DC

Unit - 1

Random Processes & Noise

1.1 Introduction:

Random signals and noise are present in several engineering systems. Practical signals seldom lend themselves to a nice mathematical deterministic description. It is partly a consequence of the chaos that is produced by nature. However, chaos can also be man-made, and one can even state that chaos is a condition sine qua non to be able to transfer information. Signals that are not random in time but predictable contain no information, as was concluded by Shannon in his famous communication theory.

A (one-dimensional) random process is a (scalar) function y(t), where t is usually time, for which the future evolution is not determined uniquely by any set of initial data or at least by any set that is knowable to you and me. In other words, random process" is just a fancy phrase that means unpredictable function". Random processes y takes on a continuum of values ranging over some interval, often but not always $-\infty$ to $+\infty$. The generalization to y's with discrete (e.g., integral) values is straightforward

Examples of random processes are:

(i) The total energy E(t) in a cell of gas that is in contact with a heat bath;

(ii) The temperature T(t) at the corner of Main Street and Center Street in Logan, Utah;

(iii) The earth-longitude (t) of a specific oxygen molecule in the earth's atmosphere.

One can also deal with random processes that are vector or tensor functions of time. Ensembles of random processes. Since the precise time evolution of a random process is not predictable, if one wishes to make predictions one can do so only probabilistically. The foundation for probabilistic predictions is an ensemble of random processes |i.e., a collection of a huge number of random processes each of which behaves in its own, unpredictable way. The probability density function describes the general distribution of the magnitude of the random process, but it gives no information on the time or frequency content of the process. Ensemble averaging and Time averaging can be used to obtain the process properties.

1.2 Mathematical definition of a random process:

Two Ways to View a Random Process

- A random process can be viewed as a function X (t, ω) of two variables, time $t \in T$ and the outcome of the underlying random experiment $\omega \in \Omega$.

- For fixed t, X (t, ω) is a random variable over Ω.

- For fixed ω, X (t, ω) is a deterministic function of t, called a sample function.

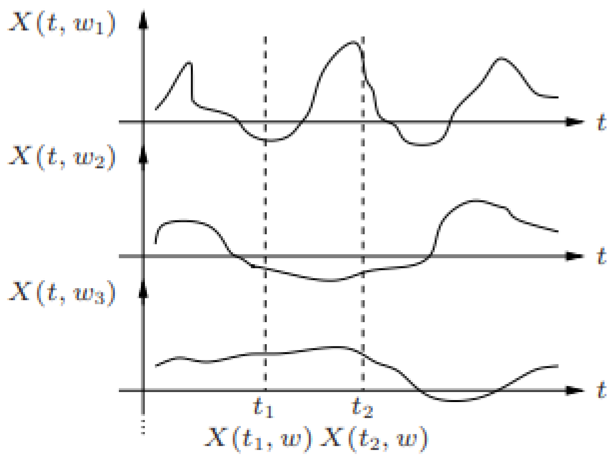


Fig: Random Process

A random process is said to be discrete time if T is a countably infinite set, e.g.,

N = {0, 1, 2, . . .}

Z = {. . . , −2, −1, 0, +1, +2, . . .}

- In this case the process is denoted by Xn, for n ∈N, a countably infinite set, and is simply an infinite sequence of random variables

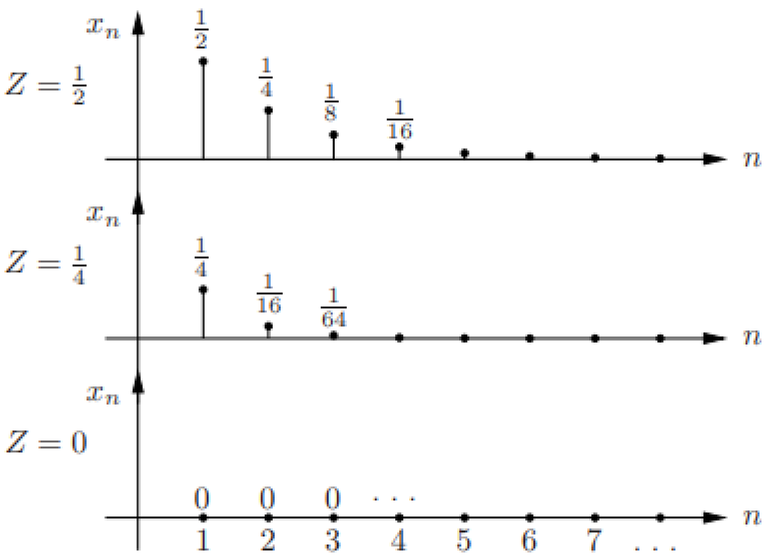
- A sample function for a discrete time process is called a sample sequence or sample path

- A discrete-time process can comprise discrete, continuous, or mixed r.v.s

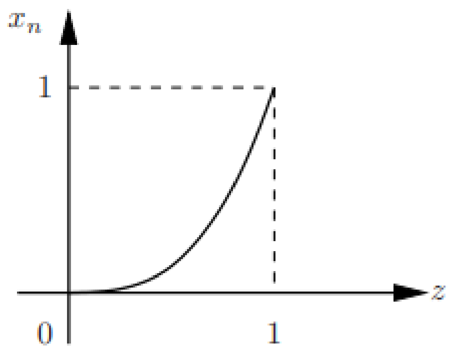
Example

- Let Z ~U [0, 1], and define the discrete time process Xn = Z n for n ≥ 1.

- Sample paths:



First-order pdf of the process: For each n, Xn = Z n is a r.v.; the sequence of pdfs of Xn is called the first-order pdf of the process



Since Xn is a differentiable function of the continuous r.v. Z, we can find its pdf as f

$f_{Xn}(x) = 1/(nx^{(n-1)/n}) = 1/n\ x^{(1/n)-1}$ $0 \leq x \leq 1$.

1.3 Stationary processes:

A **stationary process** is a stochastic process whose unconditional joint probability distribution does not change when shifted in time. Consequently, parameters such as mean and variance also do not change over time.

A random process is said to be stationary if its statistical characterization is independent of the observation interval over which the process was initiated. Ensemble averages do not vary with time. An ensemble of random processes is said to be stationary if and only if its probability distributions pn depend only on time differences, not on absolute time:

$$p_n(y_n;t_n + \tau;\ldots;y_2;t_2 + \tau;y_1;t_1 + \tau) = p_n(y_n;t_n;\ldots;y_2;t_2;y_1;t_1)$$

If this property holds for the absolute probabilities pn. Most stationary random processes can be treated as ergodic. A random process is ergodic if every member of the process carries with it the complete statistics of the whole process. Then its ensemble averages will equal appropriate time averages. Of necessity, an ergodic process must be stationary, but not all stationary processes are ergodic.

1.4 Mean:

The mean (or average), mu, of the sample is the first moment:

$$\mu = \frac{1}{N}\sum_{i=1}^{N} X_i$$

You will also see this notation sometimes

$$\bar{X} = \frac{1}{N}\sum_{i=1}^{N} X_i$$

In R, the mean () function returns the average of a vector.

1.5 Correlation & Covariance function:

Correlation

The covariance has units (units of X times units of Y), and thus it can be difficult to assess how strongly related two quantities are. The correlation coefficient is a dimensionless quantity that helps to assess this.

The correlation coefficient between X and Y normalizes the covariance such that the resulting statistic lies between -1 and 1. The Pearson correlation coefficient is

$$Cor\ (X,Y) = \frac{1}{n-1}\sum_{i=1}^{N}\frac{(X_i - \mu x)(Y_i - \mu_Y)}{\sigma_X \sigma_Y}$$

The correlation matrix for X and Y is

$$C = \begin{pmatrix} 1 & Cor\ (X,Y) \\ Cor\ (X,Y) & 1 \end{pmatrix}$$

Covariance

If we have two samples of the same size, X_i, and Y_i, where i=1,…,n, then the covariance is an estimate of how variation in X is related to variation in Y. The covariance is defined as

$$Cov\,(X,Y) = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \mu_X)(Y_i - \mu_Y)$$

Where μ_X is the mean of the X sample, and μ_Y is the mean of the Y sample. Negative covariance means that smaller X tend to be associated with larger Y (and vice versa). Positive covariance means that larger/smaller X are associated with larger/smaller Y. Note that the covariance of X and Y is exactly the same as the covariance of Y and X. Also note that the covariance of X with itself is the variance of X. The covariance matrix for X and Y is thus

$$V = \begin{pmatrix} \sigma_X^2 & Cor\,(X,Y) \\ Cor\,(X,Y) & \sigma_Y^2 \end{pmatrix}$$

1.6 Ergodic processes:

- In econometrics and signal processing, a stochastic process is said to be **ergodic** if its statistical properties can be deduced from a single, sufficiently long, random sample of the process.

- The reasoning is that any collection of random samples from a process must represent the average statistical properties of the entire process.

- In other words, regardless of what the individual samples are, a birds-eye view of the collection of samples must represent the whole process.

- Conversely, a process that is not ergodic is a process that changes erratically at an inconsistent rate.

1.7 Transmission of a random process through a LTI filter:

A **linear time-invariant (LTI)** system can be represented by its **impulse response**. More specifically, if X(t)X(t) is the input signal to the system, the output, Y(t), can be written as

$$Y(t) = \int_{-\infty}^{\infty} h(\alpha) X(t-\alpha)\, d\alpha$$

$$= \int_{-\infty}^{\infty} X(\alpha) h(t-\alpha)\, d\alpha$$

The above integral is called the convolution of h and X, and we write

$$Y(t) = h(t) * X(t) = X(t) * h(t)$$

Note that as the name suggests, the impulse response can be obtained if the input to the system is chosen to be the unit impulse function (delta function) $x(t)=\delta(t)$. For discrete-time systems, the output can be written as

$$Y(n) = h(n) * X(n) = X(n) * h(n)$$

$$= \sum_{k=-\infty}^{\infty} h(k) X(n-k)$$

$$= \sum_{k=-\infty}^{\infty} X(k) h(n-k)$$

The discrete-time unit impulse function is defined as $\delta(n) = 1$, n=0

= 0, otherwise

For the rest of this chapter, we mainly focus on continuous-time signals.

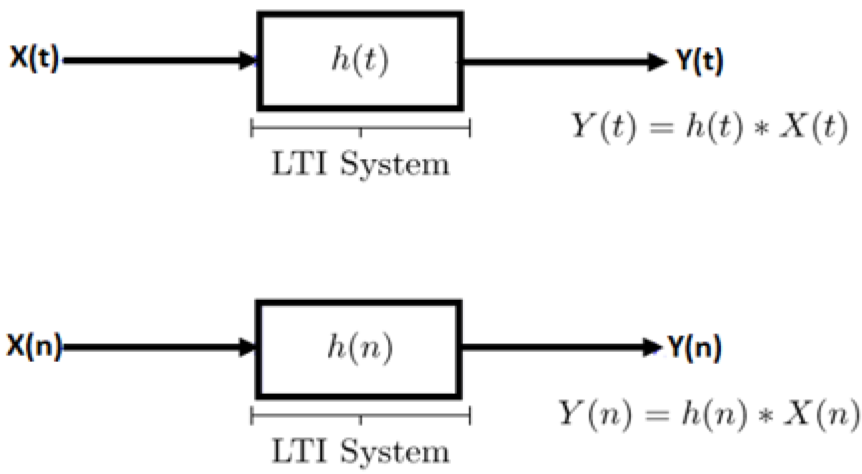


Fig: LTI system

LTI Systems with Random Inputs:

Consider an LTI system with impulse response h(t).

Let X(t) be a WSS random process. If X(t) is the input of the system, then the output, Y(t), is also a random process. More specifically, we can write

$Y(t)=h(t)*X(t)=\int_{-\infty}^{\infty} h(\alpha)X(t-\alpha)d\alpha$

Here, our goal is to show that X(t) and Y(t) are jointly WSS processes. Let's first start by calculating the mean function of Y(t), $\mu Y(t)$. We have

$\mu_Y(t)=E[Y(t)]$

$=E[\int_{-\infty}^{\infty} h(\alpha)X(t-\alpha)d\alpha]$

$=\int_{-\infty}^{\infty} h(\alpha)E[X(t-\alpha)]d\alpha$

$=\int_{-\infty}^{\infty} h(\alpha)\, \mu X d\alpha$

$=\mu_X \int_{-\infty}^{\infty} h(\alpha)\, d\alpha$

We note that $\mu Y(t)$is not a function of t, so we can write

$\mu_Y(t)=\mu_Y=\mu_X \int_{-\infty}^{\infty} h(\alpha)\, d\alpha$.

Let's next find the cross-correlation function, $R_{XY}(t1, t2)$.

We have

$R_{XY}(t1, t2)=E[X(t1)Y(t2)]=E[X(t1)\int_{-\infty}^{\infty} h(\alpha)\, X(t\mathbf{2}-\alpha)d\alpha]$

$=E[\int_{-\infty}^{\infty} h(\alpha)X(t\mathbf{1})X(t\mathbf{2}-\alpha)d\alpha]$

$=\int_{-\infty}^{\infty} h(\alpha)E[X(t\mathbf{1})X(t\mathbf{2}-\alpha)]d\alpha$

$=\int_{-\infty}^{\infty} h(\alpha)RX(t\mathbf{1},t\mathbf{2}-\alpha)d\alpha$

$=\int_{-\infty}^{\infty} h(\alpha)RX(t\mathbf{1}-t\mathbf{2}+\alpha)d\alpha$ (since X(t) is WSS).

We note that $R_{XY}(t1, t2)$is only a function of $\tau=t1-t2$, so we may write

$R_{XY}(\tau)=\int_{-\infty}^{\infty} h(\alpha)RX(\tau+\alpha)d\alpha$

$=h(\tau)*R_X(-\tau)$

$=h(-\tau)*R_X(\tau)$.

Similarly, you can show that

$R_Y(\tau)=h(\tau)*h(-\tau)*R_X(\tau)$.

From the above results we conclude that X(t)and Y(t)are jointly WSS.

Numerical:

Let X(t)be a zero-mean WSS process with $RX(\tau)=e-|\tau|$. X(t)is input to an LTI system with

$|H(f)|=\sqrt{(1+4\pi^2f^2)}$ $|f|<2$

= 0 otherwise,

Let Y(t) be the output.

1. Find $\mu_Y(t)=E[Y(t)]$
2. Find $RY(\tau)$
3. Find $E[Y(t)^2]$

Solution

Note that since X(t)is WSS, X(t)and Y(t)are jointly WSS, and therefore Y(t)is WSS.

1. To find $\mu_Y(t)$, we can write

$\mu_Y=\mu_X H(0)=0*1=0$

b. To find $R_Y(\tau)$, we first find $S_Y(f)$

$S_Y(f)=S_X(f)|H(f)|^2$

From Fourier transform tables, we can see that

$S_X(f)=F\{e^{-|\tau|}\} = 2/(1+(2\pi f)^2)$

Then, we can find $S_Y(f)$as

$SY(f)=S_X(f)|H(f)|^2$

$=2 |f|<2$

$= 0$ otherwise

We can now find $R_Y(\tau)$by taking the inverse Fourier transform of $S_Y(f)$

$R_Y(\tau)=8\text{sinc}(4\tau)$,

Where

$\text{Sinc}(f)=\sin(\pi f)/\pi f$

c. We have

$E[Y(t)^2] = RY(0)=8$.

1.8 Power spectral density

A Power Spectral Density (PSD) is the measure of signal's power content versus frequency. A PSD is typically used to characterize broadband random signals. The amplitude of the PSD is normalized by the spectral resolution employed to digitize the signal.

For vibration data, a PSD has amplitude units of g2/Hz. While this unit may not seem intuitive at first, it helps ensure that random data can be overlaid and compared independently of the spectral resolution used to measure the data.

The average power P of a signal x(t) over all time is therefore given by the following time average:

$$P = \lim_{T\to\infty} \frac{1}{T}\int_0^T |x(t)|^2 dt$$

1.9 Mathematical Representation of Noise: Some Sources of Noise

The term noise is used customarily to designate unwanted signals that tend to disturb the transmission and processing of signals in communication systems, and over which we have incomplete control. In practice, we find that there are many potential sources of noise in a communication system. The sources of noise may be external to the system (e.g., atmospheric noise, galactic noise, man-made noise) or internal to the system. The second category includes an important type of noise that arises from the phenomenon of spontaneous fluctuations of current flow that is experienced in all electrical circuits. In a physical context, the most common examples of the spontaneous fluctuation phenomenon are shot noise which arises because of the discrete nature of current flow in electronic devices; and thermal noise, which is attributed to the random motion of electrons in a conductor.

Internal Noise

Also called fundamental noise which is originated within electronic devices or circuits. They are called fundamental sources because they are the integral part of the physical nature of the material used for making electronic components.

This type of noise follows certain rules. Therefore, it can be eliminated by properly designing the electronic circuits and equipment.

The fundamental noise sources produce different types of noise. They are as follows.

(i) Thermal noise

(ii) Shot noise

(iii) Partition noise

(iv) Low frequency or flicker noise

(v) High frequency or transit time noise

Shot Noise

It is produced due to shot effect. Due to the shot effect shot noise is produced in all the amplifying devices rather in all the active devices.

The shot noise is produced due to the random variations in the arrival of electronics (or holes) at the output electrode of an amplifying devices.

Therefore, it appears as a randomly varying noise current superimposed and the output. The shot noise "Sounds" like a shower of lead shots falling on a metal sheet.

The shot noise has a uniform spectral density like thermal noise

The mean square shot noise current for a diode is given as

$In^2 = 2\ (J + Io)\ qB\ Amp^2$

Where I = direct current across the junction in amperes

Io = reverse saturation current in ampere

q = electron charge = 1.6×10^{-19} C

B = Effective noise Band width in Hz

Thermal Noise Also known as Johnson Noise.

The free electrons with in a conductor are always in random motion.

This random motion is due to the thermal energy received by them. The distribution of these free electrons with in a conductor at a given instant of time is not uniform.

It is possible that an excess number of electrons may appears at one end or the other of the conductor

The average voltage resulting from this non uniform distribution is zero but the average power is not zero

As this power results from the thermal energy are called as the "Thermal noise power"

The average thermal noise power is given by

Pn = KTB Watts

Where: k = Boltzmann's constant = 1.38 X 10^{-23} J/K

B = Bandwidth of the noise spectrum (Hz)

T = Temperature of the conductor in K

The thermal noise power Pn in proportional to the noise BW and conductor temperature

Flicker noise: Also known as low frequency noise

The flicker noise appears at frequencies below a few kHz it is sometimes called as 1/f noise in the semiconductor devices, the flicker noise is generated due to the fluctuations in the carrier density.

These fluctuations in the carrier density will cause the fluctuation in the conductivity of the material. This will produce a fluctuating voltage drop when a direct-current flow through a device this fluctuating voltage is called as flicker noise voltage.

The mean square value of flicker noise voltage is proportional to the square of direct current flowing through the device.

High frequency noise Also called Transit time noise

If the time taken by an electron to travel from the emitter to the collector of a transistor becomes comparable to the period of the signal which is being period of the signal which is being amplified then the transit time effect takes place.

This effect is observed at every high frequency. Due to the transit time effect some of the carriers may diffuse back to the emitter. This gives rise to an input admittance, the conductance component of which increases with frequency.

The minute current induced in the input of the device by the random fluctuations in the output current, will create random noise at high frequencies.

Partition noise- Partition noise is generated when the current gents divided between two or more paths. It is generated due to the random fluctuations in the division.

Therefore, the partition noise in a transistor will be higher than that in a diode.

Noise figure: It is the ratio of the noise power input to noise power output due to source resistance. The noise figure is a quantity which compares the noise in an actual amplifier with that in an ideal (noise less) amplifier. Noise figure is expressed in decibels.

S_{pi} (S_{vi})= Signal power (voltage) input

N_{pi} (N_{vi})= Noise power (voltage) input due to Rs

S_{po} (S_{vo})= Signal power (voltage) output

N_{po} (N_{vo})= Noise power (voltage output due to Rs and any noise source within the active device

Noise figure, $NF = 10log\left(\frac{\text{Total noise power input}}{\text{Noise power output due to } R_s}\right)$

$$= 10log\left(\frac{N_{po}}{A_p N_{pi}}\right)$$

Where, the power gain of the active devices is

$$A_p = \frac{S_{po}}{S_{pi}}$$

Hence, $= 10log\left(\frac{N_{po}S_{vi}}{S_{po}N_{vi}}\right)$

$$= 10log\left(\frac{S_{pi}/N_{pi}}{S_{po}/N_{po}}\right)$$

The quotient Sp/Np is called the signal to noise power ratio.

The noise figure is the Input signal to noise signal to noise power ratio

$$NF = 10log\left(\frac{S_{pi}/N_{pi}}{S_{po}/N_{po}}\right) = 20log\left(\frac{S_{vi}}{N_{vi}}\right) = 20log\left(\frac{S_{vo}}{N_{vo}}\right)$$

Where: $\frac{Sv}{Nv}$ is called signal to noise Voltage ratio

Losses

An audio sinusoidal generator Vs with source resistance Rs is connected to the input of Q. The active device is cascaded with a low noise amplifier and filter and the output of this system is measured on a true rms reading voltmeter M. The experimental procedure for determining NF is as follows-

i) Measure Rs and calculate Nvi = Vn

Ii) Adjust the audio signal voltage so that it is 10 times the noise voltage

V/s = 10 Vn or Svi = 10 Nvi

Measure the output voltage with M.

A system is used to measure the noise figure of an active device Q.

Large signal to noise ratio, we may neglect the noise and assume that the voltmeter reading gives the signal output voltage Svo

Iii) Set Vs = 0 and measure the output voltage N_{vo} with M. The low noise amplifier is required only it the noise output of Q is to low to be detected with M.

Friis Formula: This formula was named after Danish American Electrical Engineer Harald T. Friis. His formula is used in tele communications engineering to calculate the signal to noise ratio of a multistage amplifier.

He gave two formulars on relates to noise factor while the other relates to noise temperature.

Friis formula is used to calculate the total noise factor of a cascade of stage each with its own noise factor and power gain.

The total noise factor can then be used to calculate the total noise figure.

The total noise figure is given as

$$F_{total} = F_1 + \frac{F_2 - 1}{G_1} + \frac{F_3 - 1}{G_1 G_2} + \dots + \frac{F_{n-1}}{G_1 G_2 - G_{m1}}$$

Where Fi and Gi are the noise factor and available power gain respectively of the stage and ‘n’ is the number of stages

Derivation – Let us consider a 3-stage cascaded amplifier that is n=3

A source outputs a signal of power Si and noise of power Ni

Therefore, the SNR at input of the chain

$$SNR_i = \frac{S_i}{N_i}$$

This Si gets amplified by all the 3 amplifiers and the output is given as

$S_0 = S_i G_1 G_2 G$

Using the definition of the Noise factors of the amplifiers we get the final result

$$F_{total} = 1 + \frac{Na_1}{NiG_1} + \frac{Na_2}{NiG_1G_2} + \frac{Na_3}{NiG_1G_2G_3}$$

$$= F_1 = \frac{F_2 - 1}{G_1} = \frac{F_3 - 1}{G_1G_2}$$

$$F_{total} = F_1 = \frac{F_2 - 1}{G_1} = \frac{F_3 - 1}{G_1G_2}$$

Note – Friis formula for noise temperature is given as

$$T_{eq} = T_1 + \frac{T_2}{G_1} + \frac{T_3}{G_1 G_2} + \dots$$

Key takeaway

The noise power at the output of the chain consists of following 4 parts

i) Amplified noise i.e., $NiG_1G_2G_3$

Ii) The output noise of first amplifier Na1 which gets further amplified by 2nd and 3rd amplifier i.e., $Na_1G_2G_3$

Iii) Similarly output noise of 2nd amplifier Na2 amplified by 3rd i.e., Na_2G_3

Iv) Output noise of 3rd amplifier Na_3

Therefore, the total noise power at the output equals

$$N_o = NiG_1G_2G_3 + Na_1G_2G_3 + Na_2G_3 + Na_3$$

And the NSR at output becomes

$$SNR_o = \frac{SiG_1G_2G_3}{NiG_1G_2G_3 + Na_1G_2G_3 + Na_2G_3 + Na_3}$$

The Total noise factor or noise figure may now be calculated as

$$F_{total} = \frac{SNR_i}{SNR_o} = \frac{Si/N_i}{SiG_1G_2G_3/(NiG_1G_2G_3 + Na_1G_2G_3 + Na_2G_3 + Na_3)}$$

$$= 1 + \frac{Na_1}{NiG_1} + \frac{Na_2}{NiG_1G_2} + \frac{Na_3}{NiG_1G_2G_3}$$

Numericals:

1) A noise generator using diode is required to produce 15μV noise voltage in a receiver which has an input impedance of 75Ω (Purely resistive). The receiver has a noise power bandwidth of 200KHz calculate the current through the diode.

