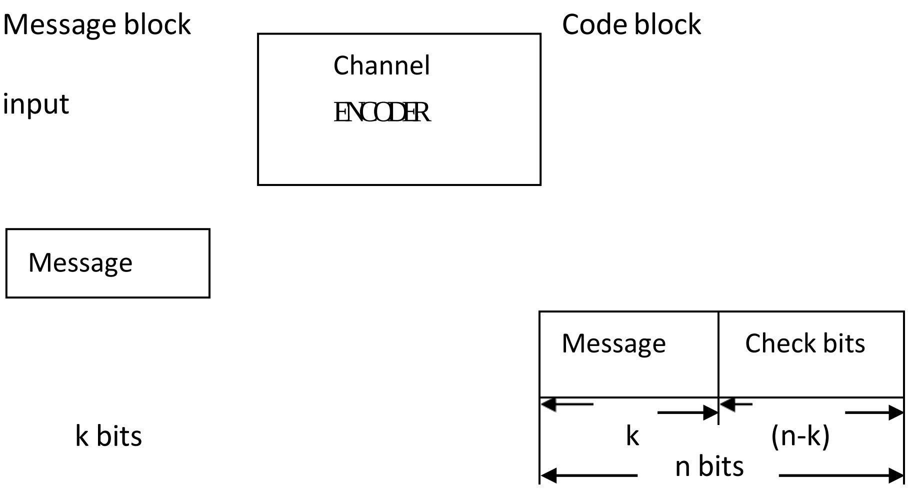


Figure: Functional block diagram of block coder

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 $q=n-k$

Whereq 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

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

(P matrix | k × k identity matrix)

where $p_{ii} = 0$ or 1

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

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

The components of **v** are

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

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

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** = ($u0$, $u1$, $u2$, $u3$) be the message to be encoded and **v** = ($v0$, $v1$, $v2$, $v3$, $v4$, $v5$,$v6$) be the 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

$$\mathbf{H} = [\mathbf{I}_{n-k} \quad \mathbf{P}^T]$$

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

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

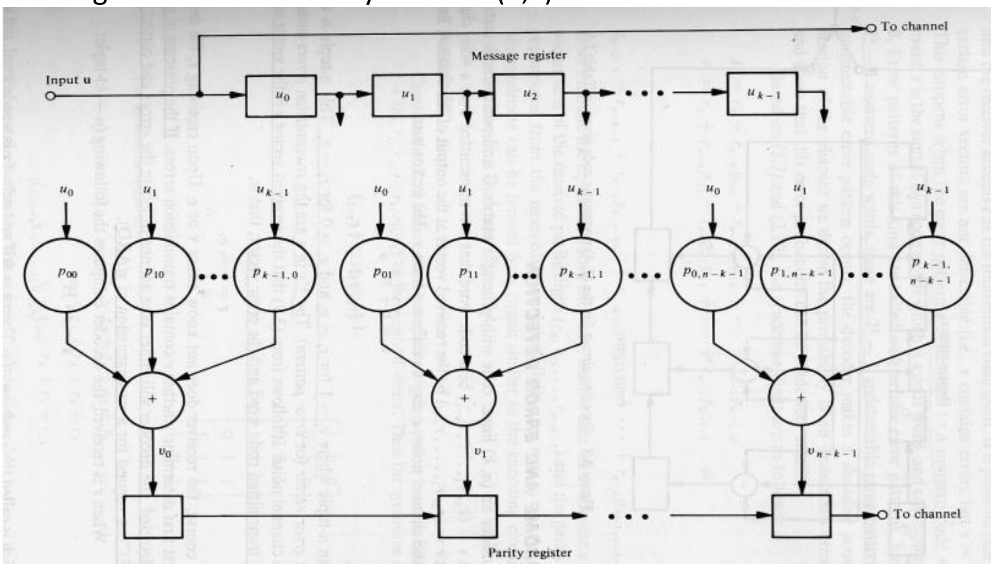


Figure: Encoding Circuit

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.