Sol- Given: $V_n = 15\mu V$

$R = 75\Omega$

B = 200 KHz

Noise current flowing through 75Ω is

$$I_n = \frac{V_n}{R} = 15 \times 10^{-6}$$

$= 0.2X10^{-6}$

$= 0.2\mu a$

Diode current –short noise current is given by

$I^2_n = 2\ (I + I_o)\ qB$

$I^2_n = 2I \ X\ Q\ X\ B$ (neglecting I_o)

$(0.2\ X\ 10^{-6})\ ^2 = 2I\ X\ 1.6\ X\ 10^{-19}\ X\ 200\ X\ 103$

I = 0.625 A = 625 μA

2. A receiver has a noise power band width of 12KHz. A register which watches with the receiver input impedance is connected across the antenna terminals what is the noise power contributed by this register in the receiver bandwidth. Assume temperature to be 30°C

Sol – Given: B = kHz; T = 30°C = 30+ 273 = 303 K

Pn = KTB

$= 1.38\ X\ 10^{-23}\ X\ 303\ X\ 12\ X\ 103$

$= 5.01768\ X\ 10^{-17}$ W

1.10 Frequency-domain Representation of Noise

The frequency domain representation of signal makes the analysis simple. The system performance is affected due to presence of these unwanted signals in form of noise. So, to make our study easy we use frequency domain analysis for study of noise. Consider a sample noise signal shown below having interval of T. A periodic noise is generated after T interval. The Fourier Series representation of the signal is given as

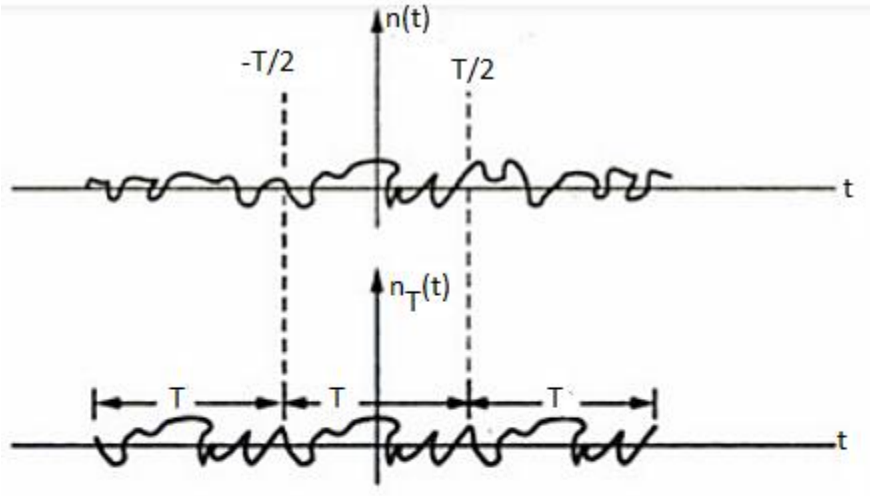


Fig 3 Sample noise waveform

$$x(t) = a_0 + \sum_{n=1}^{\infty} a_n cos\left(\frac{2\pi nt}{T_0}\right) + \sum_{n=1}^{\infty} b_n sin\left(\frac{2\pi nt}{T_0}\right)$$

$x(t) = n_T^{(s)}(t)$ i.e., signal of Fig.3.6.1(b)

$a_0 = 0;$ assuming that noise signal has no d.c. Component

$T_0 = T = \frac{1}{\Delta f}$ i.e., period of noise signal

The above equation can be written as

$$n_T(t) = \sum_{n=1}^{\infty} a_n cos\left(\frac{2\pi nt}{T}\right) + \sum_{n=1}^{\infty} b_n sin\left(\frac{2\pi nt}{T}\right)$$

$$= \sum_{n=1}^{\infty} a_n cos\,(2\pi n\Delta ft) + \sum_{n=1}^{\infty} b_n sin\,(2\pi n\Delta ft)$$

The polar form of Fourier series is represented as

$$x(t) = D_0 + \sum_{n=1}^{\infty} D_n cos\left(\frac{2\pi nt}{T_0} + \phi_n\right)$$

$$D_0 = a_0 = 0$$

$D_n = \sqrt{a_n^2 + b_n^2}$ and $\phi_n = -\,tan^{-1}\frac{b_n}{a_n}$

$$n_T(t) = \sum_{n=1}^{\infty} D_n cos\left(\frac{2\pi nt}{T_0} + \phi_n\right)$$

$$= \sum_{n=1}^{\infty} D_n cos\left(2\pi nf_{\Delta}t + \phi_n\right)$$

By Parseval's Theorem

$$P = \sum_{n=-\infty}^{\infty} |C_n|^2$$

$$P_n = |C_n|^2$$

The average power P is area under the power spectral density curve

$$P = \int_{-\infty}^{\infty} S(f)df$$

$$P = \sum_{n=0}^{\infty} S(n_{\Delta}f)_{\Delta}f$$

For one component

$$P_n = S(n_{\Delta}f)_{\Delta}f$$

$$S(n_{\Delta}f) = \frac{P_n}{\Delta f}$$

The above equation gives the PSD at frequency component $n^{\Delta}f$ where $^{\Delta}f = 1/T$

$$S(n_{\Delta}f) = \frac{|C_n|^2}{\Delta f}$$

By Fourier Series relation we have

$$D_n = 2|C_n|$$

$$S(n\Delta f) = \frac{\frac{D_n^2}{4}}{\Delta f}$$

$$= \frac{(a_n^2 + b_n^2)/4}{\Delta f}$$

The above equations give the PSD for frequency components nΔf. The power spectrum is shown below.

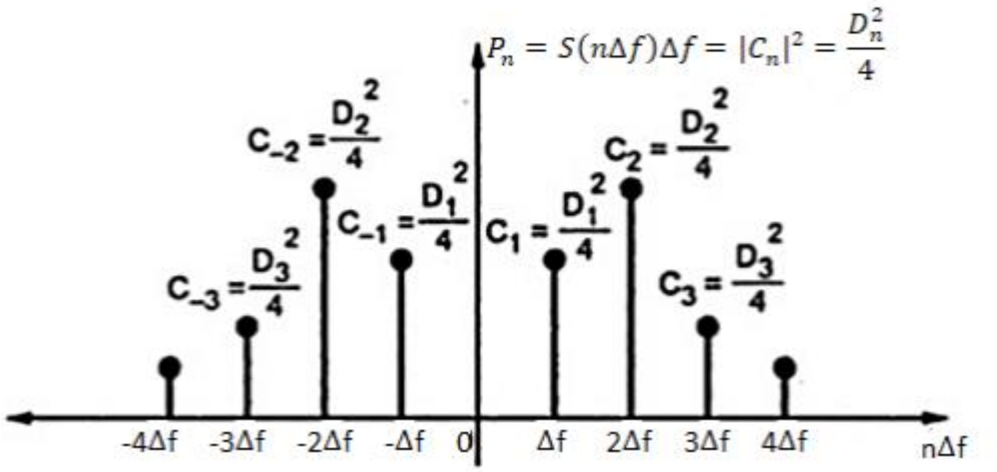


Fig: Power Spectrum

The above derivation was made by assuming noise as a periodic signal. But the fact is noise is never periodic. Now, considering T tends to infinity and Δf tends to 0. The actual noise will then be

$$n(t) = \lim_{\Delta f \to 0} \left\{ \sum_{n=1}^{\infty} a_n cos\,(2\pi n \Delta f t) + \sum_{n=1}^{\infty} b_n sin\,(2\pi n \Delta f t) \right\}$$

$$n(t) = \lim_{\Delta f \to 0} \sum_{n=1}^{\infty} D_n cos\,(2\pi n f \Delta t + \phi_n)$$

The above equation represents noise in frequency domain. The values of constants a_n and b_n are changing as noise is not constant. Hence, they are called random variables of noise, then we write $\overline{a}_n^2$ in place of a_n and $\overline{b}_n^2$ in place of a_n.

$$\overline{D_n^2} = \overline{a}_n^2 + \overline{b}_n^2$$

As Δf tends to 0 the discrete spectral lines and power spectrum gets closer and forms continuous spectrum.

$$\lim_{\Delta f \to 0} S(n\Delta f) = S(f)$$

$$S(f) = \lim_{\Delta f \to 0} \frac{\overline{D_n^2}/4}{\Delta f} = \lim_{\Delta f \to 0} \frac{(\bar{a}_n^2 + \bar{b}_n^2)/4}{\Delta f}$$

Key takeaway

$$n(t) = \lim_{\Delta f \to 0} \left\{ \sum_{n=1}^{\infty} a_n cos\,(2\pi n \Delta f t) + \sum_{n=1}^{\infty} b_n sin\,(2\pi n \Delta f t) \right\}$$

$$n(t) = \lim_{\Delta f \to 0} \sum_{n=1}^{\infty} D_n cos\,(2\pi n f \Delta t + \phi_n)$$

$$S(f) = \lim_{\Delta f \to 0} \frac{\overline{D_n^2}/4}{\Delta f} = \lim_{\Delta f \to 0} \frac{(\bar{a}_n^2 + \bar{b}_n^2)/4}{\Delta f}$$

1.11 Superposition of Noises

The equations derived in above section represents the noise as a superposition of noise spectral components. The frequency of these components is $n\Delta$f and Δf tends to 0. The n^{th} noise component can be given as

$$n_n(t) = a_n cos\,(2\pi n \Delta f t) + b_n sin\,(2\pi n \Delta f t)$$
$$n_n(t) = D_n cos\,(2\pi n \Delta f t + \phi_n)$$

The spectral components of noise are random processes. The variables a_n, b_n, D_n and ϕ_n are random variables. As $n_n(t)$ is random process the normalised power of $n_n(t)$ can be found by taking average of $[n_n(t)]^2$.

$$P_n = \overline{[n_n(t)]}^2$$

Substituting value of $n_n(t)$ and finding normalised power we get

$$P_n = \overline{a_n}^2 cos^2\,(2\pi n \Delta f t) + \overline{b_n^2} sin\,(2\pi n \Delta f t) + 2\,\overline{a_n b_n}\,sin\,(2\pi n \Delta f t) cos\,(2\pi n \Delta f t)$$

As $n_n(t)$ is stationary process value of $\overline{[n_n(t)]^2}$

At time $t=t_1$ cos $(2\pi n^{\Delta} f t_1) = 1$ and sin $(2\pi n^{\Delta} f t_1) = 0$.

$P_n = \overline{a_n}^2$

At time t=t$_2$ cos (2πn$^{\Delta}$ft$_1$) =0 and sin (2πn$^{\Delta}$ft$_1$) =1.

$P_n = \overline{b_n}^2$
$\overline{a_n}^2 = \overline{b_n}^2$

$$P_n = \overline{a_n}^2[cos^2\,(2\pi n_\Delta f t) + sin\,(2\pi n_\Delta f t)] + 2\,\overline{a_n b_n}\,sin\,(2\pi n_\Delta f t) cos\,(2\pi n_\Delta f t)]$$

$$P_n = \overline{a_n}^2 + 2\,\overline{a_n b_n}\,sin\,(2\pi n_\Delta f t) cos\,(2\pi n_\Delta f t)$$

But

$P_n = \overline{a_n}^2$
$\overline{a_n b_n} = 0$

The average of multiplication of two terms in above equation is nothing but correlation between them.

Consider two noise components $n_1(t)$ and $n_2(t)$. Then the total noise due to these components is

$n(t) = n_1(t) + n_2(t)$

This is called superposition.

The normalised power P of these components is given as

$$P = \overline{[n(t)]^2} = \overline{[n_1(t) + n_2(t)]^2}$$
$$= \overline{[n_1(t)]^2} + \overline{[n_2(t)]^2} + 2\,\overline{n_1(t)n_2(t)}$$

The term $\overline{n_1(t)n_2(t)}$ represents the cross correlation of $n_1(t)$ and $n_2(t)$. As individual spectral components of Gaussian noise are uncorrelated hence the correlation of $n_1(t)$ and $n_2(t)$ will be zero.

$$P = \overline{[n_1(t)]^2} + \overline{[n_2(t)]^2}$$
$$P = P_1 + P_2$$

1.12 Linear Filtering of Noise

Noise has wide range of frequencies. Generally, we use filters to minimize the noise power. These filters are narrow filters and they pass the frequencies of the required signals. The noise power is reduced in the output of the filter. These filters are used before the demodulators in the receivers. The filters used in noise for linear filtering are:

RC-Low Pass Filter

The transfer function of RC filter is given

$$H(f) = \frac{1}{1 + jf/f_c}$$

f_c = 3db cut-off frequency of the filters.

Considering $n_i(t)$ with PSD of $S_i(f)$ and output noise signal be $n_o(t)$ with PSD of $S_o(f)$. The output and input PSDs of noise

$$S_0(f) = |H(f)|^2 \, S_i(f)$$

If the input noise is additive white gaussian noise (AWGN), the PSDs we have,

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

The magnitude of H(f) can be obtained from

$$|H(f)|^2 = \frac{1}{1 + \left(\frac{f}{f_c}\right)^2}$$

Substituting the values, we get

$$S_0(f) = \frac{N_0}{2} \cdot \frac{1}{1 + \left(\frac{f}{f_c}\right)^2}$$

The equation gives power spectral density of output noise signal. The average power can be obtained from PSD

$$P = \int_{-\infty}^{\infty} S(f)df$$

The expression for output noise power is given as

$$P_{n0} = \int_{-\infty}^{\infty} S_0(f)df$$

$$= \int_{-\infty}^{\infty} \frac{N_0}{2} \cdot \frac{1}{1 + \left(\frac{f}{f_c}\right)^2} df$$

Let $x=f/f_c$

$$f_c\, dx = df$$

$$P_{N0} = \frac{N_0}{2} f_c \int_{-\infty}^{\infty} \frac{1}{1 + x^2} dx$$

$$= \frac{N_0}{2} f_c \cdot \pi$$

$$\int_{-\infty}^{\infty} \frac{1}{1 + x^2} dx = \pi$$

The average noise power at the output of RC filter is

$$P_{N0} = \frac{\pi N_0 f_c}{2}$$

Ideal Low Pass Filter

The transfer function of ideal LPF is given as

$$H(f) = 1 \quad for \quad -B \leq f \leq B$$
$$= 0 \qquad elsewhere$$

The input and output PSD are given as

$$S_0(f) = |H(f)|^2 \; S_i(f)$$

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

The value of average noise power at output is given as

$$P_{n0} = \int_{-\infty}^{\infty} S_0(f)df$$
$$= \int_{-\infty}^{\infty} |H(f)|^2 \frac{N_0}{2} df$$
$$= \frac{N_0}{2} \int_{-B}^{B} (1)^2 df = N_0 B$$

The above equation is the noise power at the output of ideal LPF.

1.13 Quadrature Components of Noise

The spectral component of noise is represented as

$$n(t) = \lim_{\Delta f \to 0} \sum_{k=1}^{\infty} (a_k cos\, 2\pi k \Delta f t + b_k sin\, 2\pi k \Delta f t)$$

Another way of representing the noise is shown below

$$n(t) = n_c(t) cos\, 2\pi f_0 t - n_s(t) sin\, 2\pi f_0 t$$

The above equation of noise is also called as narrowband representation of noise because there is narrow frequency band at the neighbour of frequency f_o. The quadrature component representation is also used because of the appearance in the equation of sinusoids in quadrature. Let f_o corresponds to k=k' we get

$f_o = k\Delta f$

The term $2\pi f_o t - 2\pi k \Delta f t = 0$ is added to above equation we get

$$n(t) = \lim_{\Delta f \to 0} \sum_{k=1}^{\infty} \{a_k cos\, 2\pi[f_0 + (k-K)\Delta f]t + b_k sin\, 2\pi[f_0 + (k-K)\Delta f]t\}$$

Using some trigonometric identities for cosine and sine terms we can also write above equation as

$$n_c(t) = \lim_{\Delta f \to 0} \sum_{k=1}^{\infty} \{a_k cos\, 2\pi(k-K)\Delta f t + b_k sin\, 2\pi(k-K)\Delta f t\}$$

$$n_s(t) = \lim_{\Delta f \to 0} \sum_{k=1}^{\infty} \{a_k sin\ 2\pi(k-K)\Delta f t - b_k cos\ 2\pi(k-K)\Delta f t\}$$

The variables $n_c(t)$ and $n_s(t)$ are stationary random processes represented as linear superposition of spectral components. The narrow band noise can be seen in relation to find out the quadrature component of noise. The noise spectral component in n(t) of frequency f= $k\Delta f$ gives rise in $n_c(t)$ and $n_s(t)$ to a spectral component of frequency (k-K) Δf = $f-f_o$

If we assume the noise n(t) is narrow band and extends over bandwidth B and also is selected midway in the frequency range of the noise. The noise spectrum of n(t) extends from f_o-B/2 to f_o+B/2. The spectrum of $n_c(t)$ and $n_s(t)$ extends over -B/2 to +B/2. In view of the slow variation of $n_c(t)$ and $n_s(t)$ relative to the sinusoid of frequency f_o it is useful to give the quadrature representation of noise as interpretation in terms of phasors and a phasor diagram. The term $n_c(t)\cos 2\pi f_o t$ is of frequency f_o and has varying amplitude of $n_c(t)$. The term $-n_s(t)\sin 2\pi f_o t$ is in quadrature with first term and has slow varying amplitude $n_s(t)$. In coordinate system rotating counter-clockwise with angular velocity $2\pi f_o$ with phasor diagram shown below.

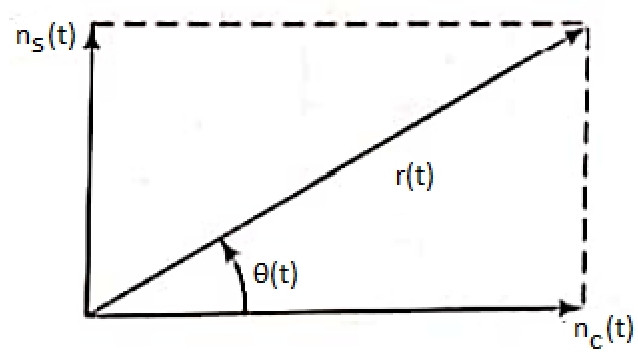


Fig: A phasor diagram of Quadrature representation of noise

The two phasors of varying amplitude give rise to third resultant phasor of amplitude $r(t) = [n_c^2(t)+n_s^2(t)]^{1/2}$

And having angle shown below.

$$\theta(t) = tan^{-1}[n_s(t)/n_c(t)]$$

Key takeaway

The quadrature representation is useful in analysis of noise and the phasor representation is useful in angle modulation communication systems.

1.14 Representation of Noise using Orthonormal Coordinates

Gram-Schmidt orthogonalization procedure

In mathematics, particularly linear algebra and numerical analysis, the Gram-Schmidt process is a method for orthogonalizing a set of vectors in an inner product space.

We know that any signal vector can be represented in terms of orthogonal basis functions $\phi_1(t), \phi_2(t), \ldots \phi_N(t)$. Gram-schmidt orthogonalization procedure is the tool to obtain the orthonormal basis function $\phi_i(t)$

To derive an expression for $\phi_1(t)$

Suppose we have set of 'M' energy signals denoted by $S_1(t), S_2(t), \ldots, S_M(t)$

Starting with $S_1(t)$ chosen from set arbitrarily the basis function is defined by

$$\phi_1(t) = \frac{S_1(t)}{\sqrt{E_1}}$$

Where E_1 is the energy of the signal $S_1(t)$.

From question (1), we can write

$$S_1(t) = \sqrt{E_1}\,\phi_1(t) \qquad ---(2)$$

We know that $\sum_{j=1}^{N} S_{ij}\,\phi_j(t)$ for N=1 eq (2) can be written as

$$S_1(t) = S_{11}(t)\phi_1(t) \qquad ---(3)$$

From the above equation (3) we obtain $S_{11} = \sqrt{E_1}$ and $\phi_1(t)$ has unit energy.

Next, using the signal $S_2(t)$ we define the co-efficient $S_{21} = \int_0^T S_2(t)\phi_1(t)dt \quad ---(4)$

Let $g_2(t)$ be a new intermediate function which is given as $g_2(t) = S_2(t) - S_{21}\,\phi_1(t) \quad (5)$

The function is orthogonal to $\phi_1(t)$ over the interval 0 to T.

The second function, which is given as

$$\phi_2(t) = \frac{g_2(t)}{\sqrt{E_{g2}}} \qquad \rightarrow (6)$$

$E_{g2} = \int_0^T g_2^2(t)dt$ is the energy of $g_2(t)$

$$\phi_2(t) = \frac{g_2(t)}{\sqrt{\int_0^T g_2^2(t)dt}} \qquad \rightarrow (7)$$

a) To prove that $\phi_2(t)$ has unit energy

Energy of $\phi_2(t)$ will be $\int_0^T \phi_2^2(t)dt = \int_0^T \left[\frac{g_2(t)}{\sqrt{E_{g2}}}\right] dt = \frac{1}{E_{g2}} \int_0^T g_2^2(t)dt$

$$= \frac{1}{E_{g2}} \int_0^1 g_2^2(t)dt$$

We know that $E_{g2} = \int_0^T g_2^2(t)dt$

$$= \frac{1}{E_{g2}} \times E_{g2}$$

$$\int_0^T \phi_2^2(t)dt = 1$$

b) To prove that $\phi_1(t)$ and $\phi_2(t)$ are orthogonal

Consider $\int_0^T \phi_1(t)\phi_2(t)\, dt$

Substitute the values of $\phi_1(t)$ and $\phi_2(t)$ in the above equation. i.e

$\phi_1(t) = \frac{s_1(t)}{\sqrt{E_1}}$ and $\phi_2(t) = \frac{g_2(t)}{\sqrt{E_{g2}}}$

$$= \int_0^T \frac{S_1(t) g_2(t)}{\sqrt{E_1}\ \sqrt{E_{g2}}} dt$$

$$\int_0^T \phi_1(t)\ \phi_2(t)dt = \frac{1}{\sqrt{E_1}\sqrt{E_{g2}}} \int_0^T s_1(t)\ g_2(t)dt$$

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

$$= \frac{1}{\sqrt{E_1}\sqrt{E_{g2}}} \int_0^T s_1(t)\,[s_2(t) - s_{21}\phi_1(t)]dt$$
$$= \frac{1}{\sqrt{E_1}\sqrt{E_{g2}}} \left[\int_0^T s_1(t)\,s_2(t)dt - \int_0^T s_1(t)\,s_{21}(t)\phi_1(t)dt \right]$$

We know that

$$s_{21} = \int_0^T s_2(t)\phi_2(t)$$
$$\int_0^T \phi_1(t)\,\phi_2(t)dt = \frac{1}{\sqrt{E_1}\sqrt{E_{g2}}} \left[\int_0^T s_1(t)\,s_2(t)dt - \int_0^T \int_0^T s_2(t)\,\phi_1(t) s_1(t)\phi_1(t)dt \right]$$

From the given equation there is a product of two terms $s_1(t)$ and $s_2(t)$. But the two symbols are not present at a time. Hence the product of $s_1(t)$ and $s_2(t)$. Hence the product of $s_1(t)$ and $s_2(t)$ i.e., $s_1(t), s_2(t) = 0$ and hence the integration terms in RHS will be zero. Ie

$\int_0^T \phi_1(t)\,\phi_2(t)dt = 0$. Thus, the noo basis function are orthonormal.

Generalized equation for orthonormal basis functions

The generalised equation for orthonormal basis function can be written by considering the following equation i.e.

$$\phi_i(t) = \frac{g_i(t)}{\sqrt{E_{gi}}}, \qquad i = 1,2,3,\ldots,N$$

Where $g_i(x)$ is given by the generalized equation $g_i(t) = s_i(t) - \sum_{j=1}^{i-1} s_{ij}\phi_j(t)$

Note

$(i-1) \rightarrow s_1(t)$ are ready taken consideration.

Where the coefficients

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

For $i = 1$ the $g_i(t)$ redues to $s_i(t)$

Given the $g_i(t)$, we may define the set of ban's function

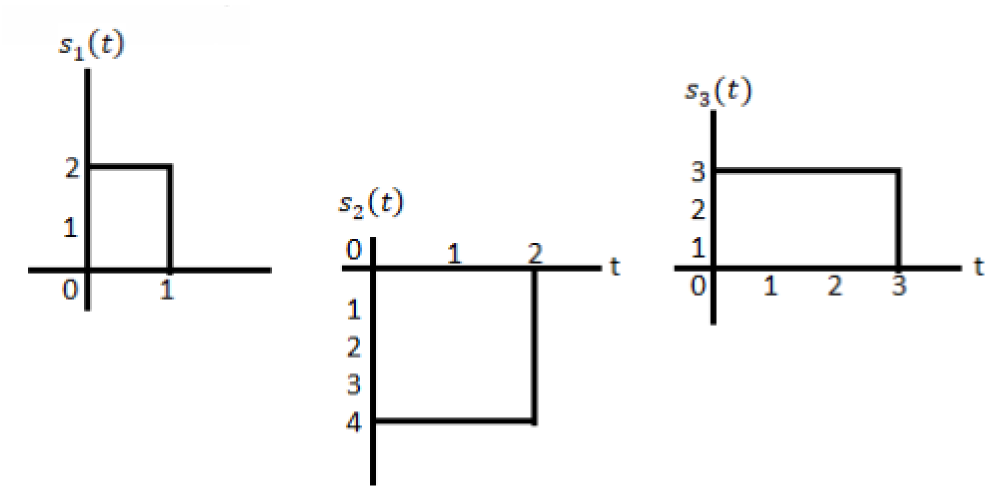
$$\phi_i(t) = \frac{g_i(t)}{\sqrt{\int_0^T g_2^2(t)dt}}$$

Which form an orthogonal set. The dimension 'N' is less than or equal to the number of given signals, M depending on one of N_0 possibilities.

- The signals $s_1(t), s_2(t), \ldots s_m(t)$ form a linearly independent set, in which case N=M.
- The signals $s_1(t), s_2(t), \ldots s_m(t)$ are not linearly independent.

Problem-1

Using the fram-schmidt orthogonalization procedure, find a set of orthonormal ban's functions represent the tree signals $s_1(t), s_2(t)$ & $s_3(t)$ as shown in figure.



Solution

All the three signals $s_1(t), s_2(t)$ & $s_3(t)$ are not linear combination of each other hence they are linearly independent. Hence, we require three ban's function.

To obtain $\phi_1(t)$

Energy of $s_1(t)$ $\quad E_1 = \int_0^T s_1^2(t)dt$

$$E_1 = \int_0^1 2^2 dt = 4[1-0]$$

$$E_1 = 4$$

$$\phi_1(t) = \frac{s_1(t)}{\sqrt{E_1}} = \frac{2}{\sqrt{4}} = \frac{2}{2} = 1$$

$$\phi_1(t) = \begin{cases} 1 & for\ 0 \le t \le 1 \\ 0 & otherwise \end{cases}$$

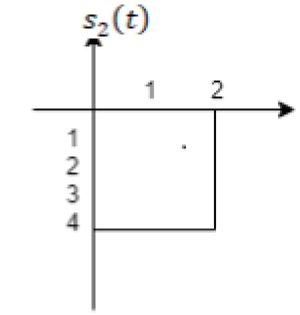
To obtain $\phi_2(t)$

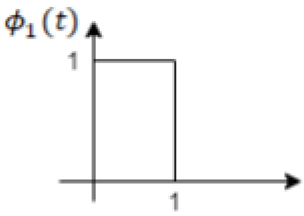
$$g_2(t) = s_2(t) - s_{21}\phi_1(t)$$

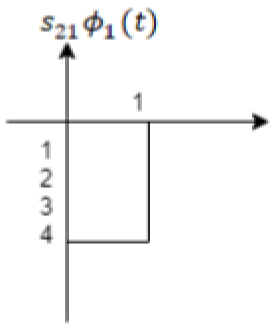
$$s_{21} = \int_0^T s_2(t)\phi_1(t)dt$$

$$s_{21} = \int_0^1 (-4)(1)dt$$

$$s_{21} = -4 \quad for\ \ 0 \le t \le 1$$







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

$$s_{21}\phi_1(t) = \begin{cases} -4 & for\ 0 \le t \le 1 \\ 0 & otherwise \end{cases}$$

$$g_2(t) = \begin{cases} -4 & for\ 1 \le t \le 2 \\ 0 & otherwise \end{cases}$$

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

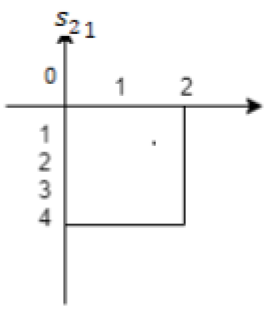
$$E_{g2} = \int_0^T g_2^2(t)dt$$

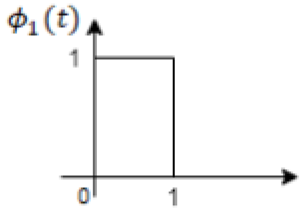
$$E_{g2} = \int_1^2 (-4)^2 dt$$

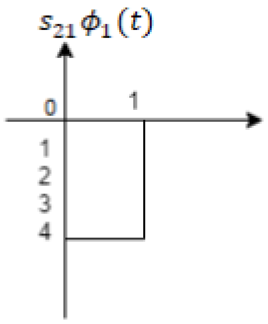
$$E_{g2} = 16(2-1)$$

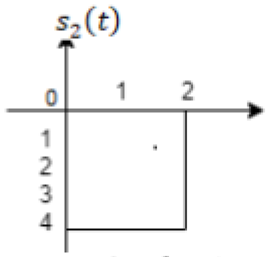
$$E_{g2} = 16$$

$$\phi_2(t) = \frac{g_2(t)}{\sqrt{E_{g2}}} = -\frac{4}{\sqrt{16}} = -\frac{4}{4}$$









$$s_{32}\phi_2(t) = \begin{cases} 3 & for\ 1 \le t \le 2 \\ 0 & otherwise \end{cases}$$

To obtain $\phi_3(t)$

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

$$g_3(t) = s_3(t) - s_{31}(t)\phi_1(t) - s_{32}(t)\phi_2(t)$$

$$s_{31}(t) = \int_0^T s_3(t)\phi_1(t)\, dt$$

$$s_3(t)\phi_1(t) = \begin{cases} 3 & for\ 0 \le t \le 1 \\ 0 & otherwise \end{cases}$$

$$s_{31}(t) = \int_0^T (3)(1)\, dt$$

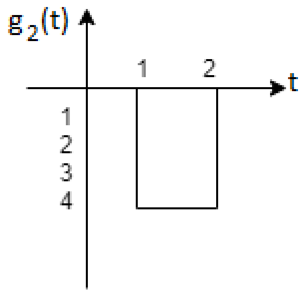
$$s_3(t) = \begin{cases} 3 & for\ 0 \le t \le 1 \\ 0 & otherwise \end{cases}$$

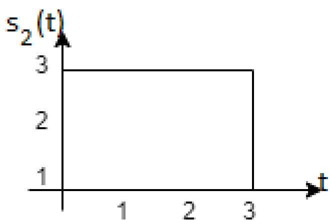
$$s_{32} = \int_1^2 (3)(-1)\, dt$$

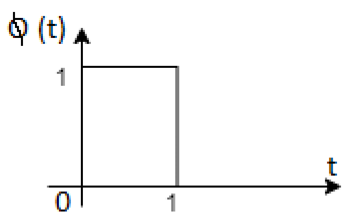
$$s_{32} = -3[2-1]$$

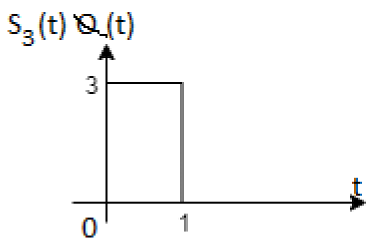
$$s_{32} = -3[1]$$

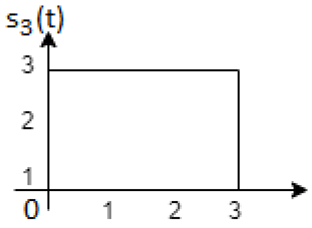
$$s_{32} = \begin{cases} 3 & for\ 1 \le t \le 2 \\ 0 & otherwise \end{cases}$$

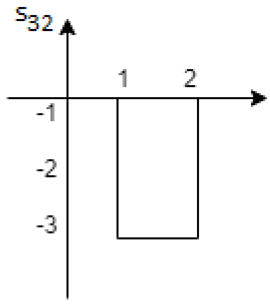


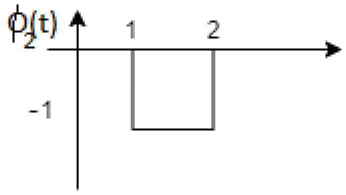


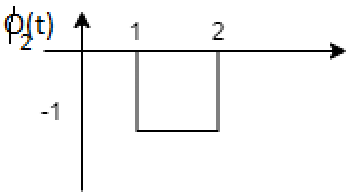


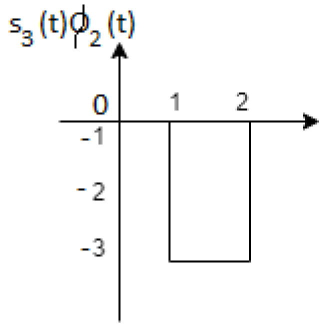


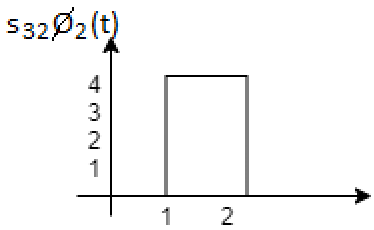


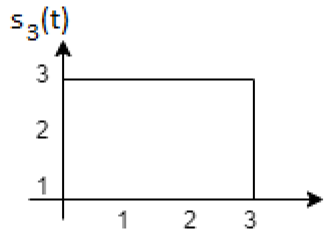












$$g_3(t) = [s_3(t) - s_{31}(t)\phi_1(t)] - s_{32}(t)\phi_2(t)$$

$$g_3(t) = \begin{cases} 3 & for\ 2 \leq t \leq 3 \\ 0 & otherwise \end{cases}$$

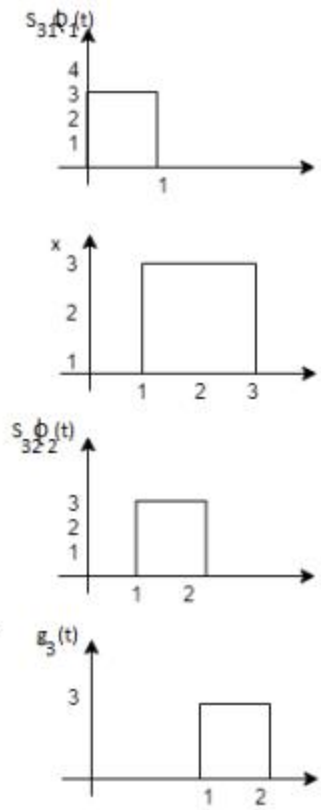
$$E_{g3} = \int_0^T g_3^2(t)dt$$

$$E_{g3} = \int_2^3 (3)^2 dt$$

$$E_{g3} = 9(3-2) = 9$$

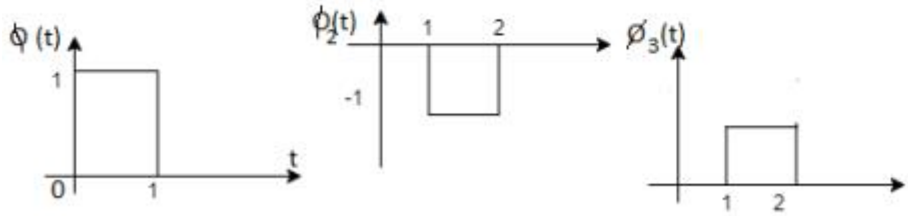
$$\phi_3(t) = \frac{g_3(t)}{\sqrt{E_{g3}}}$$

$$\phi_3(t) = \begin{cases} 1 & for\ 2 \leq t \leq 3 \\ 0 & otherwise \end{cases}$$



$$\emptyset_3(t) = \begin{cases} 1 & 2 \le r \le 3 \\ 0 & otherwise \end{cases}$$

Ban's functions as shown below



Problem-2

Consider the signals $s_1(t), s_2(t), s_3(t)$ and $s_4(t)$ as given below. Find an orthonormal basis for phase set of signals using Grammar schimdt orthogonalisation procedure.

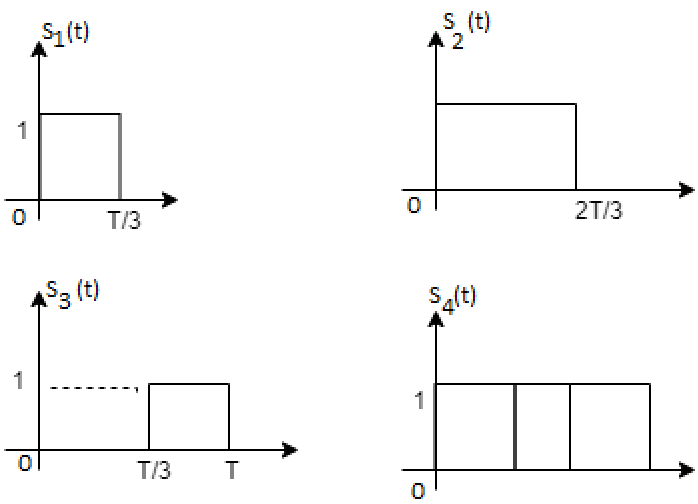


Fig. Sketch of $s_1(t), s_2(t), s_3(t)$ and $s_4(t)$

Solution:

From the above figures $s_4(t) = s_1(t) + s_3(t)$. This means all four signals are not linearly independent. Gram-schmidt orthogonalisation procedure is carried out for a subset which is linearly independent. Here $s_1(t), s_2(t), s_3(t)$ are linearly independent. Hence, we will determine orthonormal.

To obtain $\phi_1(t)$

Energy of $s_1(t)$ is $E_1 = \int_0^T s_1^2(t)dt$

$$= \int_0^{T/3} (1)^2 dt = [T]_0^3 = \left[\frac{T}{3} - 0\right]$$

$$E_1 = \frac{T}{3}$$

We know that

$$\phi_1(t) = \frac{s_1(t)}{\sqrt{E_1}} = \begin{cases} \frac{1}{\sqrt{\frac{T}{3}}} & for\ 0 \leq t \leq \frac{T}{3} \\ 0 & otherwise \end{cases}$$

$$\phi_1(t) = \begin{cases} \sqrt{3/T} & for\ 0 \leq t \leq T/3 \\ 0 & otherwise \end{cases}$$

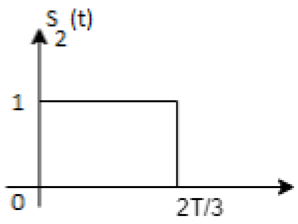
To obtain $\phi_2(t)$

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

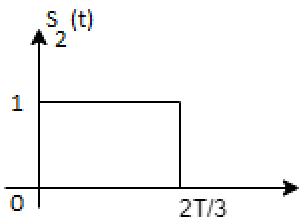
$$s_{21} = \int_0^{T/3} (1)\sqrt{3/T}dt = \left(\sqrt{\frac{3}{T}}\right) \times \left(\frac{T}{3}\right) = \frac{\sqrt{T}}{\sqrt{3}}$$

$$s_{21} = \frac{\sqrt{T}}{\sqrt{3}}$$

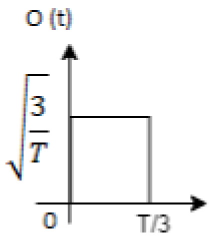
$$s_{21}\,\phi_1(t) = \begin{cases} \sqrt{\frac{3}{T}} \times \frac{\sqrt{T}}{\sqrt{3}} & for\ 0 \leq t \leq T/3 \\ 0 & otherwise \end{cases}$$



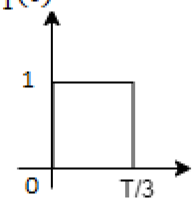
$$s_{21}\,\phi_1(t) = \begin{cases} 1 & for\ 0 \le t \le T/3 \\ 0 & otherwise \end{cases}$$



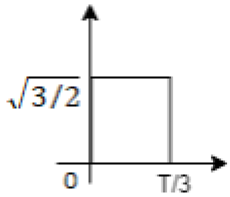
From the above figure the intermediate function can be defined as



$s_{21}\ \phi_1(t)$

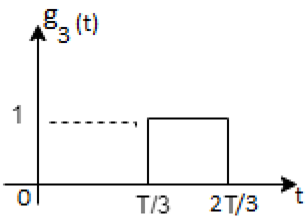


$s_2(t)\phi_1(t)$



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

$$= \begin{cases} 0 & for\ 0 \le t \le \frac{T}{3} \\ 1 & for \frac{T}{3} \le t \le \frac{2T}{3} \\ 0 & for\ \ t > \frac{2T}{3} \end{cases}$$



Energy of $g_2(t)$ will be

$$E_{g2} = \int_0^T g_2^2(t)dt = \int_{T/3}^{2T/3} (1)^2 dt = 2\frac{T}{3} - \frac{T}{3}$$

$$E_{g2} = \frac{T}{3}$$

Now

$$\phi_2(t) = \frac{g_2(t)}{\sqrt{E_{g2}}} = \frac{1}{\sqrt{T/3}} = \sqrt{\frac{3}{T}}$$

$$\phi_2(t) = \begin{cases} \sqrt{\frac{3}{T}} & for \frac{T}{3} \le t \le \frac{2T}{3} \\ 0 & otherwise \end{cases}$$

To obtain $\phi_3(t)$

We know that the generalized equation for gramm-scmidt procedure

$$g_i(t) = s_i(t) - \sum_{j=1}^{i-1} s_{ij}\phi_j(t) \quad i = 1,2,\ldots N$$

With N=3

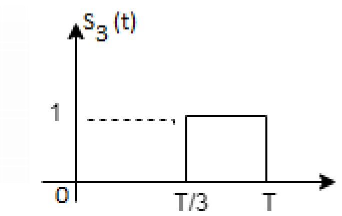
$$g_3(t) = s_3(t) - \sum_{j=1}^{3} s_{3j}\phi_j(t)$$
$$= s_3(t) - [s_{31}(t)\phi_1(t) + s_{32}(t)\phi_2(t)]$$
$$g_3(t) = s_3(t) - [s_{31}(t)\phi_1(t) + s_{32}(t)\phi_2(t)]$$

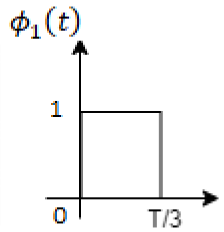
We know that

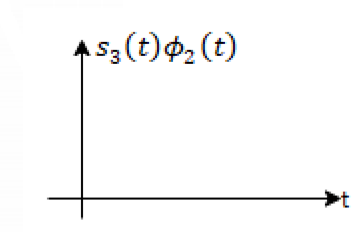
$$s_{ij} = \int_0^T s_i(t)\phi_j(t)\, dt$$
$$s_{31} = \int_0^T s_3(t)\phi_1(t)\, dt = 0$$

Since, there is no overlap between $s_3(t)$ and $\phi_1(t)$

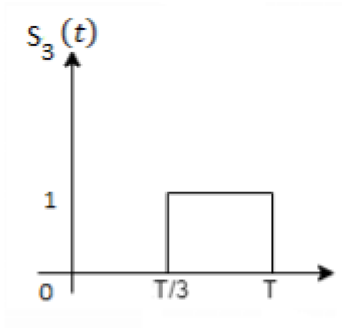
$$s_{32} = \int_0^T s_3(t)\phi_2(t)\, dt$$







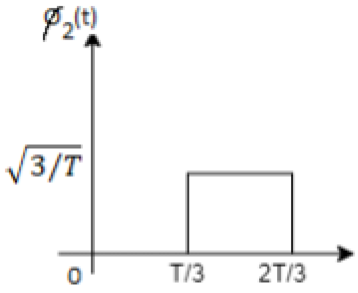
$$s_{32} = \int_{T/3}^{2T/3} s_3(t)\phi_2(t)\, dt$$
$$s_{32} = \int_{T/3}^{2T/3} (1)(\sqrt{3/T})\, dt$$
$$= \left[\sqrt{\frac{3}{T}} \times t\right]_{\frac{T}{3}}^{\frac{2T}{3}}$$
$$= \sqrt{\frac{3}{T}}\left[\frac{2T}{3} - \frac{T}{3}\right]$$
$$= \sqrt{\frac{3}{T}} \times \frac{T}{3}$$
$$= \sqrt{\frac{3}{T}}\sqrt{\frac{T}{3}}\sqrt{\frac{T}{3}}$$
$$s_{32} = \sqrt{\frac{T}{3}}$$



Here $s_{31}(t)\phi_1(t) = 0 \quad s_{31} = 0$

$$s_{32}(t)\phi_2(t) = \begin{cases} \sqrt{\frac{3}{T}} \times \sqrt{\frac{T}{3}} & for \frac{T}{3} \le t \le \frac{2T}{3} \\ 0 & otherwise \end{cases}$$

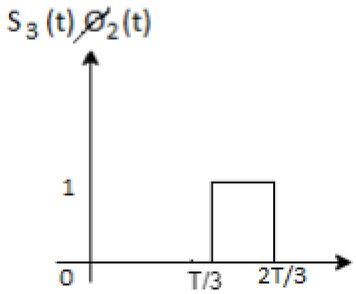
$$= \begin{cases} 1 & for \frac{T}{3} \le t \le \frac{2T}{3} \\ 0 & otherwise \end{cases}$$



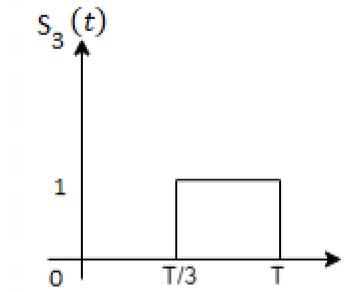
We know that

$$g_3(t) = [s_3(t) - s_{31}(t)\phi_1(t)] - s_{32}(t)\phi_2(t)$$
$$g_3(t) = s_3(t) - s_{32}(t)\phi_2(t)$$

$$g_3(t) = \begin{cases} 0 & for\ 0 \le t \le \frac{2T}{3} \\ 1 & for \frac{2T}{3} \le t \le T \\ 0 & for \quad t > T \end{cases}$$



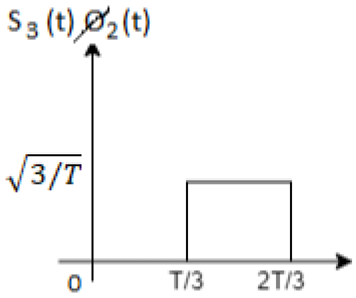
$$g_3(t) = \begin{cases} 1 & for \frac{2T}{3} \le t \le T \\ 0 & otherwise \end{cases}$$



$$E_{g3} = \int_0^T g_3^2(t)dt = \int_{T/3}^{2T/3} 2^2\, dt = [t]_{T/3}^{2T/3} = \frac{2T}{3} - \frac{T}{3} = \frac{T}{3}$$

$$E_{g3} = \frac{T}{3}$$

$$\phi_3(t) = \frac{g_3(t)}{\sqrt{E_{g3}}}$$



$$\phi_3(t) = \begin{cases} 1/\sqrt{T/3} & for \frac{2T}{3} \leq t \leq T \\ 0 & otherwise \end{cases}$$

$$\phi_3(t) = \begin{cases} \sqrt{3/T} & for \frac{2T}{3} \leq t \leq T \\ 0 & otherwise \end{cases}$$

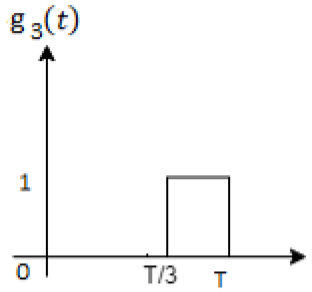
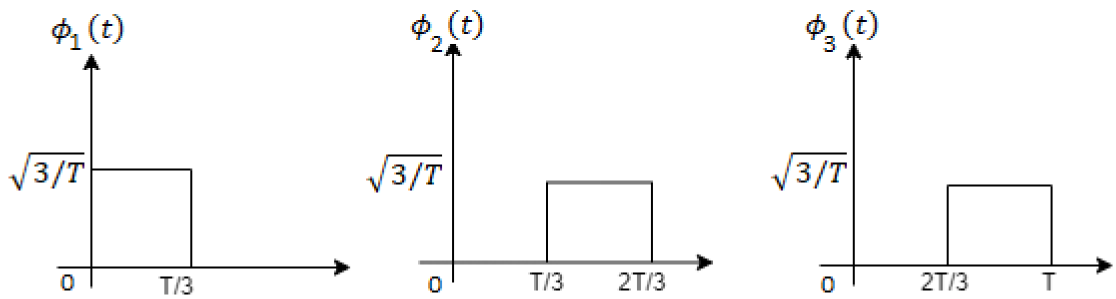
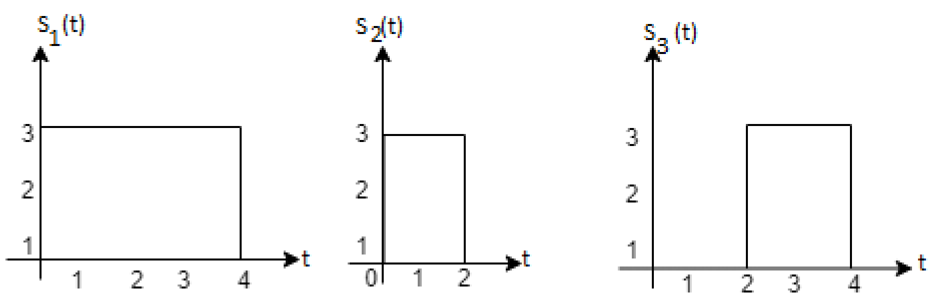


Figure below shows orthonormal basis function

Problem 3:



Three signals $s_1(t), s_2(t)\ \&\ s_3(t)$ are shown in fig. Apply Gram-schmidt procedure to obtain an orthonormal basis for the signals. Express signals $s_1(t), s_2(t)\ \&\ s_3(t)$ in terms of orthonormal basis function.



Solution

i) To obtain orthonormal basis function

Here $s_3(t) = s_1(t) - s_2(t)$

Hence, we will obtain basis solution for $s_1(t)$ and $s_2(t)$ only.