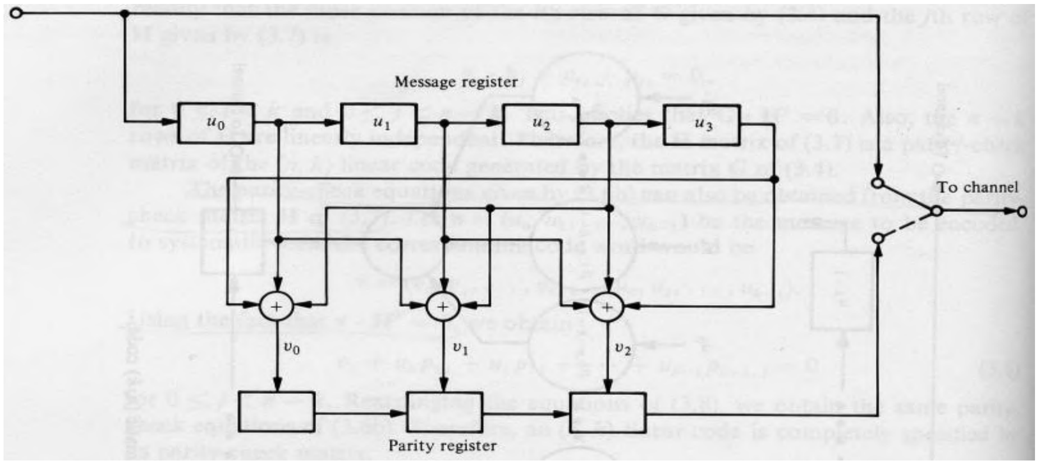
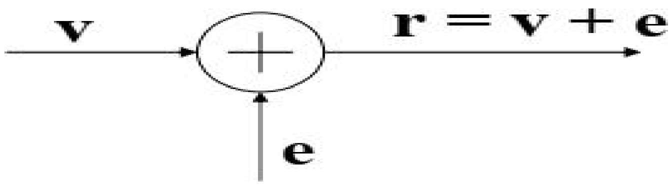


Figure: Encoding Circuit for (7,4) code

Syndrome and Error Detection

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



Where

e = r + v = (e_0, e_1, …, 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 • H^T
s = (s_0, s_1, …, 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≠ 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 • HT = 0), error patterns of this kind are called undetectable error patterns. Since there are 2k – 1 non-zero code words, there are 2k – 1 undetectable error patterns. The syndrome digits are as follows:

s0 = r0 + rn-k p00 + rn-k+1 p10 + ⋯ + rn-
1 pk-1,0 s1 = r1 + rn-k p01 + rn-k+1 p11 +
⋯ + rn-1 pk-1,1

.

sn-k-1 = rn-k-1 + rn-k p0,n-k-1 + rn-k+1 p1,n-k-1 + ⋯ + rn-1 pk-1,n-k-1

The syndrome s is the vector sum of the received parity digits (r0,r1,…,rn-k-1) and the parity- check digits recomputed from the received information digits (rn-k,rn-k+1,…,rn-1).

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

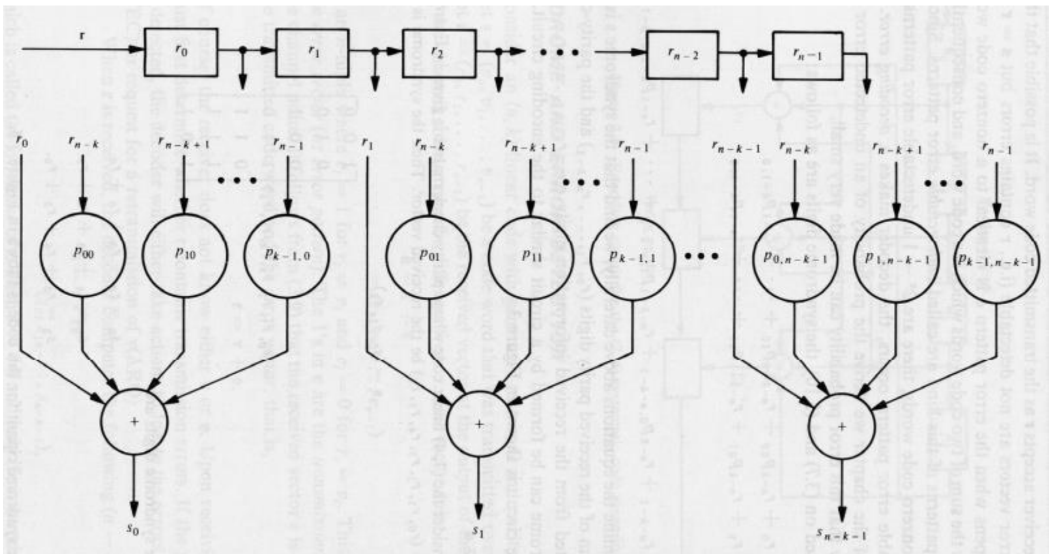


Figure: Syndrome Circuit

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 its syndrome 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 = [Im 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