To obtain $\phi_1(t)$

Energy of $s_1(t)$ is $E_1 = \int_0^T s_1^2(t)dt$

$$= \int_0^4 (3)^2 dt = [9t]_0^4 = 9[4-0] = 36$$

$$E_1 = 36$$

We know that

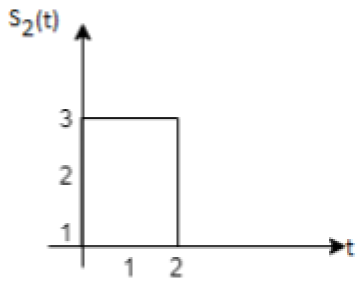
$$\phi_1(t) = \frac{s_1(t)}{\sqrt{E_1}} = \frac{3}{\sqrt{36}} = \frac{3}{6} = \frac{1}{2} = \begin{cases} 1/2 & for\ 0 \le t \le 4 \\ 0 & otherwise \end{cases}$$

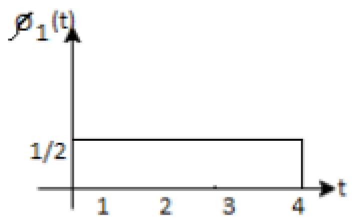
To obtain $\phi_2(t)$

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

$$s_{21} = \int_0^2 3 \times \frac{1}{2} dt = \frac{3}{2}[2-0] = 3$$

$$s_{21}\ \phi_1(t) = \begin{cases} 3 & for\ 0 \le t \le 2 \\ 0 & otherwise \end{cases}$$





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

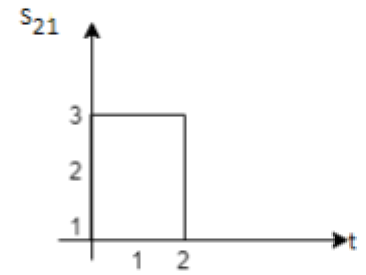
$$s_{21}\phi_1(t) = \begin{cases} 3/2 & for\ 0 \le t \le 2 \\ 0 & otherwise \end{cases}$$

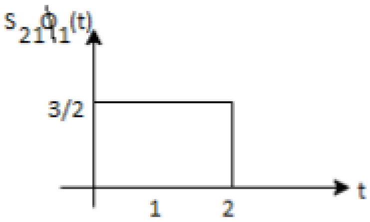
$$g_2(t) = \begin{cases} 3/2 & for\ 0 \le t \le 2 \\ 0 & otherwise \end{cases}$$

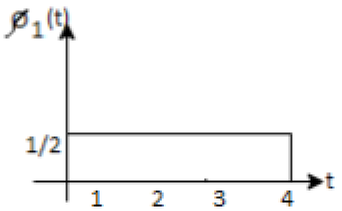
$$E_{g2} = \int_0^T g_2^2(t)dt$$

$$E_{g2} = \int_0^2 \left(\frac{3}{2}\right)^2 dt$$

$$E_{g2} = \frac{9}{4}\int_0^2 dt = \frac{9}{4}[2-0] = \frac{9}{2}$$







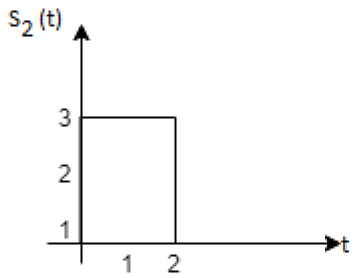
$$\mathbf{E_{g2}} = \begin{cases} \frac{9}{2} & 0 \le t \le 2 \end{cases}$$

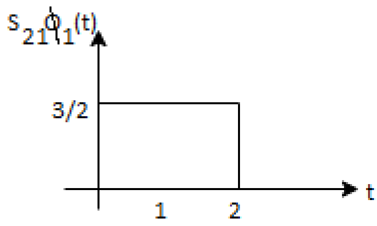
$$\phi_2(t) = \frac{g_2(t)}{\sqrt{E_{g2}}}$$

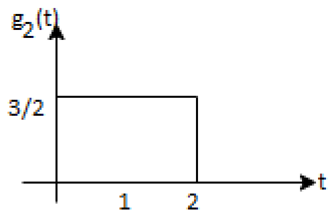
$$\phi_2(t) = \frac{\frac{3}{2}}{\sqrt{\frac{9}{2}}}$$

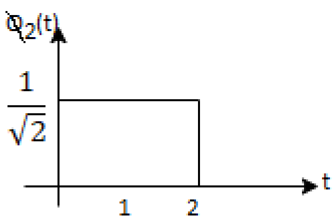
$$= \frac{\frac{3}{2}}{3/\sqrt{2}} = \frac{3}{2} \times \frac{\sqrt{2}}{3} = \frac{1}{\sqrt{2}}$$

$$\phi_2(t) = \begin{cases} \frac{1}{\sqrt{2}} & 0 \le t \le 2 \\ 0 & \text{otherwise} \end{cases}$$









References:

1. Bernard Sklar, Prabitra Kumar Ray, "Digital Communications Fundamentals and Applications", Pearson Education, 2nd Edition

2. Wayne Tomasi, "Electronic Communications System", Pearson Education, 5th Edition

3. A.B Carlson, P B Crully, J C Rutledge, "Communication Systems", Tata McGraw Hill Publication, 5th Edition

4. Simon Haykin, "Communication Systems", John Wiley & Sons, 4th Edition

5. Simon Haykin, "Digital Communication Systems", John Wiley & Sons, 4th Edition.

DC

Unit - 2

Digital Modulation-I

2.1 Baseband Signal Receiver: Probability of Error

Probability of Error defines average probability of error that can occur in a communication system.

Error Functions

(1) Error function erf(u):

$$\mathrm{erfc(u)} = \frac{2}{\sqrt{\pi}} \int_0^u \exp\left(-z^2\right) dz$$

(2) Complementary error function erfc(u):

$$\mathrm{erfc(u)} = \frac{2}{\sqrt{\pi}} \int_0^\infty \exp\left(-z^2\right) dz$$

Properties of Error function

1. Erf(-u) = -erf(u) - Symmetry.

2. Erf(u) approaches unity as u tends towards infinity.

$$\frac{2}{\sqrt{\pi}}\int_{0}^{\infty}\exp\left(-z^2\right)dz = 1$$

3. For a Random variable X, with mean mx and variance σx2, the probability of X is defined by

$$P(m_x - a < X \leq m_X + a) = \operatorname{erf}\left(\frac{a}{\sqrt{2\sigma_X}}\right)$$

Note: Relation: erfc(u) = 1 – erf(u) Tables are used to find these values.

Approximate value

$$\operatorname{erfc}(u) < \frac{\exp\left(-u^2\right)}{\sqrt{\pi}}$$

Q – Function:

An alternate form of error function. It basically defines the area under the Standardized Gaussian tail. For a standardized Gaussian random variable X of zero mean and unit variance, the Q-function is defined by

$$Q(v) = \frac{1}{\sqrt{2\pi}}\int_{0}^{\infty}\exp\left(-\frac{x^2}{2}\right)dx$$

Relations between Q-function and erfc function:

i) $Q(v) = \frac{1}{2}\operatorname{erfc}\left(\frac{v}{\sqrt{2}}\right)$

ii) $\operatorname{erfc}(u) = 2Q(\sqrt{2}u)$

2.2 Optimal Receiver Design

An optimum filter is such a filter used for acquiring a best estimate of desired signal from noisy measurement. It is different from the classic filters. These filters are optimum because they are designed based on optimization theory to minimize the mean square error between a processed signal and a desired signal, or equivalently provides the best estimation of a desired signal from a measured noisy signal.

It is pervasive that when we measure a (desired) signal d(n), noise v(n) interferes with the signal so that a measured signal becomes a noisy signal

x(n) x(n)=d(n)+v(n)

It is also very common that a signal d(n) is distorted in its measurement (e.g., an electromagnetic signal distorts as it propagates over a radio channel). Assuming that the system causing distortion is characterized by an impulse response of h (n) s, the measurement of d(n) can be expressed by the sum of distorted signal s(n) and noise

v(n) x(n)=s(n)+v(n)= h (n)∗ s d(n)+v(n)

Where s(n)= h (n)∗ s d(n).

If both d(n) and v(n) are assumed to be wide-sense stationary (WSS) random processes, then x(n) is also a WSS process. The signals that we discuss in this chapter will be WSS if they are not specially specified. If signal d(n) and measurement noise v(n) are assumed to be uncorrelated (this is true in many practical cases), then r (k) = r (k) = 0.

In this case, the noisy signal,

x(n)= h (n)∗ s d(n)+v(n),

The relation of r (k) x with r (k) d and r (k) v (the autocorrelations of x(n), d(n) and v(n), respectively) as follows,

$$r_x(k) = E\{x(n+k)x^*(n)\} = r_s(k) + r_v(k) = h_s(k) * h_2^*(-k) * r_d(k) + r_v(k)$$
$$r_{ds}(k) = E\{d(n+k_x^*(n)\} = E\{d(n+k)[s(n)+v(n)]^*\} = r_{ds}(k) = h_s^*(-k) * r_d(k)$$

For the noisy signal of the form x(n)= d(n)+v(n), a special case of where h (n) s =δ (n) and no distortion happens to d(n) in its measurement, we have

$$r_x(k) = r_d(k) + r_v(k)$$
$$r_{dx}(k) = r_d(k)$$
$$P_x(z) = H_x(z)H_x^*\left(\frac{1}{z^*}\right)P_d(z) + P_v(z)$$
$$P_{dx}(z) = H_x^*\left(\frac{1}{z^*}\right)P_d(z)$$
$$P_x(z) = P_d(z) + P_v(z\_$$
$$P_{dx}(z) = P_d(z)$$

Optimum filtering is to acquire the best linear estimate of a desired signal from a measurement. The main issues in optimal filtering contain

• Filtering that deals with recovering a desired signal d(n) from a noisy signal (or measurement) x(n);

• Prediction that is concerned with predicting a signal d(n+m) for m>0 from observation x(n);

• Smoothing that is an a posteriori form of estimation, i.e., estimating d(n+m) for m

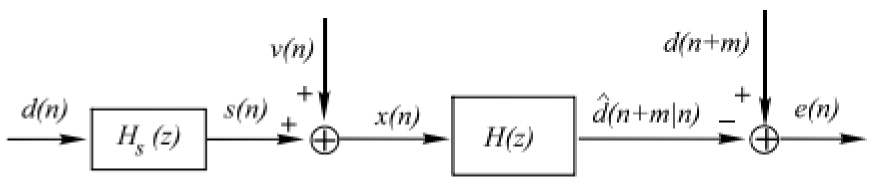


Fig 1 Optimum Filtering

Key takeaway

An **optimum filter** is such a filter used for acquiring a best estimate of desired signal from noisy measurement. It is different from the classic filters.

2.3 Digital Modulation: Generation, Reception, Signal Space Representation and Probability of Error Calculation for Binary Phase Shift Keying (BPSK)

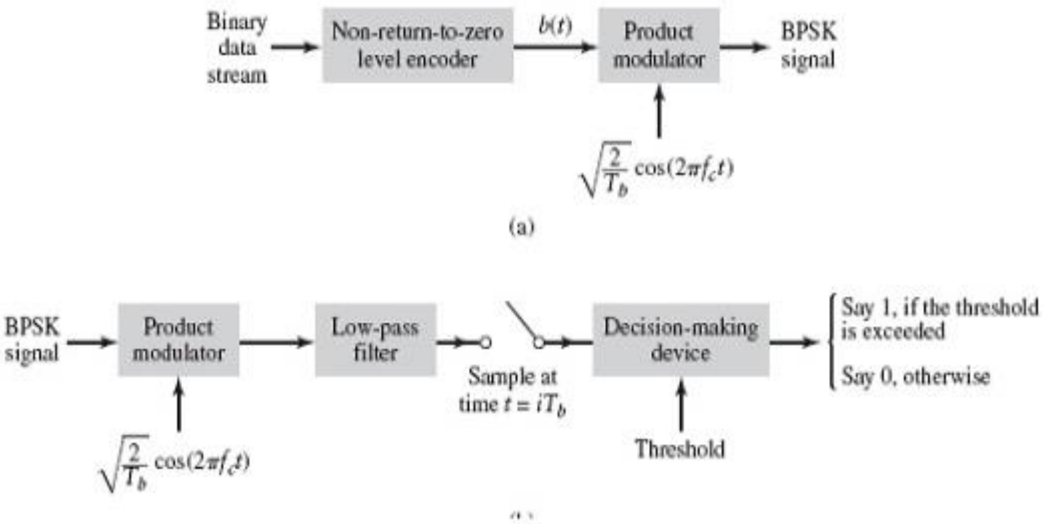
GENERATION AND COHERENT DETECTION OF BPSK SIGNALS

(i) Generation

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

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

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



$$r_A(t) = \left[s(t)+w(t)\right].\sqrt{\frac{2}{T_b}}.\cos\left(w_c t+\theta\right)$$

Fig. 2: Generation of BPSK Signal

(ii) Detection

To detect the original binary sequence of 1s and 0s, the BPSK signal at the channel output is applied to a receiver that consists of four sections

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

(b) Low-pass filter, designed to remove the double-frequency components of the product modulator output (i.e., the components centered on) and pass the zero-frequency components.

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

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

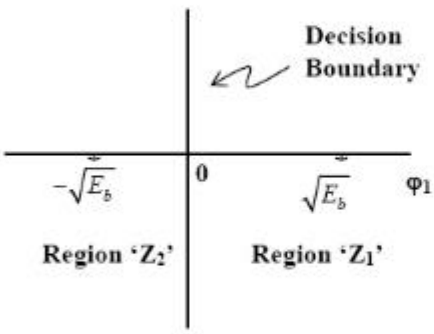


Fig. 3: Decision-making device

The signal at (B) is

$$r_1 = \sqrt{2/T_b}\int_0^{T_2}\left[d(t)\sqrt{\frac{2E_b}{T_b}}\cos(w_c t+\theta) + w(t)\right]\cos(w_c t+\theta)dt$$

$$= \sqrt{E_b}d(t) + \sqrt{\frac{2}{T_b}}\int_0^{T_2} w(t)\cos(w_c t+\theta)dt$$

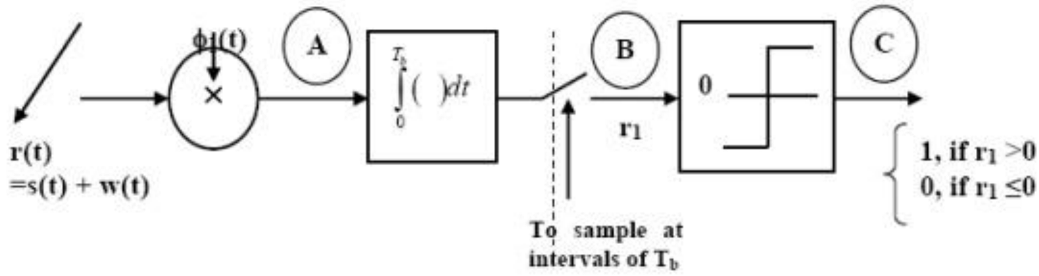


Fig.4: Decision-making device

Error Calculation for BPSK

In BPSK system the basic function is given by

$$\phi_1(t) = \sqrt{\frac{2}{T}} \cos 2\pi f_C t \qquad 0 \le t \le T_b$$

The signals S1(t) and S2(t) are given by

$S_1(t) = \sqrt{E_b}\phi_1(t) \quad 0 \le t \le T_b$ for symbol 1

$S_2(t) = -\sqrt{E_b}\phi_1(t) \quad 0 \le t \le T_b$ for symbol 0

The signal space representation is as shown in fig (N=1 & M=2)

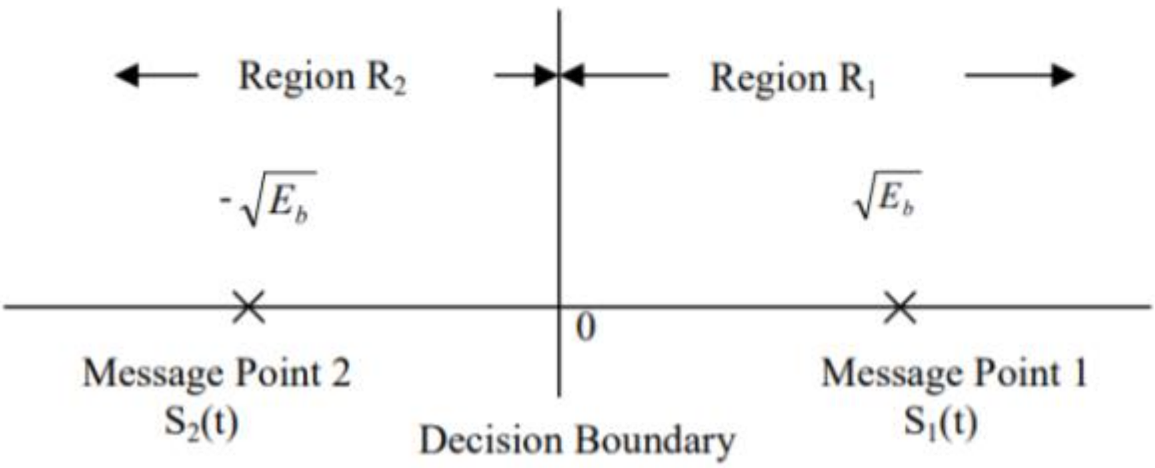


Fig.5 Signal Space Representation of BPSK

The observation vector x1 is related to the received signal x(t) by

$$x_1 \int_0^T x(t)\phi_1(t)dt$$

If the observation element falls in the region R1, a decision will be made in favour of symbol '1'. If it falls in region R2 a decision will be made in favour of symbol '0'. The error is of two types

1) Pe (0/1) i.e., transmitted as '1' but received as '0' and

2) Pe (1/0) i.e., transmitted as '0' but received as '1'. Error of 1st kind is given by

Therefore, the above equation becomes

$$P_e\left(\frac{1}{0}\right) = \frac{1}{\sqrt{2\pi\sigma^2}}\int_0^\infty exp\left[\frac{(x_! - \mu)^2}{2\sigma^2}\right] dx_1$$

Where μ=mean value = -$\sqrt{E_b}$ for the transmission of symbol '0'

$\sigma^2 = Variance = \frac{N_0}{2}$ for additive white Gaussiance noise.

Threshold value λ=0. [Indicates lower limit in integration]

$$P_{e0} = P_e\left(\frac{1}{0}\right) = \frac{1}{\sqrt{\pi N_0}}\int_0^\infty expp\left[-\frac{(x_1 + \sqrt{E_b})^2}{N_0}\right]dx_1$$

Put $Z = \frac{(x_1 + \sqrt{E_b})}{N_0}$

$$P_{e0} = P_e\left(\frac{1}{0}\right) = \frac{1}{\sqrt{\pi}}\int_0^\infty exp\,[(-Z)^2]dz$$

$$P_e\left(\frac{1}{0}\right) = \frac{1}{2}erfc\sqrt{E_b/N_0}$$

$$P_e\left(\frac{0}{1}\right) = \frac{1}{2}erfc\sqrt{\frac{E_b}{N_0}}$$

$$P_e = \frac{1}{2}\left[P_e\left(\frac{1}{0}\right) + P_e\left(\frac{0}{1}\right)\right]$$

The total probability of error Pe = Pe (1/ 0) Pe (0) + Pe (0 /1) Pe (1) assuming probability of 1's and 0's are equal.

$$P_e = \frac{1}{2} erfc \sqrt{\frac{E_b}{N_0}}$$

2.4 Binary Frequency Shift Keying (BFSK)

GENERATION AND COHERENT DETECTION OF BFSK SIGNALS

(i) Generation

On-off level encoder:

Here, the output is of constant amplitude √Eb for input 1 and 0 for input 0.

Pair of oscillators:

Frequency f1 and f2 differ by integer multiple of 1/Tb. The lower oscillator has frequency f2 preceded by inverter. When in a signal interval, the input symbol is 1, the upper oscillator is switched on, and signal $s_1(t)$ is transmitted, while lower oscillator is switched off.

When input is 0, upper oscillator is off, lower oscillator is on and signal $s_2(t)$ is transmitted.

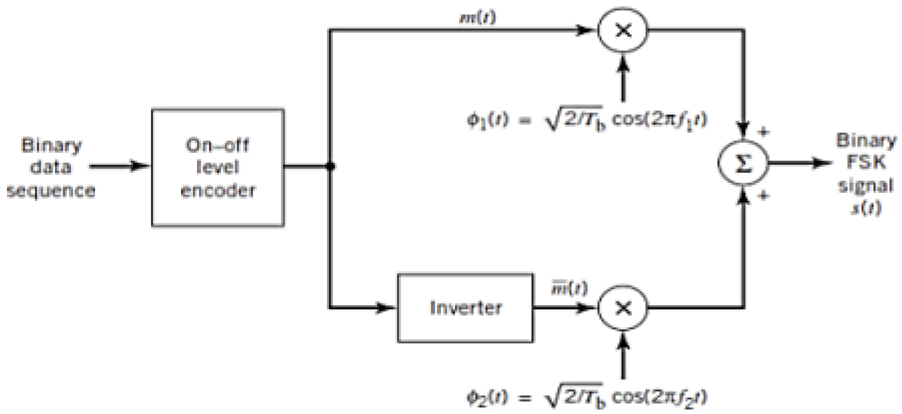


Fig.6: BFSK generation

(ii) Detection

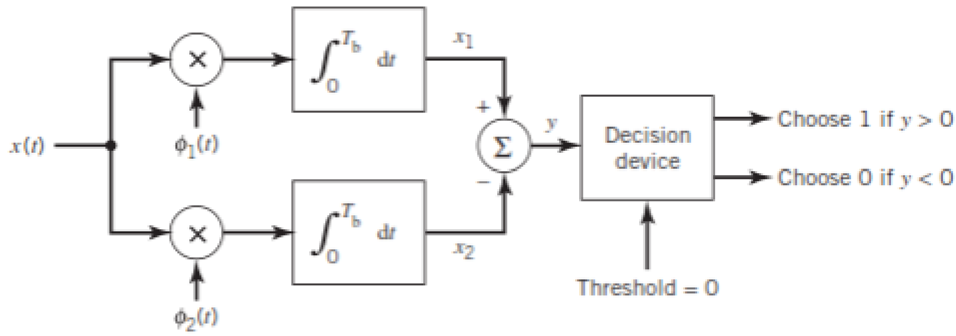


Fig.7: BFSK detection

It consists of two correlators with a common input, and reference signals $\varnothing_1(t)$, $\varnothing_2(t)$ are applied.

Then $y = x_1 - x_2$

The output y is compared with the threshold =0

If y>0 then output = 1 else 0.

But if y=0 then the receiver makes a random guess of 0 or 1.

Error Calculation

In binary FSK system the basic functions are given by

$$\phi_1(t) = \sqrt{\frac{2}{T_b}} \cos 2\pi f_1 t \qquad 0 \le t \le T_b$$
$$\phi_2(t) = \sqrt{\frac{2}{T_b}} \cos 2\pi f_2 t \qquad 0 \le t \le T_b$$

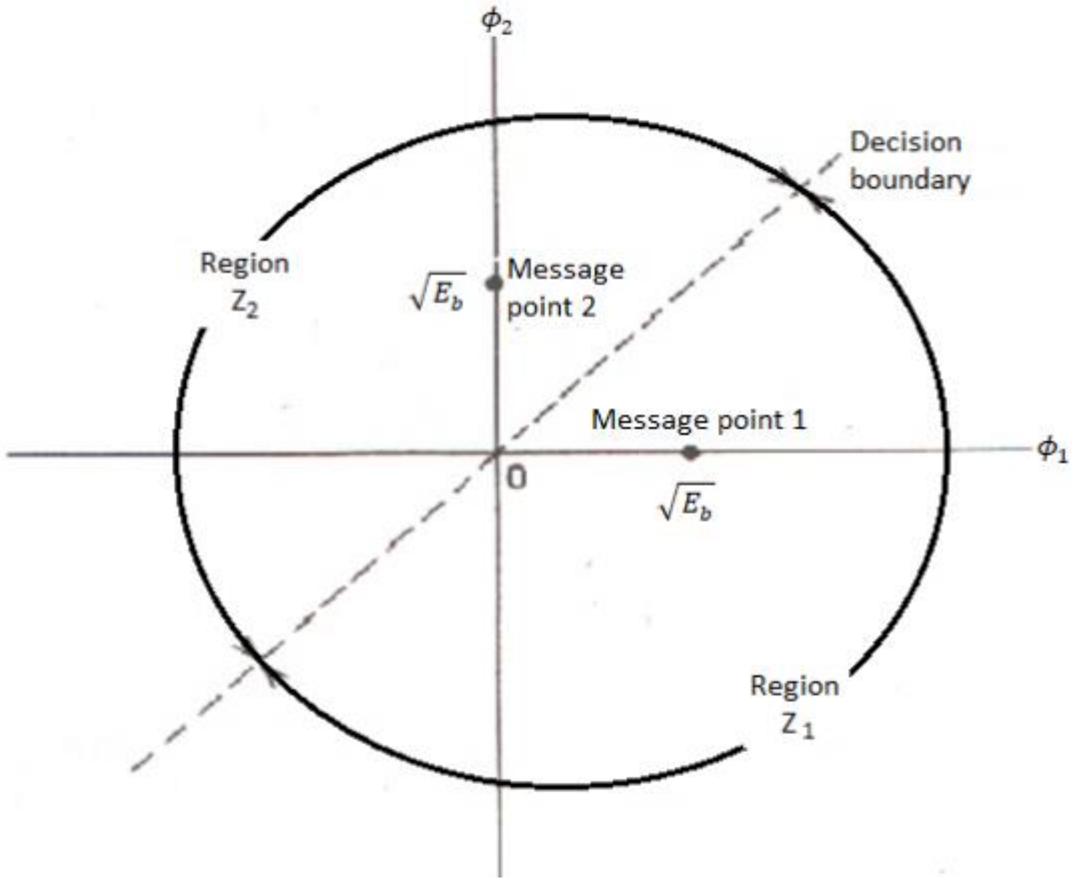
The transmitted signals S1(t) and S2(t) are given by

$S_1(t) = \sqrt{E_b}\phi_1(t)$ for symbol 1

$S_2(t) = \sqrt{E_b}\phi_2(t)$ for symbol 0

Therefore, Binary FSK system has 2-dimensional signal space with two messages S1(t) and S2(t), [N=2, m=2] they are represented as shown in figure

Fig.8



Signal Space diagram of Coherent binary FSK system.

The two message points are defined by the signal vector

$S_1 = \begin{bmatrix} \sqrt{E_b} \\ 0 \end{bmatrix}$ and $S_2 = \begin{bmatrix} 0 \\ \sqrt{E_b} \end{bmatrix}$

The observation vector x1 and x2 (output of upper and lower correlator) are related to input signal x(t) as

$x_1 = \int_0^{T_b} x(t)\phi_1(t)dt$ and $x_2 = \int_0^{T_b} x(t)\phi_2(t)dt$

Assuming zero mean additive white Gaussian noise with input PSD $N_0/2$. With variance $N_0/2$. The new observation vector 'l' is the difference of two random variables x_1 & x_2.

$l = x_1 - x_2$

When symbol '1' was transmitted x1 and x2 has mean value of 0 and √Eb respectively. Therefore, the conditional mean of random variable 'l' for symbol 1 was transmitted is

$$E\left[\frac{l}{1}\right] = E\left[\frac{x_1}{1}\right] - E\left[\frac{x_2}{1}\right] = \sqrt{E_b} - 0 = \sqrt{E_b}$$

Similarly, for '0' transmission

$$E\left[\frac{l}{0}\right] = -\sqrt{E_b}$$

The total variance of random variable 'l' is given by

$Var[l] = Var[x_1] + Var[x_2] = N_o$

The probability of error is given by

$$P_e\left(\frac{1}{0}\right) = P_{e0} = \frac{1}{\sqrt{2\pi N_0}}\int_0^\infty exp\left[-\frac{(l+\sqrt{E_b})^2}{2N_0}\right]dl$$

$$Z = \frac{l+\sqrt{E_b}}{\sqrt{2N_0}}$$

$$P_{e0} - \frac{1}{\pi}\int_{\frac{\sqrt{E_b}}{\sqrt{2N_0}}}^{\infty} exp\,(-z)^2 dz$$

$$Z = \frac{1+\sqrt{E_b}}{\sqrt{2N_0}}$$

Similarly,

$$P_{e1} = \frac{1}{2} erfc\left[\sqrt{\frac{E_b}{2N_0}}\right]$$

The total probability of error =

$$P_e = \frac{1}{2}\left[P_e\left(\frac{1}{0}\right) + P_e\left(\frac{0}{1}\right)\right]$$

Assuming 1's & 0's with equal probabilities

$$P_e = \frac{1}{2}[P_{e0} + P_{e1}]$$

$$P_e = \frac{1}{2} erfc\left[\sqrt{\frac{E_b}{2N_0}}\right]$$

Key Takeaways:

The FSK modulator block diagram comprises of two oscillators with a clock and the input binary sequence

The main methods of FSK detection are **asynchronous detector** and **synchronous detector**.

2.5 Quadrature Phase Shift Keying (QPSK)

GENERATION AND COHERENT DETECTION OF QPSK SIGNALS

(i) Generation

The QPSK Modulator uses a bit-splitter, two multipliers with local oscillator, a 2-bit serial to parallel converter, and a summer circuit.

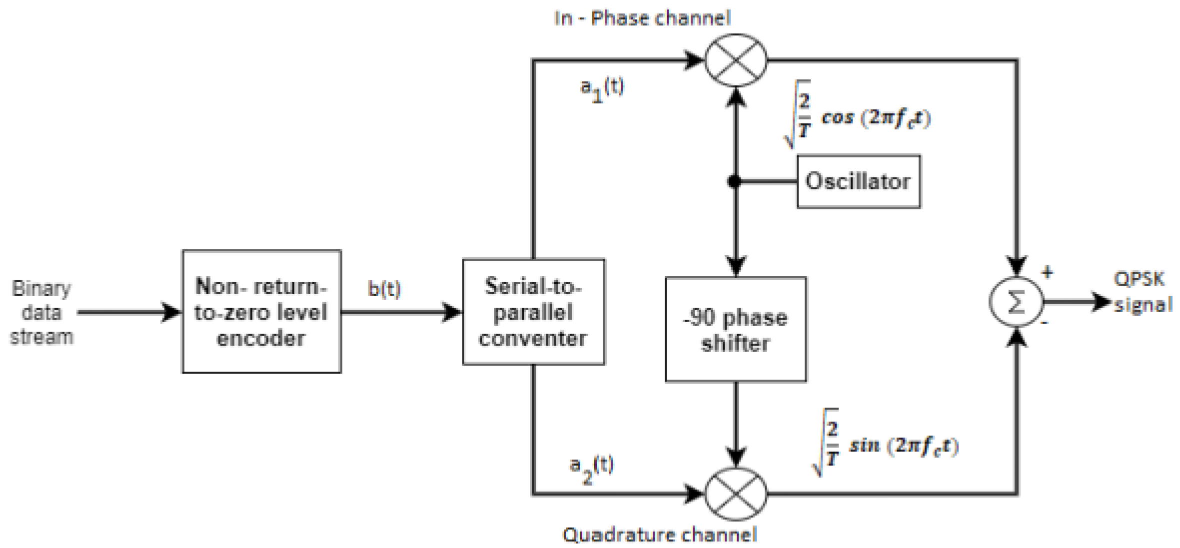
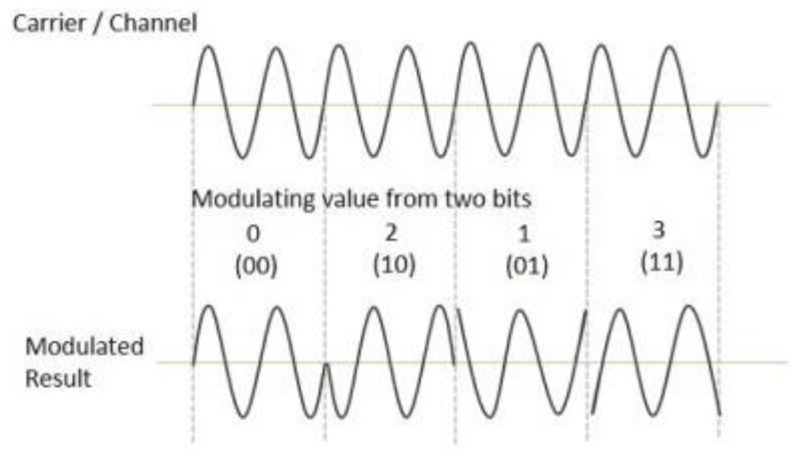


Fig.9: QPSK generation

At the modulator's input, the message signal's even bits (i.e., 2nd bit, 4th bit, 6th bit, etc.) and odd bits (i.e., 1st bit, 3rd bit, 5th bit, etc.) are separated by the bits splitter and are multiplied with the same carrier to generate odd BPSK (called as **PSK$_I$**) and even BPSK (called as **PSK$_Q$**). The **PSK$_Q$** signal is anyhow phase shifted by 90° before being modulated.

The QPSK waveform for two-bits input is as follows, which shows the modulated result for different instances of binary inputs.



(ii) Detection

The QPSK Demodulator uses two product demodulator circuits with local oscillator, two band pass filters, two integrator circuits, and a 2-bit parallel to serial converter.

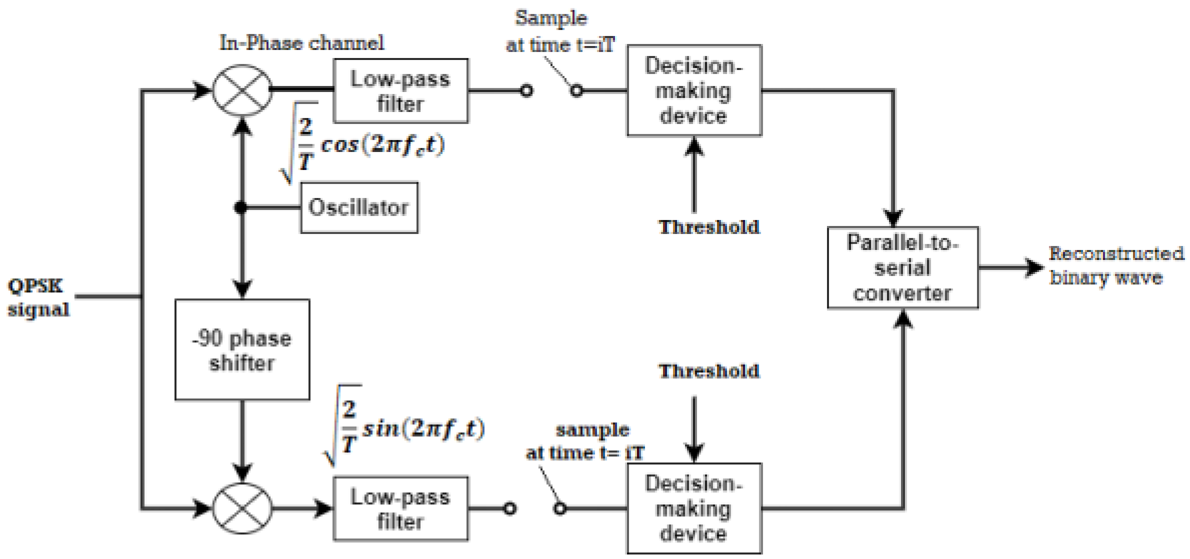


Fig.10: QPSK detectors

The two product detectors at the input of demodulator simultaneously demodulate the two BPSK signals. The pair of bits are recovered here from the original data. These signals after processing are passed to the parallel to serial converter.

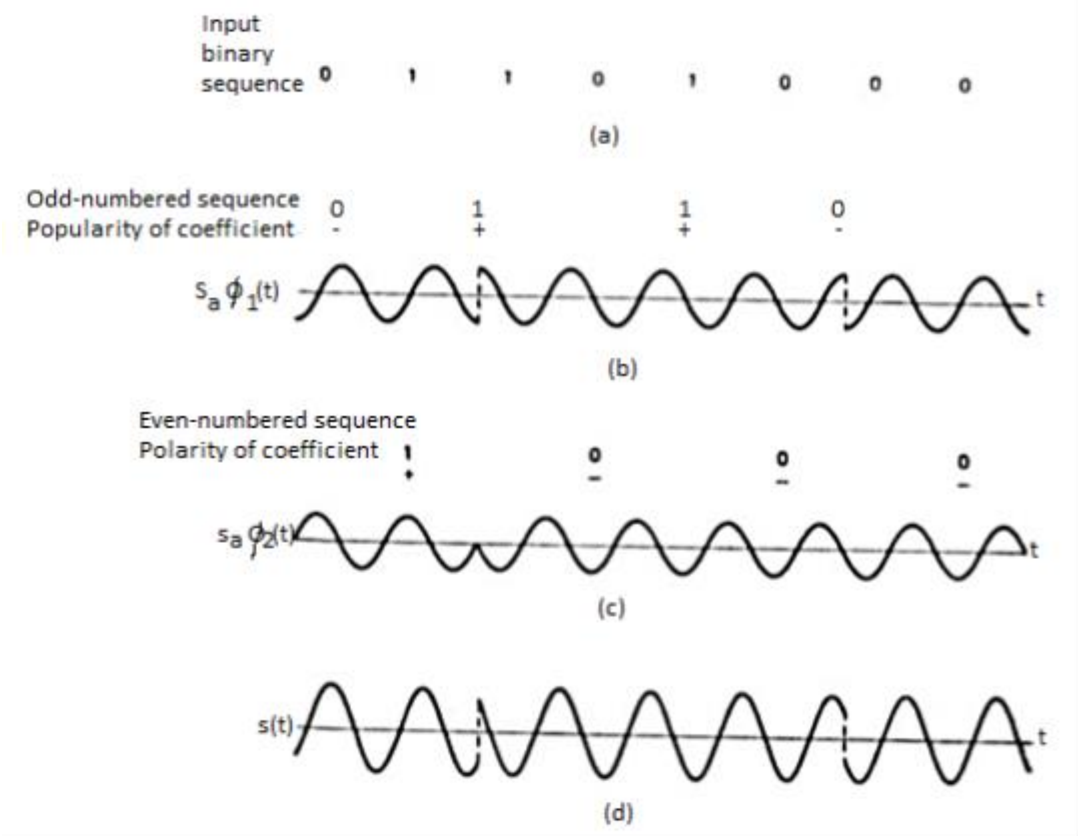


Fig.11: QPSK output

Error Calculation

In QPSK system the information carried by the transmitted signal is contained in the phase. The transmitted signals are given by

$$S_1(t) = \sqrt{\frac{2E}{T}} cos\left[2\pi f_c t + \frac{\pi}{4}\right] \qquad \rightarrow for\ input\ di\ bit\ 10$$

$$S_2(t) = \sqrt{\frac{2E}{T}} cos\left[2\pi f_c t + \frac{3\pi}{4}\right] \qquad \rightarrow for\ input\ di\ bit\ 00$$

$$S_3(t) = \sqrt{\frac{2E}{T}} cos\left[2\pi f_c t + \frac{5\pi}{4}\right] \qquad \rightarrow for\ input\ di\ bit\ 01$$

$$S_{43}(t) = \sqrt{\frac{2E}{T}} cos\left[2\pi f_c t + \frac{7\pi}{4}\right] \qquad \rightarrow for\ input\ di\ bit\ 11$$

Where the carrier frequency $f_c=n_c/7$ for some fixed integer n_c.

E = the transmitted signal energy per symbol.

T = Symbol duration.

The basic functions $\phi_1(t)$ and $\phi_2(t)$ are given by

$$\phi_1(t) = \sqrt{\frac{2}{T_b}}\cos\left[2\pi f_c t\right] \qquad 0 \le t < T$$

$$\phi_2(t) = \sqrt{\frac{2}{T_b}}\sin\left[2\pi f_c t\right] \qquad 0 \le t < T$$

There are four message points and the associated signal vectors are defined by

$$S_i \begin{bmatrix} \sqrt{E}\cos\left[(2i-1)\frac{\pi}{4}\right] \\ -\sqrt{E}\sin\left[(2i-1)\frac{\pi}{4}\right] \end{bmatrix}$$

The table shows the elements of signal vectors, namely Si1 & Si2

Input dibit	Phase of QPSK signal (radians)	Coordinates of message points	Input dibit
		S_{i1}	S_{i2}
10	$\frac{\pi}{4}$	$+\sqrt{\frac{E}{2}}$	$-\sqrt{\frac{E}{2}}$
00	$\frac{3\pi}{4}$	$-\sqrt{\frac{E}{2}}$	$-\sqrt{\frac{E}{2}}$
01	$\frac{5\pi}{4}$	$+\sqrt{\frac{E}{2}}$	$+\sqrt{\frac{E}{2}}$
11	$\frac{7\pi}{4}$	$+\sqrt{\frac{E}{2}}$	$+\sqrt{\frac{E}{2}}$

Therefore, a QPSK signal is characterized by having a two-dimensional signal constellation (i.e., N=2) and four message points (i.e., M=4) as illustrated in figure.

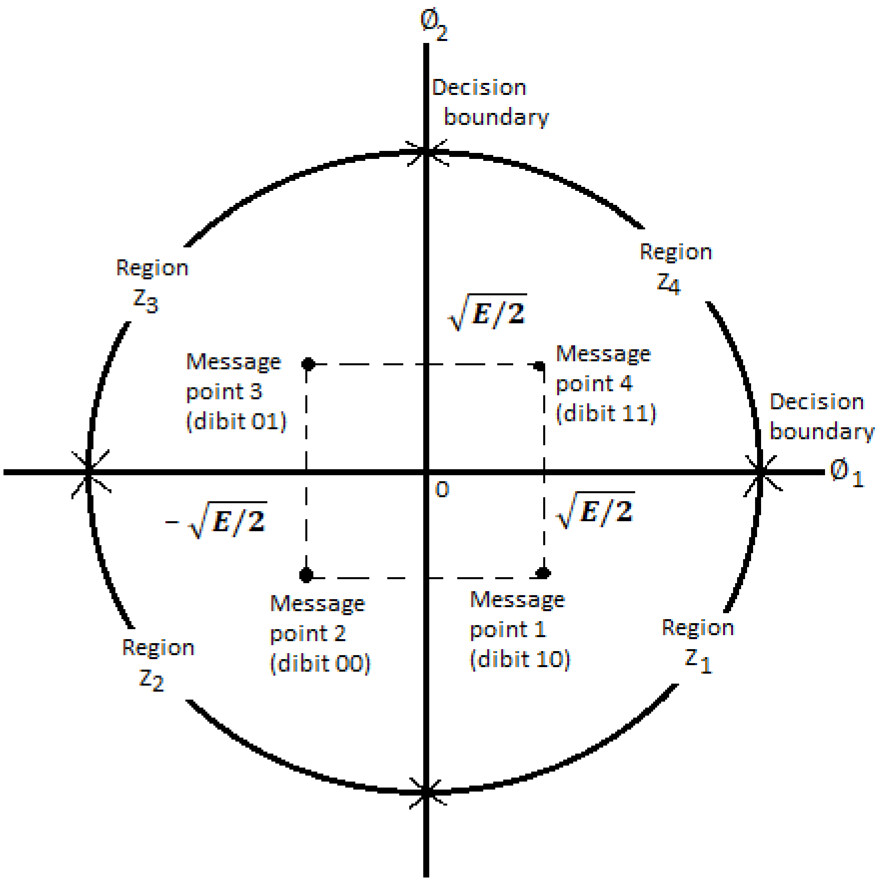


Fig 12. Signal-space diagram of coherent QPSK system

A QPSK system is in fact equivalent to two coherent binary PSK systems working in parallel and using carriers that are in-phase and quadrature. The in-phase channel output x1 and the Q-channel output x2 may be viewed as the individual outputs of the two coherent binary PSK systems.

Thus, the two binary PSK systems may be characterized as follows

The signal energy per bit √E/2

The noise spectral density is $N_o/2$

The average probability of bit error in each channel of the coherent QPSK system is $P^1 = \frac{1}{2} erfc\left[\sqrt{\frac{E/2}{N_0}}\right]$ $\left(E = \frac{E}{2}\right)$

$$= \frac{1}{2} erfc\left[\sqrt{\frac{E}{2N_0}}\right]$$

The bit errors in the I-channel and Q-channel of the QPSK system are statistically independent. The I-channel makes a decision on one of the two bits constituting a symbol (di bit) of the QPSK signal and the Q-channel takes care of the other bit. Therefore, the average probability of a direct decision resulting from the combined action of the two channels working together is p_c= probability of correct reception p^1= probability of error

$$P_C = [1 - P^1]^2 = \left[1 - \frac{1}{2}erfc\left[\sqrt{\frac{E}{2N_0}}\right]\right]^2$$

$$= 1 - erfc\left[\sqrt{\frac{E}{2N_0}}\right] + \frac{1}{4}erfc^2\left[\sqrt{\frac{E}{2N_0}}\right]$$

The average probability of symbol error for coherent QPSK is given by

$$P_e = 1 - P_C = erfc\left[\sqrt{\frac{E}{2N_0}}\right] - \frac{1}{4}erfc^2\left[\frac{E}{\sqrt{2N_0}}\right]$$

In the region where $E/2N_o$ We may ignore the second term and so the approximate formula for the average probability of symbol error for coherent QPSK system is

$$P_e = erfc\sqrt{\frac{E}{2N_0}}$$

Key takeaway

QPSK Modulator uses a bit-splitter, two multipliers with local oscillator, a 2-bit serial to parallel converter, and a summer circuit.

2.6 M-ary Phase Shift Keying (MPSK)

Minimum Shift Key Modulation is another type of digital modulation technique used to convert a digital signal into analog signals. It is also called Minimum-shift keying (MSK) or Advance Frequency Shift Keying because it is a type of continuous-phase frequency-shift keying.

Key features

- It is encoded with bits alternating between quadrature components, with the Q component delayed by half the symbol period.
- Minimum Shift Keying is the most effective digital modulation technique. It can be implemented for almost every stream of bits much easier than the Phase Shift Key, Frequency Shift Key and Amplitude Shift Key of digital modulation technique.
- The Minimum Shift Keying's concept is based on the positioning of bits such as even bits and odd bits for the given bitstream and the bit positioning frequency generating table.
- MSK is the most widely used digital modulation technology because of its ability and flexibility to handle "One (1)" and "Zero (0)" transition of binary bits.

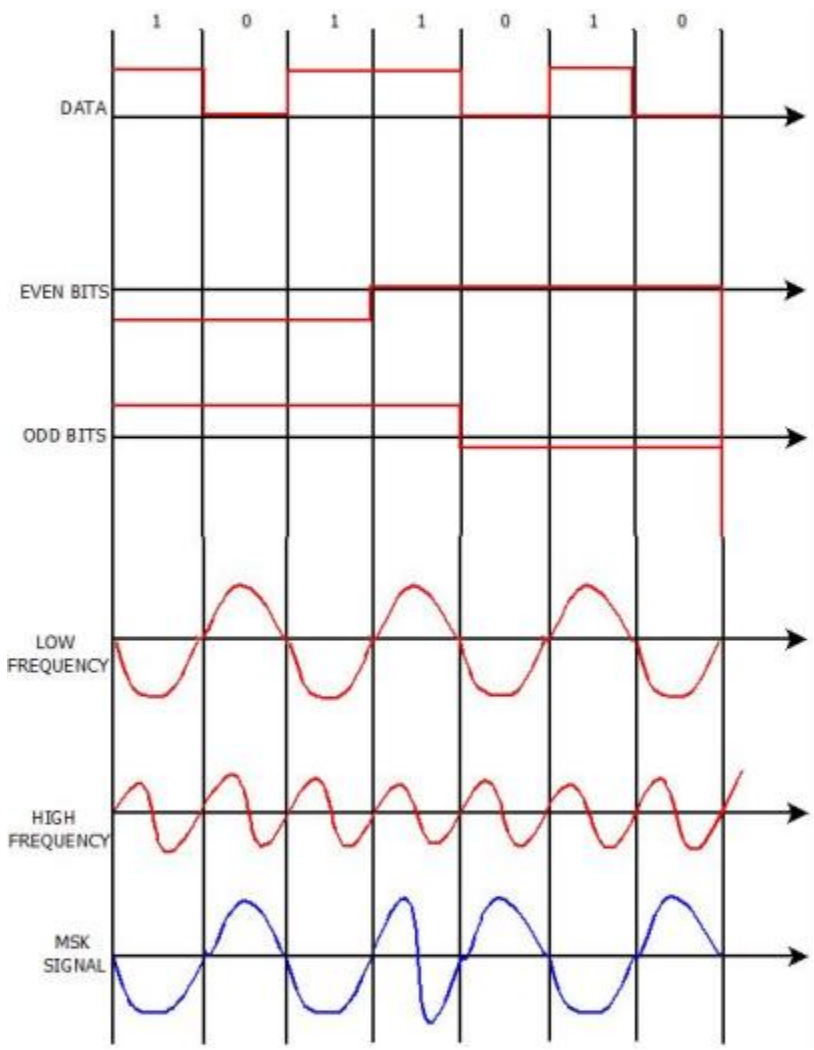


Fig.13: MSK output

The word binary represents two bits. **M** 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.

M-ary Equation

If a digital signal is given under four conditions, such as voltage levels, frequencies, phases, and amplitude, then **M = 4**.

The number of bits necessary to produce a given number of conditions is expressed mathematically as

$N=\log_2 M$

Where

N is the number of bits necessary

M is the number of conditions, levels, or combinations possible with **N** bits.

The above equation can be re-arranged as

$2^N=M$

For example, with two bits, **22 = 4** conditions are possible.

M-ary PSK

This is called as M-ary Phase Shift Keying M−ary PSK

The **phase** of the carrier signal, takes on **M** different levels.

Representation of M-ary PSK

$S_i(t)=\sqrt{2E/T}\cos(w_0 t+\phi_i t)$ $0\le t\le T$ and $i=1,2...M$

ϕi(t)=2πiM where i=1,2,3......M

Some prominent features of M-ary PSK are −

- The envelope is constant with more phase possibilities.
- This method was used during the early days of space communication.
- Better performance than ASK and FSK.
- Minimal phase estimation error at the receiver.
- The bandwidth efficiency of M-ary PSK decreases and the power efficiency increases with the increase in **M**.

So far, we have discussed different modulation techniques. The output of all these techniques is a binary sequence, represented as **1s** and **0s**

Error Calculation

In M-ary PSK, M different phase shifts of the carrier are used to convey the information. The $M = 2^k$ signal waveforms, each representing k information bits, are represented as

$$s_i(t) = A_C v(t) cos\left[2\pi f_c t + \psi_i + \varphi\right],\ 0 \le t \le T \qquad i = 1,\ldots,M$$

$$\varphi = 0\ \ or \frac{\pi}{M}$$

$$\psi_i = \frac{2\pi(i-1)}{M} = M$$

The M-ary PSK signal

$$x(t) = \sqrt{\frac{2E_s}{T}} \sum_{n=-\infty}^{\infty} v(t-nT) cos\left[2\pi f_C t + \psi_n + \psi_n + \varphi\right]$$

By choosing the same basis functions as for QPSK, it is possible to express all waveforms in the M-PSK signal set as vectors in the plane spanned by φ1 and φ2 as

$$s = \left(a_n^I \sqrt{E_s}, a_n^Q \sqrt{E_s}\right)$$

Where

$$a_n^I = cos\ (\psi_n + \varphi)$$
$$a_n^Q = sin\ (\psi_n + \varphi)$$

The signal vectors lie around a circle of radius √ Es The constellation for 8- PSK (M = 8) is shown in Figure

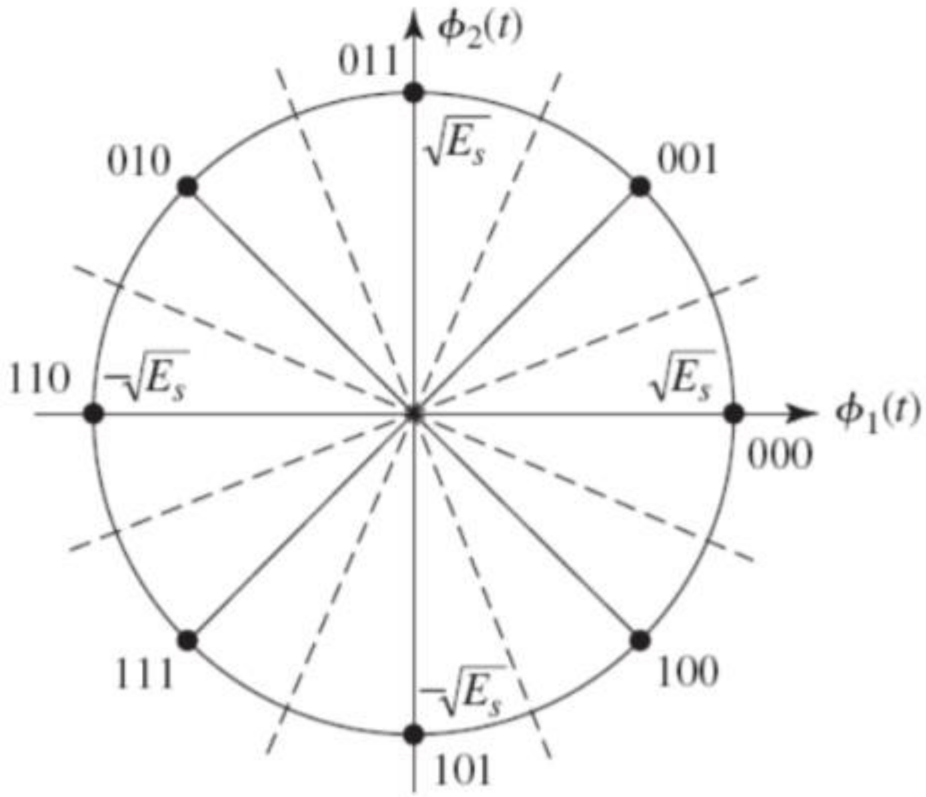


Fig 14 Signal space diagram for MSK system

The minimum distance between two adjacent signal points is

$$d_{min} \ = 2D = 2\sqrt{E_s} sin\left(\frac{\pi}{M}\right)$$

The nearest-neighbour estimate of Pe is

$$P \approx 2Q\left[\sqrt{\frac{2E_s}{N_0}} sin^2\left(\frac{\pi}{M}\right)\right]$$
$$= 2Q\left[\sqrt{\frac{2E_b log_2 M}{N_0}} sin^2\left(\frac{\pi}{M}\right)\right]$$
$$BER_{MPSK} \backslash apprx \frac{1}{log_2 M} 2Q\left[\sqrt{\frac{2E_b log_2 M}{N_0}} sin^2\left(\frac{\pi}{M}\right)\right]$$

Key takeaway

The transmitted M number of signals are equal in energy and duration.