Solution

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

Therefore, $\mathbf{G} = \begin{bmatrix} 1 & 1 & 1 & 0 & \\ 1 & 1 & 0 & 1 & \vdots\mathbf{I}_4 \\ 0 & 1 & 1 & 1 & \\ 1 & 0 & 1 & 1 & \end{bmatrix}$

and then $\mathbf{H} = \begin{bmatrix} & 1 & 1 & 0 & 1 \\ \mathbf{I}_4\vdots & 1 & 1 & 1 & 0 \\ & 1 & 0 & 1 & 1 \\ & 0 & 1 & 1 & 1 \end{bmatrix}$

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

(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.

$$\mathbf{s} = \mathbf{r}\mathbf{H}^{T} = [1\,1\,1\,0\,0\,0\,0\,0]\left[\mathbf{I}_4 \vdots \begin{matrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{matrix}\right]^{T} = [1\,1\,1\,0] \neq \mathbf{0}$$

Convolution

codes

$$x_1 = m \oplus m_1 \oplus m_2 \quad \text{... (4.4.1)}$$

and

$$x_2 = m \oplus m_2 \quad \text{... (4.4.2)}$$

The output switch first samples x_1 and then x_2. The shift register then shifts contents of m_1 to m_2 and contents of m to m_1. Next input bit is then taken and stored in m. Again x_1 and x_2 are generated according to this new combination of 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,

$$X = x_1 x_2 x_1 x_2 x_1 x_2 \ldots \text{ and so on} \quad \text{... (4.4.3)}$$

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,

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

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 $\rightarrow$ Message bit is entered in position 'm'.

Second shift $\rightarrow$ Message bit is shifted in position m_1.

Third shift $\rightarrow$ 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 $\{g_0^{(1)}, g_1^{(1)}, g_2^{(1)} ...... g_m^{(1)}\}$ denote the impulse response of the adder which generates x_1 in Fig. 4.4.1 Similarly, Let the sequence $\{g_0^{(2)}, g_1^{(2)}, g_2^{(2)} ...... g_m^{(2)}\}$ 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 $\{m_0, m_1, m_2 ....\}$. 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,

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

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

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

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,

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

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

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

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.,

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

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

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.,

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

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

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

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.

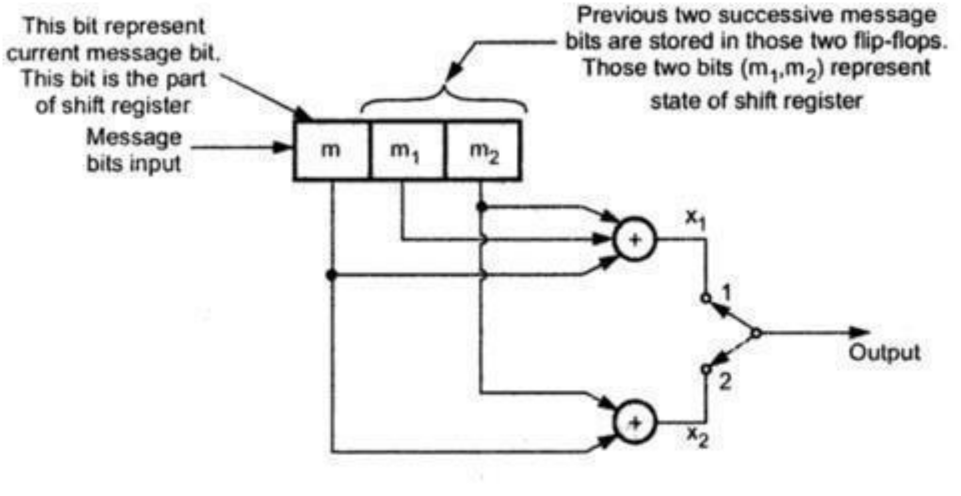


Fig. 4.4.4 Convolutional encoder with k = 1 and n = 2

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

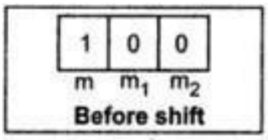
Table 4.4.1 States of the encoder of Fig. 4.4.4

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.

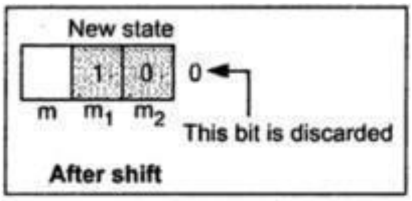
1) When $m = 1$ i.e. first bit

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.