The signals are separated by 1/2Ts Hz making the signals orthogonal to each other.

Since M signals are orthogonal, there is no crowding in the signal space.

Modulation scheme	Distance	Bandwidth
BPSK	$d = 2\sqrt{P_S T_b} = 2\sqrt{E_b}$	$B = 2f_s = f_b$
QPSK	$d = 2\sqrt{P_S T_b} = 2\sqrt{E_b}$	$B = 2f_s = f_b$
M-PSK	$d = \sqrt{4E_s sin^2\left(\frac{\pi}{M}\right)} = \sqrt{4NE_b sin^2\left(\frac{\pi}{2^N}\right)}$	$B = 2f_S = \frac{2f_b}{N}$
16-QAM	$d = 2\sqrt{0.4E_b}$	$B = \frac{f_b}{2}$
BFSK	$d = \sqrt{2E_b}$	(Orthogonal BFSK)

$E_b = energy\ contained\ in\ a\ bit\ duration,$

N- number of bits per symbol

$\boldsymbol{M = 2^N (number\ of\ symbols)}$

$$\boldsymbol{f_b = \frac{1}{T_b}(bit\ rate),\ \ f_S = \frac{1}{T_s} = \frac{1}{NT_b} = \frac{f_b}{N}\quad (symbol\ rate)}$$

Solved Examples

Q1) Determine the baud and minimum bandwidth necessary to pass a 10-kbps binary signal using amplitude shift keying.

A1) For ASK, N = 1, and the baud and minimum bandwidth are determined from

$B=f_b/N$

Where N is the number of bits encoded into each signaling element.

$B = 10{,}000 / 1 = 10{,}000$

Baud = 10, 000 /1 = 10,000

Q2) Determine (a) the peak frequency deviation, (b) minimum bandwidth, and (c) baud for a binary FSK signal with a mark frequency of 49 kHz, a space frequency of 51 kHz, and an input bit rate of 2 kbps.

A2)

a) Frequency deviation is $\Delta f = |fm - fs| / 2$

Where Δf = frequency deviation (hertz) $|fm - fs|$ = absolute difference between the mark and space frequencies (hertz)

$\Delta f = |149\text{kHz} - 51\text{ kHz}| / 2 = 1\text{ kHz}$

b. The minimum bandwidth is $B = 2(\Delta f + fb)$

Where B= minimum Nyquist bandwidth (hertz)

Δf= frequency deviation | (fm– fs) | (hertz)

Fb = input bit rate (bps)

B = 2(1000 + 2000) = 6 kHz c. For FSK, N = 1, and the baud is determined from Equation 2.11 as baud = 2000 / 1 = 2000

c. For FSK, N = 1, and the baud is determined from Equation 2.11 as baud = 2000 / 1 = 2000

Q3) For a BPSK modulator with a carrier frequency of 70 MHz and an input bit rate of 10 Mbps, determine the maximum and minimum upper and lower side frequencies, draw the output spectrum, determine the minimum Nyquist bandwidth, and calculate the baud.

A3) Substituting into Equation

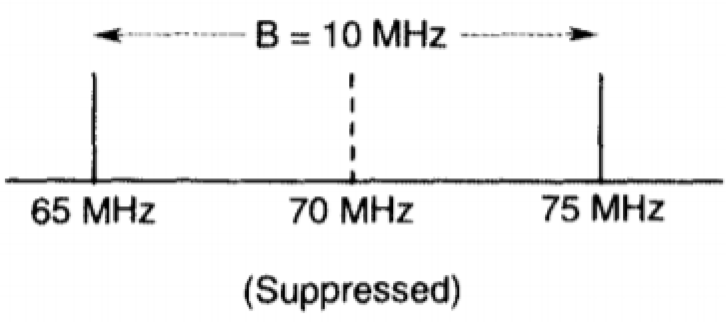
BPSK output = [sin (2πfa t)] x [sin (2πfc t)]

Fa = fb / 2 = 5 MHz = [sin 2π(5MHz) t)] x [sin 2π(70MHz) t)] = 0.5cos [2π (70MHz – 5MHz) t] – 0.5cos [2π (70MHz + 5MHz) t]

Minimum lower side frequency (LSF): LSF=70MHz - 5MHz = 65MHz

Maximum upper side frequency (USF): USF = 70 MHz + 5 MHz = 75 MHz

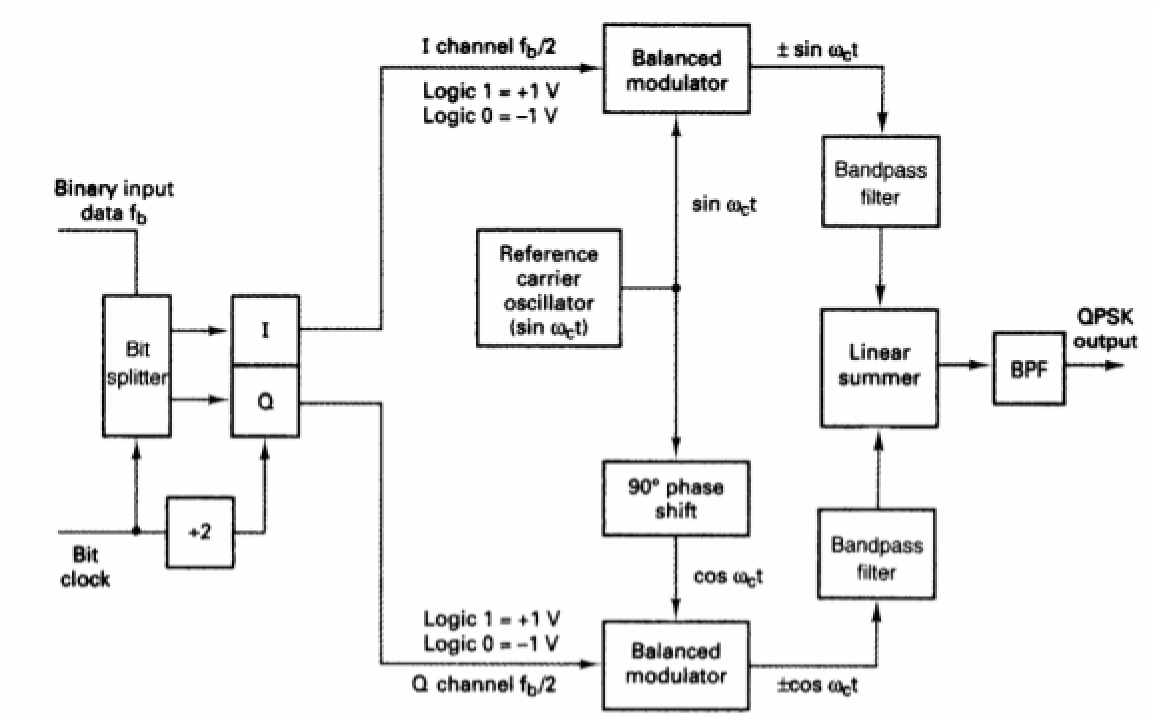
Therefore, the output spectrum for the worst-case binary input conditions is as follows: The minimum Nyquist bandwidth (B) is



B = 75 MHz - 65 MHz = 10 MHz

And the baud = fb or 10 megabaud.

Q4) For the QPSK modulator shown in Figure, construct the truth table, phasor diagram, and constellation diagram.



A4) For a binary data input of Q = O and I= 0, the two inputs to the I balanced modulator are -1 and sin ωc t, and the two inputs to the Q balanced modulator are -1 and cos ωc t.

Consequently, the outputs are

I balanced modulator = (-1) (sin ωc t) = -1 sin ωc t

Q balanced modulator = (-1) (cos ωc t) = -1 cos ωc t and

The output of the linear summer is -1 cos ωc t - 1 sin ωc t = 1.414 sin (ωc t - 135°)

For the remaining dibit codes (01, 10, and 11), the procedure is the same.

Binary input		QPSK output phase
Q	I	
0	0	-135
0	1	-45
1	0	+135
1	1	+45

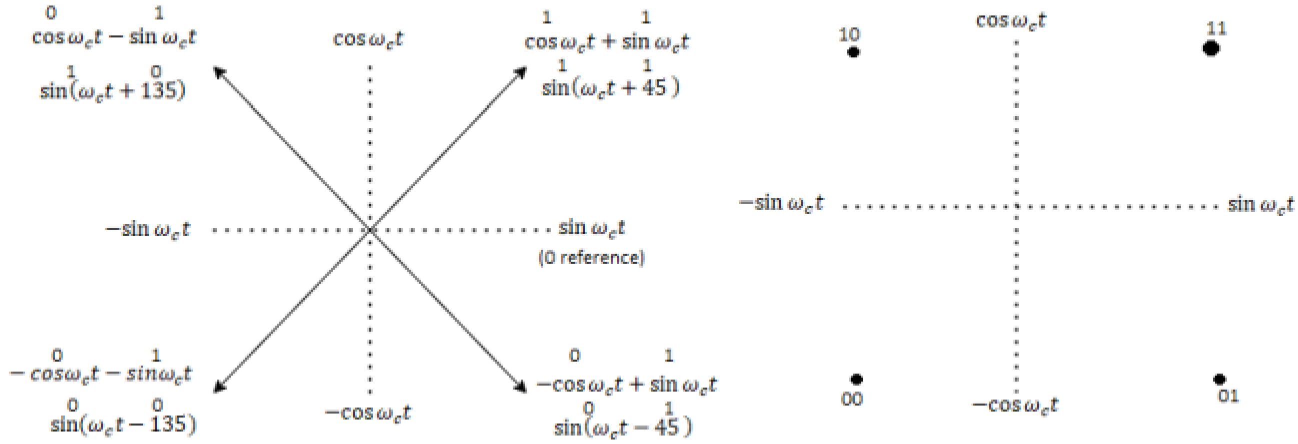


Fig QPSK modulator: (a) truth table; (b) phasor diagram; (c) constellation diagram

In Figures b and c, it can be seen that with QPSK each of the four possible output phasors has exactly the same amplitude. Therefore, the binary information must be encoded entirely in the phase of the output signal.

Figure b, it can be seen that the angular separation between any two adjacent phasors in QPSK is 90°. Therefore, a QPSK signal can undergo almost a+45° or -45° shift in phase during transmission and still retain the correct encoded information when demodulated at the receiver. Figure shows the output phase-versus-time relationship for a QPSK modulator.

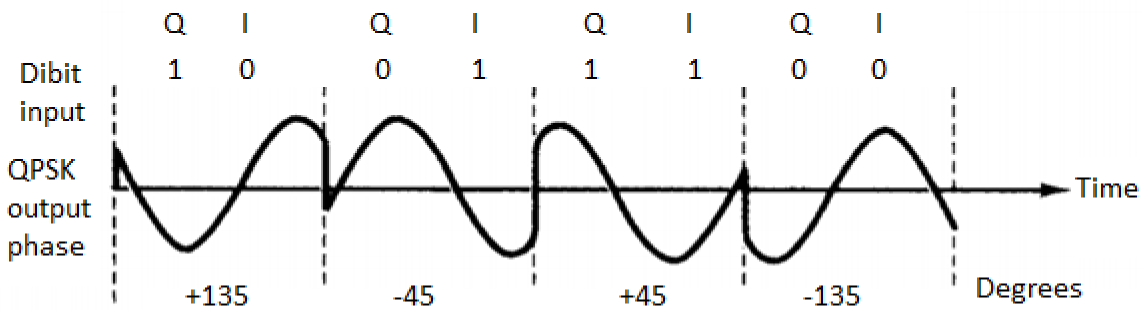


Fig Output phase-versus-time relationship for a PSK modulator.

Q5) For a QPSK modulator with an input data rate (fb) equal to 10 Mbps and a carrier frequency 70 MHz, determine the minimum double-sided Nyquist bandwidth (fN) and the baud.

A5) The bit rate in both the I and Q channels is equal to one-half of the transmission bit rate, or

$f_{bQ} = f_{bI} = f_b / 2 = 10$ Mbps / 2 = 5 Mbps

The highest fundamental frequency presented to either balanced modulator is

Fa= f_{bQ} / 2 = 5 Mbps / 2 = 2.5 MHz

The output wave from each balanced modulator is

(sin 2πfa t) (sin 2πfc t)

0.5 cos 2π (fc – fa) t – 0.5 cos 2π (fc + fa) t

0.5 cos 2π [(70 – 2.5) MHz]t – 0.5 cos 2π [(70 – 2.5) MHz]t

0.5 cos 2π(67.5MHz) t - 0.5 cos 2π(72.5MHz) t

The minimum Nyquist bandwidth is B= (72.5-67.5) MHz = 5MHz

The symbol rate equals the bandwidth: thus, symbol rate = 5 megabaud

Q6) For an 8-PSK system, operating with an information bit rate of 24 kbps, determine (a) baud, (b) minimum bandwidth, and (c) bandwidth efficiency.

A6)

a. Baud is determined by

Baud = 24 kbps / 3 = 8000

b. Bandwidth is determined by

B = 24 kbps / 3 = 8000

c. Bandwidth efficiency is calculated

$B\eta$ = 24, 000 / 8000 = 3 bits per second per cycle of bandwidth

Q7) For a QPSK system and the given parameters, determine a. Carrier power in dBm. b. Noise power in dBm. c. Noise power density in dBm. d. Energy per bit in dBJ. e. Carrier-to-noise power ratio in dB. f. EblNo ratio. $C = 10^{-12}$ W, F_b = 60 kbps, $N = 1.2 \times 10^{-14}$W, B = 120 kHz

A7) a. The carrier power in dBm is determined by

$C = 10 \log (10^{-12} / 0.001) = - 90$ dBm

b. The noise power in dBm is determined by

$N = 10 \log [(1.2 x 10^{-14}) / 0.001] = -109.2$ dBm

c. The noise power density is determined by

No = -109.2 dBm – 10 log 120 kHz = -160 dBm d.

The energy per bit is determined by $Eb = 10 \log (10^{-12} / 60 \text{ kbps}) = -167.8$ dBJ

e. The carrier-to-noise power ratio is determined by

$C / N = 10 \log (10^{-12} / 1.2 \text{ x } 10\text{-}14) = 19.2$ dB

f. The energy per bit-to-noise density ratio is determined by

Eb / No = 19.2 + 10 log 120 kHz / 60 kbps = 22.2 dB

Q8) Determine the minimum bandwidth required to achieve a P(e) of 10-7 for an 8-PSK system operating at 10 Mbps with a carrier-to noise power ratio of 11.7 dB.

A8) The minimum bandwidth is

B / fb = Eb / No = C / N = 14.7 dB – 11.7 dB = 3 dB

B / fb = antilog 3 = 2

B = 2 x 10 Mbps = 20 MHz

Q9) We need to send data 3 bits at a time at 9 bit rate of 3mbps. The carrier frequency is 10MHz. Calculate the number of levels the baud rate and bandwidth?

A9) n= 3 bits

R= 3Mbps

f_c= 10MHz

L=2^n = 23=8

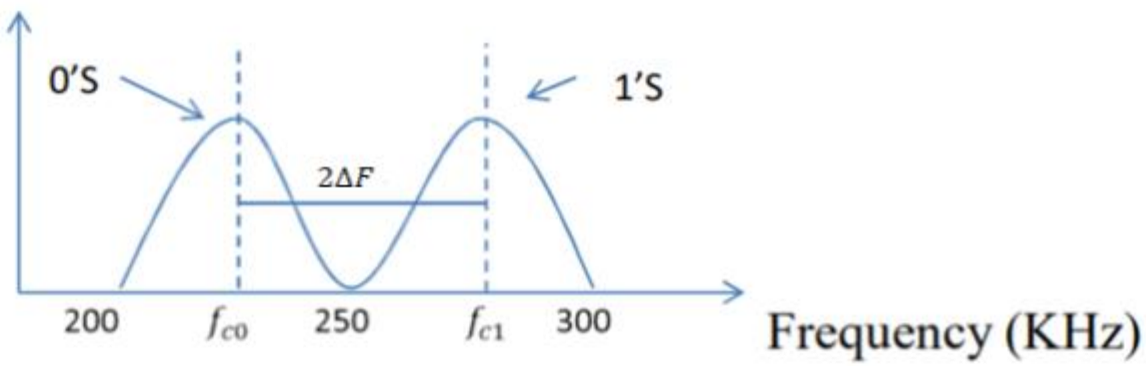
The number of levels =8

Baud rate = Bit rate/n= 3Mpbs/3 =1Mbaud

Bandwidth = L*r=8*1Mbaud= 8MHz

Q10) We have an available bandwidth of 100kHz which spans from 200- 300kHz. What should be the carrier frequency and the bit rate if we modulated the data by using FSK?

A10)



Q

$f_{c0} = 225\ KHz$
$f_{c1} = 275\ KHz$

$$
\begin{aligned}
&B.W = f_{C1} - f_{c0} + 2R_b \\
&100K = 275 - 225 + 2R_b \\
&100K = 50 + 2R_b \\
&R_b = 25K\ baud
\end{aligned}
$$

Q11) A voice signal is sampled at the rate of 8000 samples/sec and each sample is encoded in to 8 bits using PCM. The binary data is transmitted into free space after modulation. Determine the bandwidth of modulated signal, if the modulation technique is ASK, FSK, PSK for f_1 =10MHz, f_2= 8MHz?

A11)

Bit rate $R_b = r * n$

Baud rate=8000 samples/sec

No. Of bits per sample =8

$\therefore R_b = 8000 \times 8 = 64000$

Band width of A$SK = 2 \times R_b = 2 \times 64000 = 128\ kHz$

Band width of $PSK = 2 \times R_b = 2 \times 64000 = 128\ kHz$

Band width of $FSK = (f_1 - f_2) + 2R_b$

$$
\begin{aligned}
&= (10 - 8) + (2 \times 64000) \\
&= 2M + 128000 \\
&= 2M + 0.128\ M \\
&= 2.128\ M
\end{aligned}
$$

Q12) Find the maximum bit rates of an FSK signal in bps, if the bandwidth of the medium is 12 kHz and the difference between the two carriers is 2 kHz (given that transmission mode is full duplex)

A12) As transmission is full duplex only 6kHz is allotted for each direction.

Bandwidth = 2Bit Rate + f_{c1}-f_{c0}

2Bit Rate= Bandwidth – (f_{c1}-f_{c0}) =6000-2000=4000

Bit rate= 2000bps

Q13) A BPSK modulator has a carrier frequency of 70MHz and an input bit rate of 10Mbps. Calculate the maximum upper side band frequency in MHz?

A13)

$$\left(\frac{o}{p}\right)BPSK = (\sin 2\pi f_a t)(\sin 2\pi f_C t)$$

f_a is maximum fundamental frequency of binary input $= fb/2$

f_b is input bit rate

f_c is reference carrier frequency

The output of a BPSK modulator with carrier frequency of 70 MHz and bit rate of 10 Mbps is

$$\left(\frac{O}{P}\right)_{BPSK} = [\sin (2\pi \times 5 \times 10^6)t] \times [\sin (2\pi \times 70 \times 10^6)t]$$
$$= 0.5\cos [2\pi \times (70 \times 10^6 - 5 \times 10^6)t)] - 0.5\cos [2\pi \times (70 \times 10^6 + 5 \times 10^6)t)]$$

Therefore, the maximum upper sideband frequency

$$= 70 \times 10^6 + 5 \times 10^6 = 75\ MHz$$

Q14) In digital communication channel employing FSK, the 0 and 1 bits are represented by sine waves of 10kHz and 25kHz respectively. These waveforms will be orthogonal for a bit interval of?

A14)

For orthogonality of two sine waves in T_b duration there should be integral multiple of cycles of both sine wave.

Time period of first sine wave

$$T_{b1} = \frac{1}{10 \times 10^3} = 100\mu s$$

Time period of the second sine wave

$$T_{b2} = \frac{1}{25 \times 10^3} = 40\mu s$$

Therefore, 200μs is the integral multiple of both T_{b1} and T_{b2}. Hence the two given waveforms will be orthogonal for a bit interval of 200μs.

Q15) Find the minimum bandwidth (in kHz) required for transmitting an ASK signal at 2 kbps in half duplex transmission mode.

A15) In ASK, the baud rate and bit rate are the same. Therefore, baud rate is 2000 bauds. An ASK signal requires a minimum bandwidth equal to its baud rate. Therefore, minimum bandwidth = 2000 Hz = 2 kHz

Q16) A communication system employs ASK to transmit a 10 kbps binary signal. Find the baud rate required in bauds.

A16) For an ASK system, the minimum bandwidth is the same as the bit rate of the signal. Therefore, minimum bandwidth = 10000 Hz

Q17) For the data given in the above Question, find the minimum bandwidth required in hertz

A17) For an ASK system, the minimum bandwidth is the same as the bit rate of the signal. Therefore, minimum bandwidth = 10000 Hz

Q18) Determine the peak frequency deviation, for a binary FSK signal with a mark frequency of 51 kHz, a space frequency of 49 kHz, and an input bit rate of 2.

A18) The peak frequency deviation is Δf = (mark frequency-space frequency)/2 =|49kHz - 51 kHz| / 2 =1 kHz kbps.

Q19) Determine minimum bandwidth for a binary FSK signal with a mark frequency of 51 kHz, a space frequency of 49 kHz, and an input bit rate of 2.

A19) minimum bandwidth = 2(Δf + $2R_b$) = 2(1+2) =6K

References:

1. "Communication Systems", Simon Haykin, Wiley publication, 4th Edition, 2004

2. "Digital Communication Fundamentals and Applications", Bernard Sklar, Pearson Education India, 2nd Edition, 2009

3. "Modern Electronic Communication", Miller Gary M, Prentice-Hall, 6th Edition, 1999

4. "Digital Communications", John Proakis, Tata Mc Graw hill, 5th Edition, 2007

5. "Electronic Communication Systems, Fundamentals Through Advanced", Wayne Tomsi, Pearson Education, 4th Edition, 2001

6. Communication systems by B.P. Lathi

Unit - 3

Digital Modulation-II

3.1 Generation, Reception, Signal Space Representation and Probability of Error Calculation for Quadrature Amplitude Shift Keying (QASK)

QAM (also known as QASK) is a combination of ASK and PSK. Here, both the amplitude and the phase are varied to transmit more bits per symbol.

The QAM modulator essentially follows the idea that can be seen from the basic QAM theory where there are two carrier signals with a phase shift of 90° between them. These are then amplitude modulated with the two data streams known as the I or In-phase and the Q or quadrature data streams. These are generated in the baseband processing area.

Basic QAM I-Q modulator circuit

The two resultant signals are summed and then processed as required in the RF signal chain, typically converting them in frequency to the required final frequency and amplifying them as required.

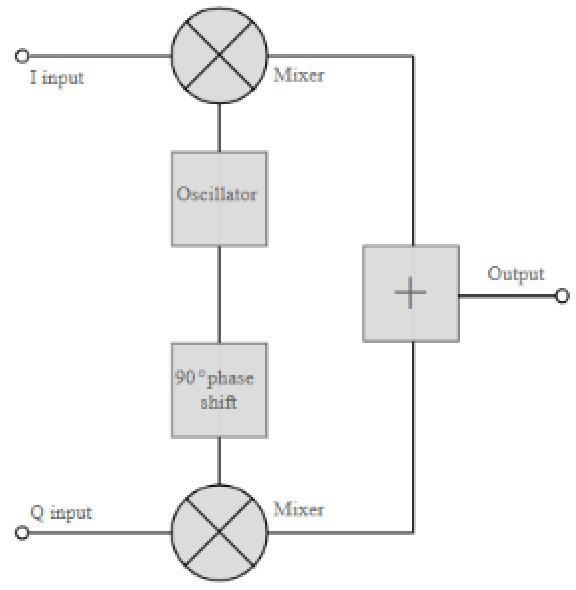


Fig.1: Basic QAM

It is worth noting that as the amplitude of the signal varies any RF amplifiers must be linear to preserve the integrity of the signal. Any non-linearities will alter the relative levels of the signals and alter the phase difference, thereby distorting he signal and introducing the possibility of data errors.

Basic QAM I-Q demodulator circuit

The basic modulator assumes that the two quadrature signals remain exactly in quadrature.

A further requirement is to derive a local oscillator signal for the demodulation that is exactly on the required frequency for the signal. Any frequency offset will be a change in the phase of the local oscillator signal with respect to the two double sideband suppressed carrier constituents of the overall signal.

Systems include circuitry for carrier recovery that often utilises a phase locked loop - some even have an inner and outer loop. Recovering the phase of the carrier is important otherwise the bit error rate for the data will be compromised.

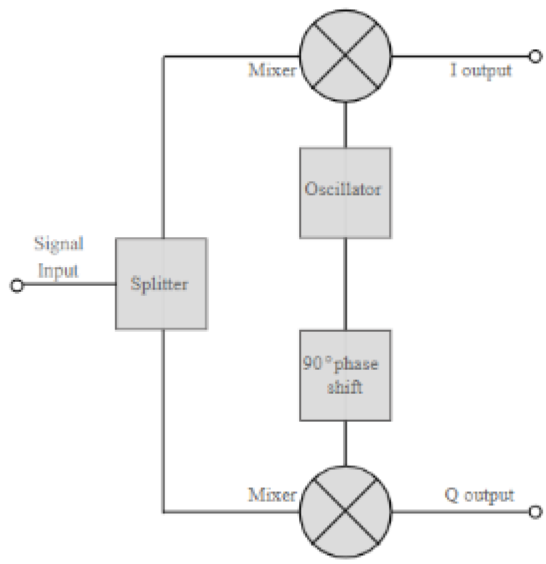


Fig.2: Basic QAM demodulator

The circuits shown above show the generic IQ QAM modulator and demodulator circuits that are used in a vast number of different areas. Not only are these circuits made from discrete components, but more commonly they are used within integrated circuits that are able to provide a large number of functions.

Error Probability

BER for QAM constellation

The SER for a rectangular M-QAM (16-QAM, 64-QAM, 256-QAM etc) with size $L = M^2$ can be calculated by considering two M-PAM on in-phase and quadrature components. The error probability of QAM symbol is obtained by the error probability of each branch (M-PAM) and is given by

$$P_e \leq 15Q\left(\sqrt{\frac{d^2_{min}}{2N_O}}\right) = 15Q\left(\sqrt{\frac{2A^2}{N_O}}\right)$$

Signal Space Representation

For the case when $M = 2^k$, k even, the resulting signal space diagram has a "square constellation." In this case the QAM signal can be thought of as 2 PAM signals in quadrature. For $M = 2^k$, k odd, the constellation takes on a "cross" form. For example, 16-QAM constellation is

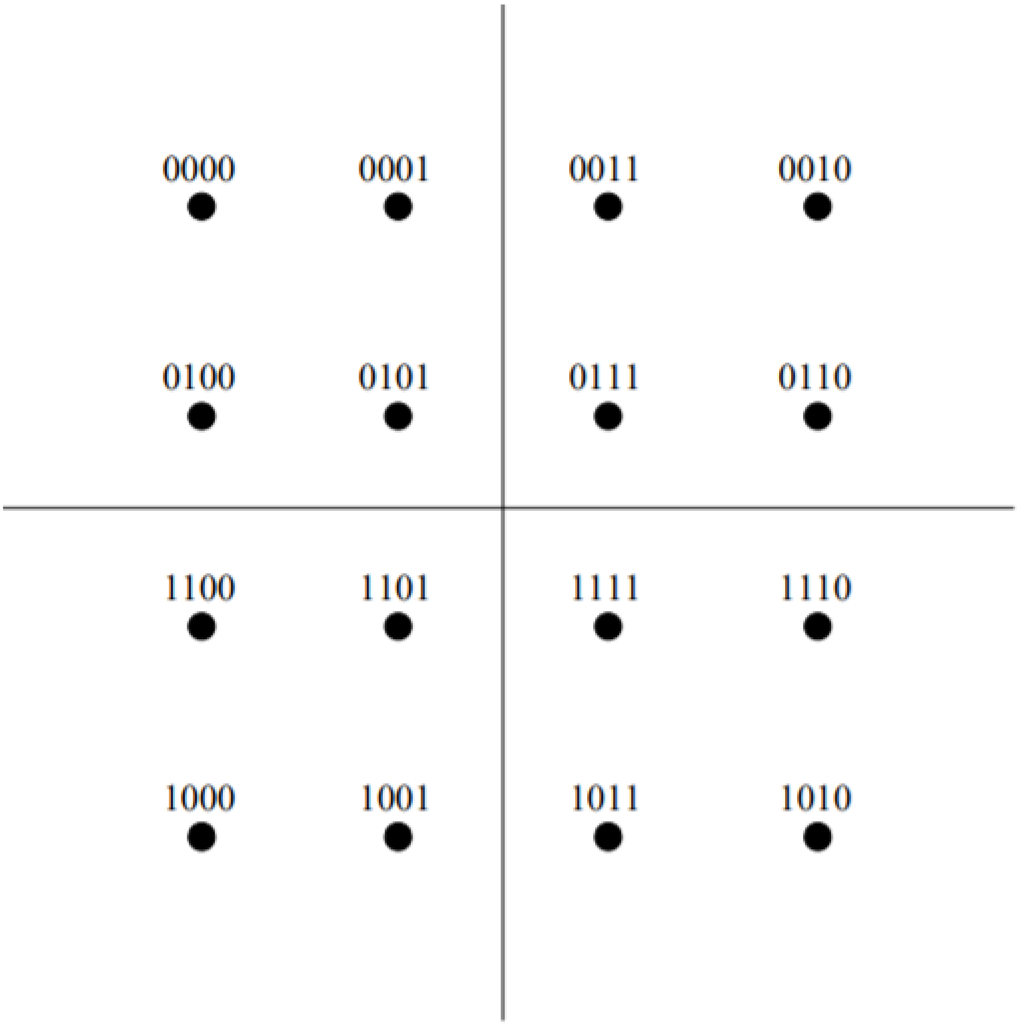


Fig 3:16-QAM constellation

Key takeaway

QAM (also known as QASK) is a combination of ASK and PSK. Here, both the amplitude and the phase are varied to transmit more bits per symbol.

3.2 M-ary FSK (MFSK)

Consider next the M-ary version of FSK, for which the transmitted signals are defined by

$$s_i(t) = \sqrt{\frac{2E}{T}}\cos\left[\frac{\pi}{T}(n_c + i)t\right], \qquad 0 \le t \le T$$

Where $i = 1, 2, \ldots, M$, and the carrier frequency $f_c = n_c/(2T)$ for some fixed integer n_c. The transmitted symbols are of equal duration T and have equal energy E. Since the individual signal frequencies are separated by 1/(2T) Hz, the M-ary FSK signals constitute an orthogonal set; that is,

$$\int_0^T s_i(t)s_j(t)dt = 0, \qquad i \neq j$$

Hence, we may use the transmitted signals si (t) themselves, except for energy normalization, as a complete orthonormal set of basis functions, as shown by

$\phi_i(t) = \frac{1}{\sqrt{E}}s_i(t)$, for $0 \le t \le T$ and $i = 1,2,\ldots,M$

Accordingly, the M-ary FSK is described by an M-dimensional signal-space diagram. For the coherent detection of M-ary FSK signals, the optimum receiver consists of a bank of M correlators or matched filters, with ϕ_i (t) providing the basis functions. At the sampling times t = kT, the receiver makes decisions based on the largest matched filter output in accordance with the maximum likelihood decoding rule. An exact formula for the probability of symbol error is, however, difficult to derive for a coherent M-ary FSK system. Nevertheless, we may use the union bound to place an upper bound on the average probability of symbol error for M-ary FSK. Specifically, since the minimum distance d_{min} in M-ary FSK is $\sqrt{2E}$, we have

$$P_e \le (M-1)Q\left(\sqrt{\frac{E}{N_0}}\right)$$

For fixed M, this bound becomes increasingly tight as the ratio E/N0 is increased. Indeed, it becomes a good approximation to Pe for values of $P_e \leq 10^{-3}$.

Power Spectra of M-ary FSK Signals

The spectral analysis of M-ary FSK signals is much more complicated than that of M-ary PSK signals. A case of particular interest occurs when the frequencies assigned to the multilevel make the frequency spacing uniform and the frequency deviation h = 1/2. That is, the M signal frequencies are separated by 1/2T, where T is the symbol duration. For h = 1/2, the baseband power spectral density of M-ary FSK signals is plotted in Figure below for M = 2, 4, 8.

Bandwidth Efficiency of M-ary FSK Signals

When the orthogonal signals of an M-ary FSK signal are detected coherently, the adjacent signals need only be separated from each other by a frequency difference 1/2T so as to maintain orthogonality. Hence, we may define the channel bandwidth required to transmit M-ary FSK signals as

B=M/2T

For multilevels with frequency assignments that make the frequency spacing uniform and equal to 1/2T, the bandwidth B contains a large fraction of the signal power.

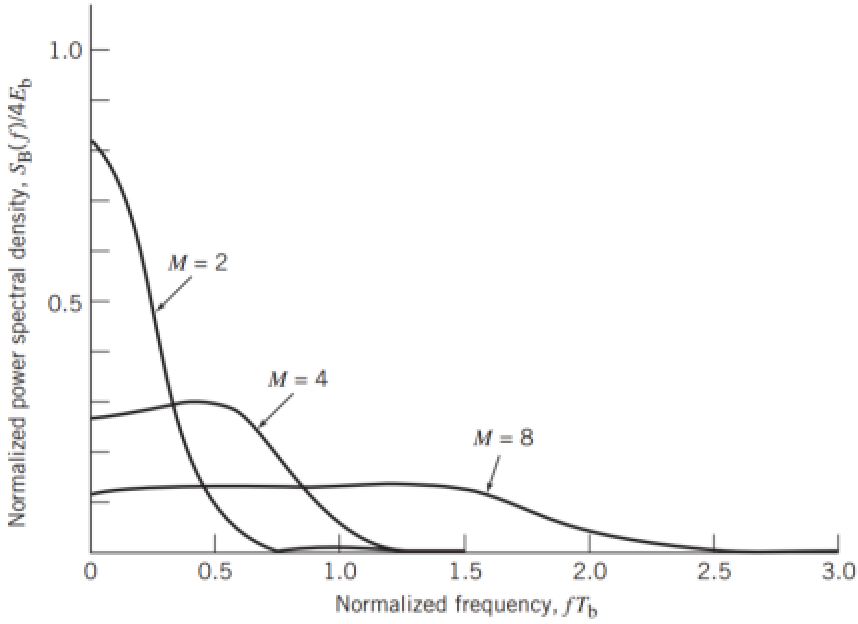


Fig 4: Power spectra of M-ary PSK signals for M = 2, 4, 8.[5]

Hence, using $R_b=1/T_b$, we may redefine the channel bandwidth B for M-ary FSK signals as

$$B = \frac{R_b M}{2 \log_2 M}$$

The bandwidth efficiency of M-ary signals is therefore

$$\rho = \frac{R_b}{B} = \frac{2 \log_2 M}{N}$$

Key takeaway

When the orthogonal signals of an M-ary FSK signal are detected coherently, the adjacent signals need only be separated from each other by a frequency difference 1/2T so as to maintain orthogonality. Hence, we may define the channel bandwidth required to transmit M-ary FSK signals as

B=M/2T

3.3 Minimum Shift Keying (MSK)

In the coherent detection of binary FSK signal, the phase information contained in the received signal is not fully exploited, other than to provide for synchronization of the receiver to the transmitter. We now show that by proper use of the continuous-phase property when performing detection, it is possible to improve the noise performance of the receiver significantly. Here again, this improvement is achieved at the expense of increased system complexity. Consider a continuous-phase frequency-shift keying (CPFSK) signal, which is defined for the signaling interval $0 \leq t \leq Tb$ as follows:

$$s(t) = \begin{cases} \sqrt{\frac{2E_b}{T_b}} cos\,[2\pi f_1 t + \theta(0)] \; for\ symbol\ 1 \\ \sqrt{\frac{2E_b}{T_b}} cos\,[2\pi f_2 t + \theta(0)] \; for\ symbol\ 0 \end{cases}$$

Where Eb is the transmitted signal energy per bit and Tb is the bit duration.

Another useful way of representing the CPFSK signal s(t) is to express it as a conventional angle-modulated signal:

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

Where $\theta(t)$ is the phase of s(t) at time t. When the phase $\theta(t)$ is a continuous function of time, we find that the modulated signal s(t) is itself also continuous at all times, including the inter-bit switching times. The phase $\theta(t)$ of a CPFSK signal increases or decreases linearly with time during each bit duration of T_b seconds, as shown by

$$\theta(t) = \theta(0) \pm \left(\frac{\pi h}{T_b}\right) t, \quad 0 \leq t \leq T_b$$

Here the plus sign corresponds to sending symbol 1 and the minus sign corresponds to sending symbol 0; the dimensionless parameter h is to be defined. We deduce the following pair of relations:

$$f_c + \frac{h}{2T_B} = f_1$$
$$f_c - \frac{h}{2T_B} = f_2$$

Solving this pair of equations for fc and h, we get

$$f_c = \frac{1}{2}(f_1 + f_2)$$

$$h - T_b(f_1 - f_2)$$

Signal-Space Diagram of MSK

Using a well-known trigonometric identity of conventional angle modulated signal, we may expand the CPFSK signal s(t) in terms of its in-phase and quadrature components as

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

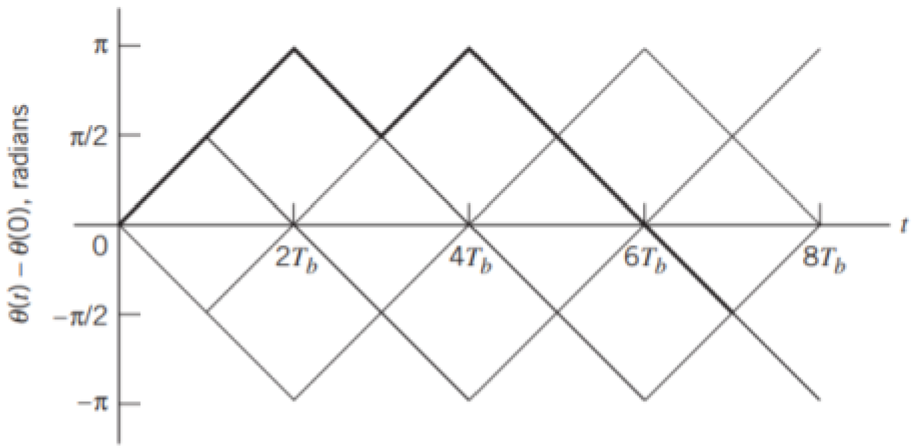


Fig 5: Boldfaced path represents the sequence 1101000.

Consider, first, the in-phase component $\sqrt{2E_b/T_b}\cos\theta(t)$

$$\theta(t) = \theta(0) \pm \frac{\pi}{2T_b}, \quad 0 \le t \le T_b$$

Where the plus sign corresponds to symbol 1 and the minus sign corresponds to symbol 0. A similar result holds for θ(t) in the interval $-T_b \le t \le 0$, except that the algebraic sign is not necessarily the same in both intervals. Since the phase θ(0) is 0 or π depending on the past history of the modulation process, we find that in the interval $-T_b \le t \le T_b$, the polarity of cosθ(t) depends only on θ(0), regardless of the sequence of 1s and 0s transmitted before or after t = 0. Thus, for this time interval, the in-phase component consists of the half-cycle cosine pulse:

$$s_I(t) = \sqrt{\frac{2E_b}{T_b}} \cos\theta(t)$$

$$= \sqrt{\frac{2E_b}{T_b}} \cos\theta(0) \cos\left(\frac{\pi}{2T_b}t\right)$$

$$= \pm \sqrt{\frac{2E_b}{T_b}} \cos\left(\frac{\pi}{2T_b}t\right) \qquad -T_b \le t \le T_b$$

Where the plus sign corresponds to $\theta(0) = 0$ and the minus sign corresponds to $\theta(0) = \pi$. In a similar way, we may show that, in the interval $0 \le t \le 2Tb$, the quadrature component of s(t) consists of the half-cycle sine pulse:

$$s_Q(t) = \sqrt{\frac{2E_b}{T_b}} \sin\theta(t)$$

$$= \sqrt{\frac{2E_b}{T_b}} \sin\theta\,(T_b) \sin\left(\frac{\pi}{2T_b}t\right)$$

$$= \pm \sqrt{\frac{2E_b}{T_b}} \sin\left(\frac{\pi}{2T_b}t\right) \qquad 0 \le t \le 2T_b$$

Where the plus sign corresponds to $\theta(Tb) = \pi/2$ and the minus sign corresponds to $\theta(Tb)=-\pi/2$. From the discussion just presented, we see that the in-phase and quadrature components of the MSK signal differ from each other in two important respects:

- They are in phase quadrature with respect to each other and
- The polarity of the in-phase component sI(t) depends on $\theta(0)$, whereas the polarity of the quadrature component sQ(t) depends on $\theta(Tb)$.

Moreover, since the phase states $\theta(0)$ and $\theta(Tb)$ can each assume only one of two possible values, any one of the following four possibilities can arise:

1. $\theta(0) = 0$ and $\theta(Tb) = \pi/2$, which occur when sending symbol 1.

2. $\theta(0) = \pi$ and $\theta(Tb) = \pi/2$, which occur when sending symbol 0.

3. $\theta(0) = \pi$ and $\theta(Tb) = -\pi/2$ (or, equivalently, $3\pi/2$ modulo 2π), which occur when sending symbol 1.

4. $\theta(0) = 0$ and $\theta(Tb) = -\pi/2$, which occur when sending symbol 0.

This fourfold scenario, in turn, means that the MSK signal itself can assume one of four possible forms, depending on the values of the phase-state pair: $\theta(0)$ and $\theta(T_b)$.

Signal-Space Diagram

We see that there are two orthonormal basis functions $\phi_1(t)$ and $\phi_2(t)$ characterizing the generation of MSK; they are defined by the following pair of sinusoidally modulated quadrature carriers:

$$\phi_1(t) = \sqrt{\frac{2}{T_b}} cos\left(\frac{\pi}{2T_b}t\right) cos\,(2\pi f_C t) \qquad 0 \le t \le T_b$$

$$\phi_2(t) = \sqrt{\frac{2}{T_b}} sin\left(\frac{\pi}{2T_b}t\right) sin\,(2\pi f_C t) \qquad 0 \le t \le T_b$$

$$s(t) = s_1\phi_1(t) + s_1\phi_1(t) \;\; 0 \le t \le T_b$$

Where the coefficients s_1 and s_2 are related to the phase states $\theta(0)$ and $\theta(T_b)$, respectively. To evaluate s_1, we integrate the product $s(t)\phi_1(t)$ with respect to time t between the limits $-T_b$ and T_b, obtaining

$$s_1 = \int_{-T_b}^{T_b} s(t)\phi_1(t)dt$$

$$= \sqrt{E_b} cos\,[\theta(0)] \quad -T_b \le t \le T_b$$

Similarly, to evaluate s_2 we integrate the product $s(t)\phi_2(t)$ with respect to time t between the limits 0 and $2T_b$, obtaining

$$s_2 = \int_0^{2T_b} s(t)\phi_2(t)dt = \sqrt{E_b} sin\,[\theta(t_b)], \qquad 0 \le t \le T_b$$

From above two equations we conclude that

1. Both integrals are evaluated for a time interval equal to twice the bit duration.

2. The lower and upper limits of the integral used to evaluate s_1 are shifted by the bit duration T_b with respect to those used to evaluate s_2.

3. The time interval $0 \le t \le T_b$, for which the phase states $\theta(0)$ and $\theta(T_b)$ are defined, is common to both integrals.

It follows, therefore, that the signal constellation for an MSK signal is two-dimensional (i.e., N = 2), with four possible message points (i.e., M = 4), as illustrated in the signal space diagram of Figure below.

$(+\sqrt{E_b}, +\sqrt{E_b})$, $(-\sqrt{E_b}, +\sqrt{E_b})$, $(-\sqrt{E_b}, -\sqrt{E_b})$ and $(+\sqrt{E_b}, -\sqrt{E_b})$.

The possible values of θ(0) and θ(Tb), corresponding to these four message points, are also included in Figure below. The signal-space diagram of MSK is thus similar to that of QPSK in that both of them have four message points in a two-dimensional space. However, they differ in a subtle way that should be carefully noted:

- QPSK, moving from one message point to an adjacent one, is produced by sending a two-bit symbol (i.e., dibit).
- MSK, on the other hand, moving from one message point to an adjacent one, is produced by sending a binary symbol, 0 or 1. However, each symbol shows up in two opposite quadrants, depending on the value of the phase-pair: $\theta(0)$ and $\theta(T_b)$.

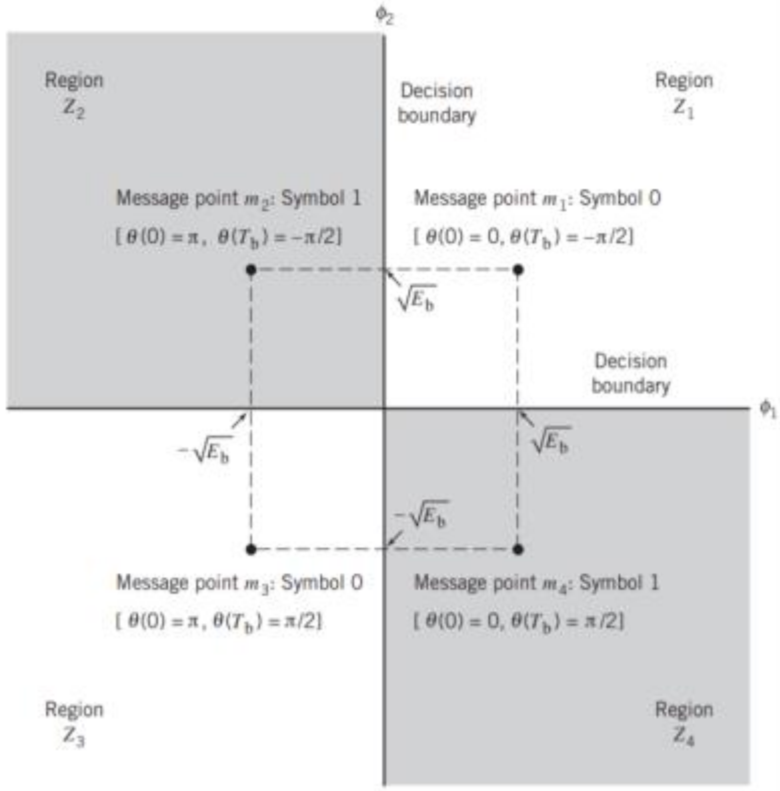


Fig 6: Signal-space diagram for MSK system

Generation and Coherent Detection of MSK Signals

With h = 1/2, we may use the block diagram of Figure(a) below to generate the MSK signal. The advantage of this method of generating MSK signals is that the signal coherence and deviation ratio are largely unaffected by variations in the input data rate. Two input sinusoidal waves, one of frequency $f_c = n_c/4T_b$ for some fixed integer n_c and the other of frequency $1/4T_b$, are first applied to a product modulator. This modulator produces two phase-coherent sinusoidal waves at frequencies f1 and f2, which are related to the carrier frequency fc and the bit rate $1/T_b$ in accordance with above equations for deviation ratio h = 1/2. These two sinusoidal waves are separated from each other by two narrowband filters, one centered at f_1 and the other at f_2. The resulting filter outputs are next linearly combined to produce the pair of quadrature carriers or orthonormal basis functions $\phi1(t)$ and $\phi2(t)$. Finally, $\phi_1(t)$ and $\phi_2(t)$ is multiplied with two binary waves a1(t) and a2(t), both of which have a bit rate equal to 1/(2Tb)

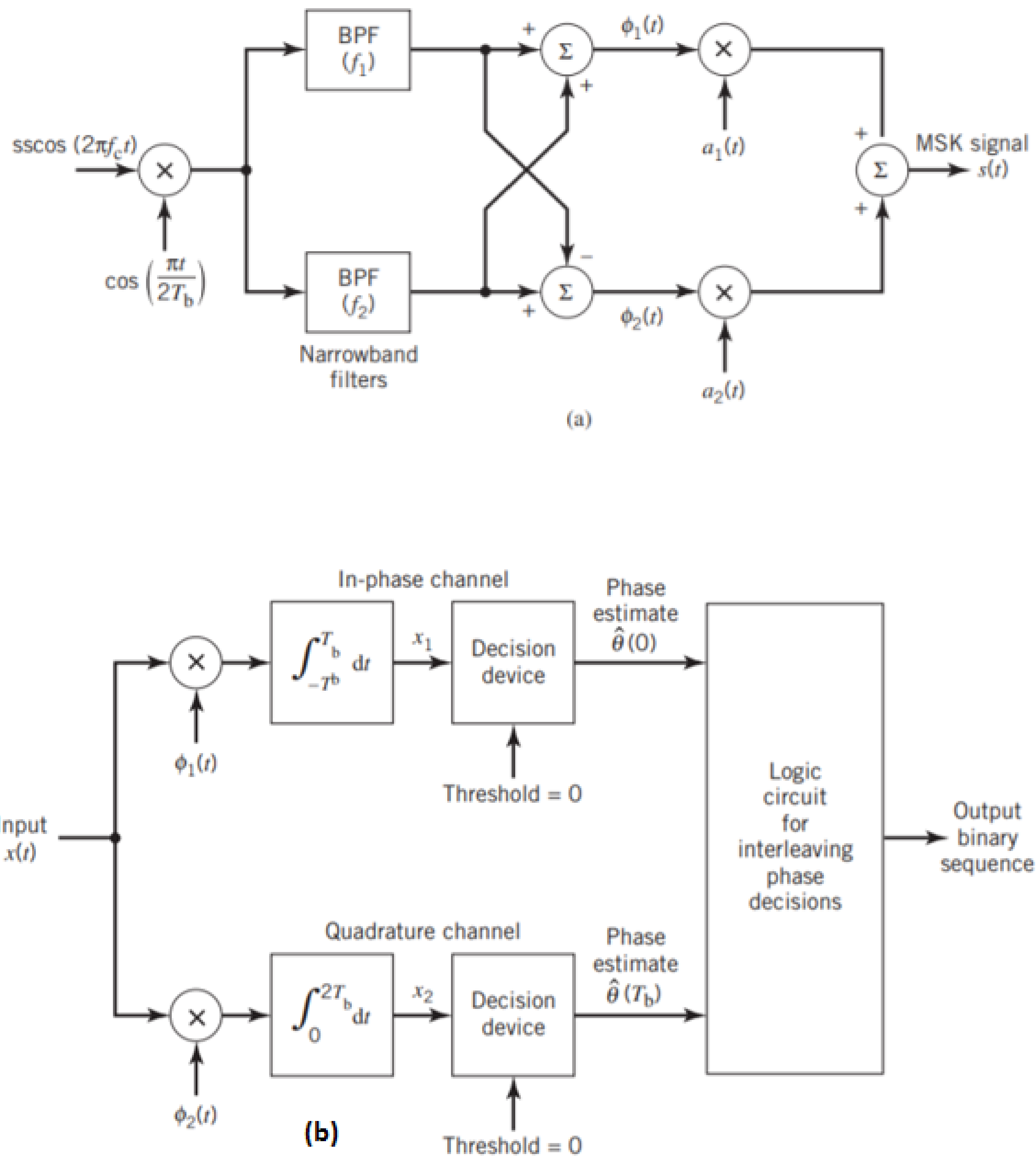


Fig 7: Block diagrams for (a) MSK transmitter and (b) coherent MSK receiver

Figure (b) above shows the block diagram of the coherent MSK receiver. The received signal x(t) is correlated with $\phi_1(t)$ and $\phi_2(t)$. In both cases, the integration interval is $2T_b$ seconds, and the integration in the quadrature channel is delayed by T_b seconds with respect to that in the in-phase channel. The resulting in-phase and quadrature channel correlator outputs, x_1 and x_2, are each compared with a threshold of zero

Error Probability of MSK

In the case of an AWGN channel, the received signal is given by

x(t) = s(t) + w(t)

Where s(t) is the transmitted MSK signal and w(t) is the sample function of a white Gaussian noise process of zero mean and power spectral density $N_0/2$. To decide whether symbol 1 or symbol 0 was sent in the interval $0 \leq t \leq T_b$, say, we have to establish a procedure for the use of x(t) to detect the phase states $\theta(0)$ and $\theta(T_b)$. For the optimum detection of $\theta(0)$, we project the received signal x(t) onto the reference signal over the interval $-Tb \leq t \leq Tb$, obtaining

$$x_1 = \int_{-T_b}^{T_b} x(t)\phi_1(t)dt \ = s_1 + w_1$$

w_1 is the sample value of a Gaussian random variable of zero mean and variance $N_0/2$. From the signal-space diagram, we see that if $x1 > 0$, the receiver chooses the estimate. On the other hand, if $x_1 < 0$, it chooses the estimate. Similarly, for the optimum detection of θ (T_b), we project the received signal x(t) onto the second reference signal $\phi2(t)$ over the interval $0 \leq t \leq 2T_b$, obtaining

$$x_2 = \int_{0}^{2T_b} x(t)\phi_2(t)dt \ = s_2 + w_2 \qquad 0 \leq t \leq 2T_b$$

w2 is the sample value of another independent Gaussian random variable of zero mean and variance N0/2. Referring again to the signal space diagram, we see that if $x2 > 0$, the receiver chooses the estimate $\theta(T_b) = -\pi/2$. If, however, $x_2 < 0$, the receiver chooses the estimate $\hat{\theta}(T_b) \ = \pi/2$

It follows, therefore, that the BER for the coherent detection of MSK signals is given by

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

Key takeaway

- In M-ary PSK using coherent detection, increasing M improves the bandwidth efficiency, but the Eb/N0 required for the idealized condition of “error-free” transmission moves away from the Shannon limit as M is increased.
- In M-ary FSK, as the number of frequency-shift levels M is increased—which is equivalent to increased channel-bandwidth requirement—the operating point moves closer to the Shannon limit.