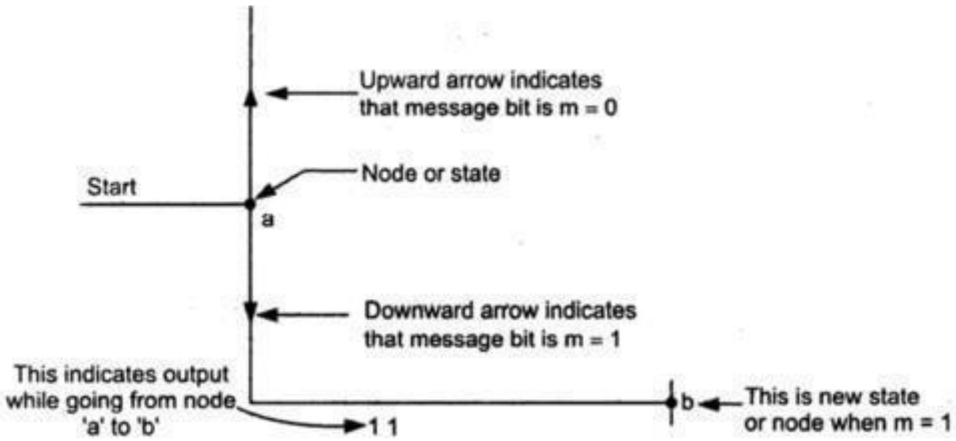
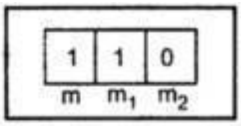


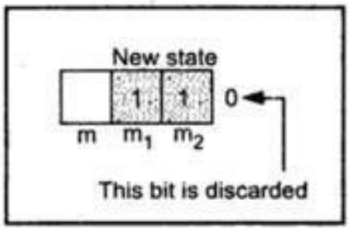
Fig. 4.4.5 Code tree from node *'a'* to *'b'*

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.*

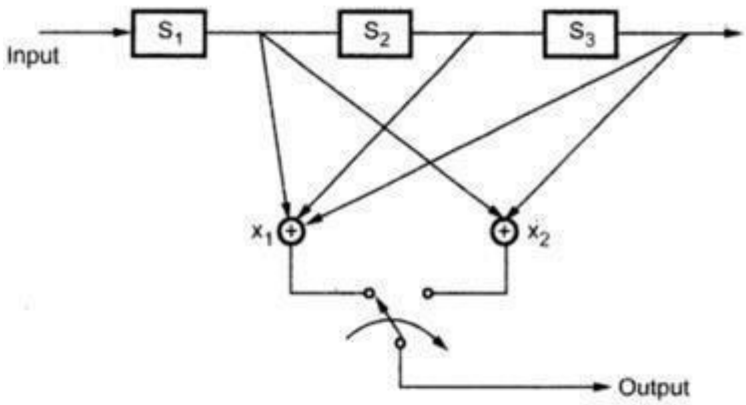


Fig. 4.4.32 Convolutional encoder of example 4.4.8

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'

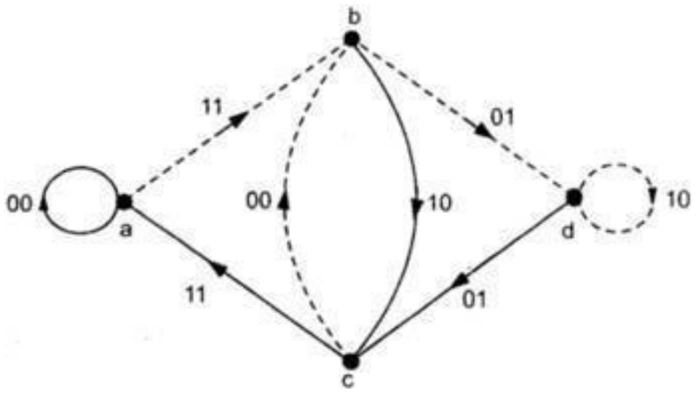
$s_3 s_2 = 1\,1$, state 'd'

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.

Decoding methods of Convolution code:

1.Veterbi decoding 2.Sequential

decoding 3.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 data Hence

there areno code vectorsand blocks such as.

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

Surviving path: this is the path of the decoded signal with minimum metric In

veterbi decoding ametric is assigned 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.