3.4 Pulse Shaping to reduce Inter-channel and Inter-symbol Interference, some Issues in transmission and reception.

Pulse **shaping** is the process of changing the waveform of **transmitted** pulses. Its purpose is to make the **transmitted** signal better suited to its purpose or the communication channel, typically by limiting the effective bandwidth of the **transmission**.

In communications systems, two important requirements of a wireless communications channel demand the use of a pulse shaping filter. These requirements are:

1) Generating bandlimited channels, and

2) r\Reducing inter symbol interference (ISI) from multi-path signal reflections. Both requirements can be accomplished by a pulse shaping filter which is applied to each symbol.

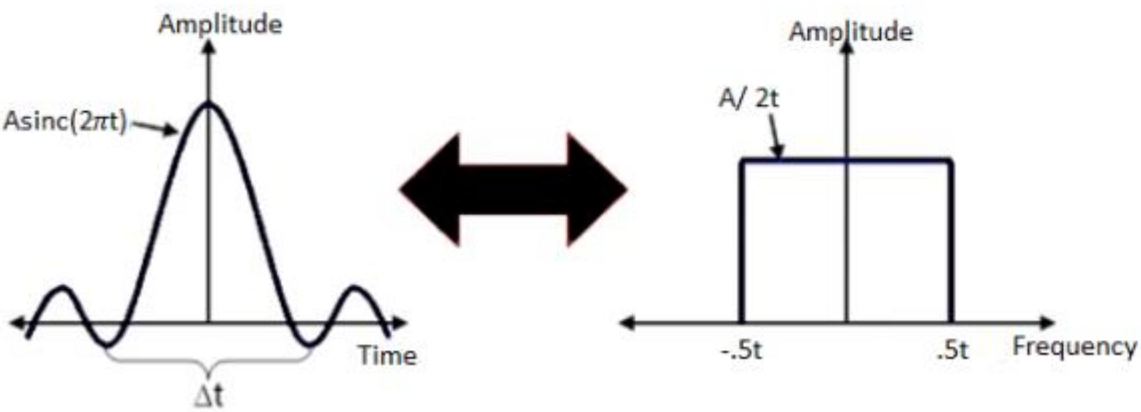


Fig 8: Time v/s Frequency Domain of Sinc Pulse

In fact, the sinc pulse, shown below, meets both of these requirements because it efficiently utilizes the frequency domain to utilize a smaller portion of the frequency domain, and because of the windowing affect that it has on each symbol period of a modulated signal. A sinc pulse is shown below along with an FFT spectrum of the given signal.

Inter-symbol Interference

This is a form of distortion of a signal, in which one or more symbols interfere with subsequent signals, causing noise or delivering a poor output.

Causes of ISI

The main causes of ISI are −

- Multi-path Propagation
- Non-linear frequency in channels

The ISI is unwanted and should be completely eliminated to get a clean output. The causes of ISI should also be resolved in order to lessen its effect.

To view ISI in a mathematical form present in the receiver output, we can consider the receiver output.

The receiving filter output y(t)y(t) is sampled at time ti=iTb (with **i** taking on integer values), yielding –

y(ti)= μ∑akp(iTb−kTb)

= μai+μ∑akp(iTb−kTb)

In the above equation, the first term μai is produced by the $\mathbf{i^{th}}$ transmitted bit.

The second term represents the residual effect of all other transmitted bits on the decoding of the $\mathbf{i^{th}}$ bit. This residual effect is called as **Inter Symbol Interference**.

In the absence of ISI, the output will be −

y(ti)=μai

This equation shows that the i^{th} bit transmitted is correctly reproduced. However, the presence of ISI introduces bit errors and distortions in the output.

While designing the transmitter or a receiver, it is important that you minimize the effects of ISI, so as to receive the output with the least possible error rate.

Key takeaway

The main causes of ISI are –

- Multi-path Propagation
- Non-linear frequency in channels

Eye patterns

An eye diagram or eye pattern is simply a graphical display of a serial data signal with respect to time that shows a pattern that resembles an eye.

The signal at the receiving end of the serial link is connected to an oscilloscope and the sweep rate is set so that one- or two-bit time periods (unit intervals or UI) are displayed. This causes bit periods to overlap and the eye pattern to form around the upper and lower signal levels and the rise and fall times. The eye pattern readily shows the rise and fall time lengthening and rounding as well as the horizontal jitter variation.

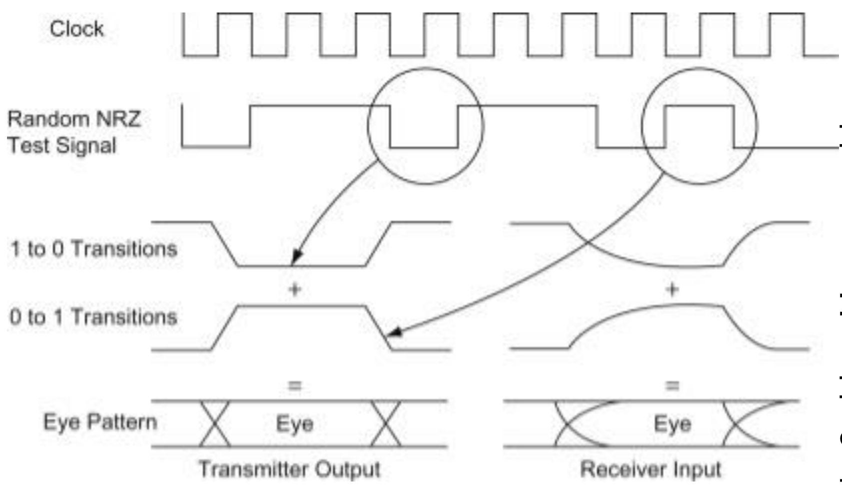


Fig 9: Eye diagram

Equalization

For reliable communication, we need to have a quality output.

The transmission losses of the channel and other factors affecting the quality of the signal. The most occurring loss, as we have discussed, is the ISI.

To make the signal free from ISI, and to ensure a maximum signal to noise ratio, we need to implement a method called **Equalization**.

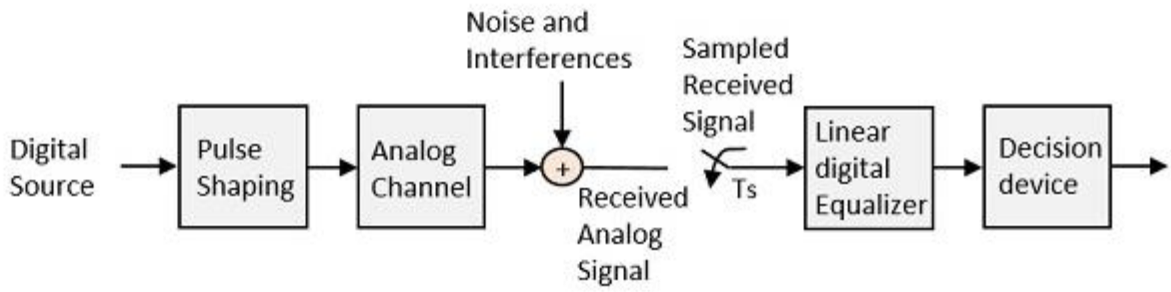


Fig 10: Equalization

The noise and interferences which are denoted in the figure, are likely to occur, during transmission. The regenerative repeater has an equalizer circuit, which compensates the transmission losses by shaping the circuit. The Equalizer is feasible to get implemented.

Key takeaway

To make the signal free from ISI, and to ensure a maximum signal to noise ratio, we need to implement a method called **Equalization**.

3.5 Orthogonal Frequency Division Multiplexing (OFDM)

- The Orthogonal Frequency Division Multiplexing (OFDM) transmission scheme is the optimum version of the multicarrier transmission scheme. In the past, as well as in the present, the OFDM is referred in the literature Multi-carrier, Multi-tone and Fourier Transform.
- In mid-1960 parallel data transmission and frequency multiplexing came. In high-speed digital communication OFMD is employed. OFMD has a

drawback of massive complex computation and high-speed memory which are no more problems now due to introduction of DSP and VLSI.

- The implementation of this technology is cost effective as FFT eliminates array of sinusoidal generators and coherent demodulation required in parallel data systems· The data to be transmitted is spread over large number of carriers. Each of them is then modulated at low rates. By choosing proper frequency between them the carriers are made orthogonal to each other.
- In OFDM the spectral overlapping among the sun carrier is allowed, at the receiver due to orthogonality the subcarriers can be separated. This provides better spectral efficiency and use to steep BPF is eliminated.
- The problems arising in single carrier scheme are eliminated in OFDM transmission system. It has the advantage of spreading out a frequency selective fade over many symbols. This effectively randomizes burst errors caused by fading or impulse interference so that instead of several adjacent symbols being completely destroyed, many symbols are only slightly distorted.
- Because of this reconstruction of majority of them even without forward error correction is possible. As the entire bandwidth is divided into many narrow bandwidths, the sub and become smaller than the coherent bandwidth of the channel which makes the frequency response over individual sub bands is relatively flat. Hence, equalization becomes much easier than in single carrier system and it can be avoided altogether if differential encoding is employed.
- The orthogonality of sub channels in OFDM can be maintained and individual subchannels can be completely separated by the FFT at the receiver when there are no inter symbol interference (ISI) and intercarrier interference (ICI) introduced by the transmission channel distortion.
- Since the spectra of an OFDM signal is not strictly band limited, linear distortions such as multipath propagation causes each subchannel to spread energy into the adjacent channels and consequently cause ISI.
- One way to prevent ISI is to create a cyclically extended guard interval, where each OFDM symbol is preceded by a periodic extension of the signal itself. When the guard interval is longer than the channel impulse response or multipath delay, the ISI can be eliminated.
- By using time and frequency diversity, OFDM provides a means to transmit data in a frequency selective channel. However, it does not suppress

fading itself. Depending on their position in the frequency domain, individual subchannels could be affected by fading.

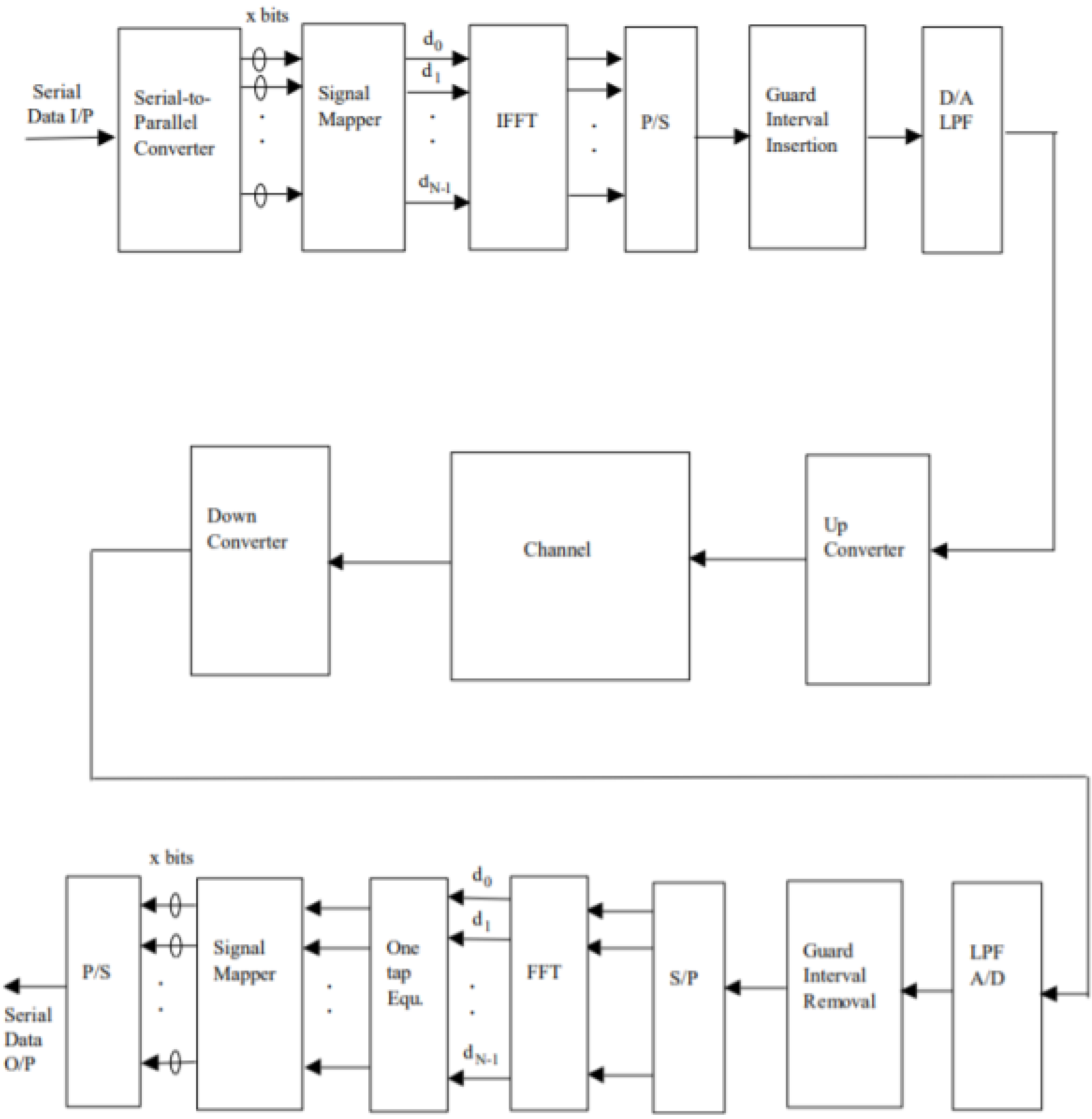


Fig 11: General OFDM System

The incoming signal is converted to parallel form by serial to parallel converter. After that it is grouped in x bits to form a complex number. These numbers are then modulated in a baseband manner by IFFT. Then through parallel to serial converted they are converted to serial data and transmitted. In order to avoid inter symbol interference a guard interval is inserted between the symbols. The discrete symbols are converted to analog and lowpass filtered for RF up-conversion. The receiver performs the inverse process of the transmitter. One tap equalizer is used to correct channel distortion. The tap coefficients of the filter are calculated based on channel information.

Key takeaway

The OFDM scheme differs from traditional FDM in the following interrelated ways:

- Multiple carriers (called subcarriers) carry the information stream
- The subcarriers are orthogonal to each other.
- A guard interval is added to each symbol to minimize the channel delay spread and inter-symbol interference.

3.6 Comparison of digital modulation systems

Modulation scheme	$s_1(t), s_2(t)$	**BW**	**Probability of error** (P_e)	**Comments**
Coherent ASK	$s_1(t) = A\cos\omega_c t$ $s_2(t) = 0$	$\cong 2R_b$	$Q\left(\sqrt{\frac{E_b}{2\eta}}\right)$ or $\frac{1}{2}\mathrm{erfc}\left(\sqrt{\frac{E_b}{4\eta}}\right)$	Rarely used
Non-Coherent ASK	$s_1(t) = A\cos\omega_c t$ $s_2(t) = 0$	$\cong 2R_b$	$\frac{1}{2}\exp\left(-\frac{E_b}{8\eta}\right)$	Rarely used
Coherent FSK	$s_1(t) = A\cos\omega_1 t$ $s_2(t) = A\cos\omega_2 t$	$> 2R_b$	$Q\left(\sqrt{\frac{2\times 0.6E_b}{\eta}}\right)$ or $\mathrm{erfc}\sqrt{\frac{0.6E_b}{\eta}}$	2.2 dB more power required than PSK Requires more Bandwidth.

Non-Coherent FSK	$s_1(t) = A\cos\omega_1 t$ $s_2(t) = A\cos\omega_2 t$	$> 2R_b$	$\frac{1}{2}\exp\left(-\frac{E_b}{4\eta}\right)$	No advantage over PSK, so seldom used
Coherent PSK	$s_1(t) = A\cos\omega_c t$ $s_2(t) = -A\cos\omega_c t$	$\cong 2R_b$	$Q\left(\sqrt{\frac{2E_b}{\eta}}\right)$ or $\frac{1}{2}\mathrm{erfc}\left(\sqrt{\frac{E_b}{\eta}}\right)$	3dB power advantage over ASK
Non-Coherent PSK DPSK	$s_1(t) = A\cos\omega_c t$ $s_2(t) = -A\cos\omega_c t$	$\cong 2R_b$	$\frac{1}{2}\exp\left(-\frac{E_b}{\eta}\right)$	DPSK is non-coherent version of PSK, but only little inferior than coherent PSK 1dB more power)

P_e – Probability of error, $\eta/2$ – two-sided noise PSD, T_b – bit duration, R_b is bit rate

Energy $(E_B) = \frac{A^2 T_b}{2}$, where A-carrier amplitude

Case Study

QAM for Wireless Communication

Another major development occurred in 1987 when Sundberg, Wong and Steele published a pair of papers. Considering QAM for voice transmission over Rayleigh fading channels, the first major paper considering QAM for mobile radio applications. In these papers, it was recognized that when a Gray code mapping scheme was used, some of the bits constituting a symbol had different error rates from other bits. Gray coding is a method of assigning bits to be transmitted to constellation points in an optimum manner. For the 16-level constellation two classes of bits occurred, for the 64-level three classes and so on. Efficient mapping schemes for pulse code modulated (PCM) speech coding was, discussed where the most significant bits (MSBs) were mapped onto the class with the highest integrity. A number of other schemes including variable threshold systems and weighted systems were also discussed. Simulation and theoretical results were compared and found to be in reasonable agreement. They used no carrier recovery, clock recovery or AGC, assuming these to be ideal, and came to the conclusion that channel coding and post-enhancement techniques would be required to achieve acceptable performance.

This work was continued, resulting in a publication in 1990 by Hanzo, Steele and Fortune, again considering QAM for mobile radio transmission, where again a theoretical argument was used to show that with a Gray encoded square constellation, the bits encoded onto a single symbol could be split into a number of subclasses, each subclass having a different average BER. The authors then showed that the difference in BER of these different subclasses could be reduced by constellation distortion at the cost of slightly increased total BER, but was best dealt with by using different error correction powers on the different 16- QAM subclasses. A 16 kbit/s sub-band speech coder was subjected to bit sensitivity analysis and the most sensitive bits identified were mapped onto the higher integrity 16-QAM subclasses, relegating the less sensitive speech bits to the lower integrity classes. Furthermore, different error correction coding powers were considered for each class of bits to optimise performance. Again, ideal clock and carrier recovery were used, although this time the problem of automatic gain control (AGC) was addressed. It was suggested that as bandwidth became increasingly congested in mobile radio, microcells would be introduced supplying the required high SNRs with the lack of bandwidth being an incentive to use QAM.

In the meantime, CNET were still continuing their study of QAM for point-to-point applications, and detailing an improved carrier recovery system using a novel combination of phase and frequency detectors which seemed promising. However, interest was now increasing in QAM for mobile radio usage and a paper was published in 1989 by J. Chuang of Bell Labs considering NLF-QAM for mobile radio and concluding that NLF offered slight improvements over raised cosine filtering when there was mild inter symbol interference (ISI).

A technique, known as the transparent tone in band method (TTIB) was proposed by McGeehan and Bateman from Bristol University, UK, which facilitated coherent detection of the square QAM scheme over fading channels and was shown to give good performance but at the cost of an increase in spectral occupancy. At an IEE colloquium on multilevel modulation techniques in March 1990 a number of papers were presented considering QAM for mobile radio and point-to-point applications. Matthews proposed the use of a pilot tone located in the centre of the frequency band for QAM transmissions over mobile channels. Huish discussed the use of QAM over fixed links, which was becoming increasingly widespread. Webb et al. Presented two papers describing the problems of square QAM constellations when used for mobile radio transmissions and introduced the star QAM constellation with its inherent robustness in fading channels.

Further QAM schemes for hostile fading channels characteristic of mobile telephony can be found in the following recent references. If Feher's previously mentioned NLA concept cannot be applied, then power-inefficient class A or AB linear amplification has to be used, which might become an impediment in lightweight, low-consumption handsets. However, the power consumption of the low-efficiency class A amplifier is less critical than that of the digital speech and channel codecs. In many applications 16- QAM, transmitting 4 bits per symbol reduces the signalling rate by a factor of 4 and hence mitigates channel dispersion, thereby removing the need for an equaliser, while the higher SNR demand can be compensated by diversity reception.

A further important research trend is hallmarked by Cavers' work targeted at pilot symbol assisted modulation (PSAM), where known pilot symbols are inserted in the information stream in order to allow the derivation of channel measurement information. The recovered received symbols are then used to linearly predict the channel's attenuation and phase. A range of advanced QAM modems have also been proposed by Japanese researchers doing cutting-edge research in the field.

References:

1. Bernard Sklar, Prabitra Kumar Ray, "Digital Communications Fundamentals and Applications", Pearson Education, 2nd Edition

2. Wayne Tomasi, "Electronic Communications System", Pearson Education, 5th Edition

3. A.B Carlson, P B Crully, J C Rutledge, "Communication Systems", Tata McGraw Hill Publication, 5th Edition

4. Simon Haykin, "Communication Systems", John Wiley & Sons, 4th Edition

5. Simon Haykin, "Digital Communication Systems", John Wiley & Sons, 4th Edition.

Unit - 4

Spread Spectrum Modulation

4.1 Use of Spread Spectrum

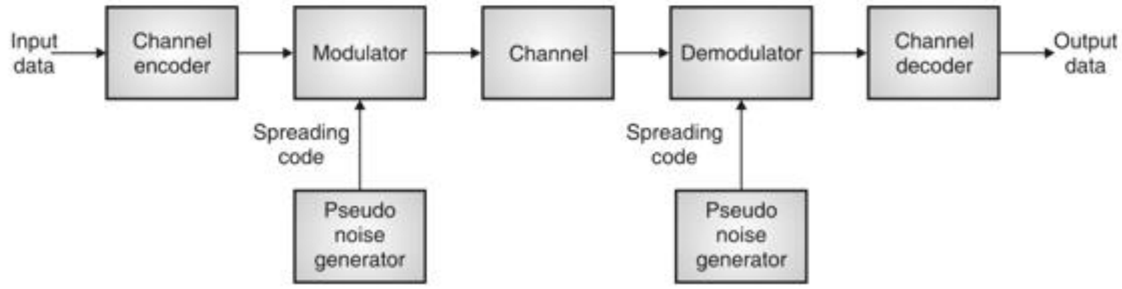


Fig.1: General spread spectrum structure

- Spread spectrum is specially used for wireless communication signal spreading. Transmitted signals frequency varies deliberately.
- Frequency hopping and direct sequence are two popular spread spectrums.
- In frequency hopping, signals are broadcast over any random series of frequency while in direct sequence each bit is in order of multiple bit of transmitting signal it uses chipping code.

Spread spectrum signals are used for

- Combating or suppressing the detrimental effects of interference due to jamming (Intentional interference). It can be used in military applications also.

- Accommodating multiple users to transmit messages simultaneously over the same channel bandwidth. This type of digital communication in which each user (transmitter-receiver pair) has a distinct PN code for transmitting over a common channel bandwidth is called as Code Division Multiple Access (CDMA) or Spread Spectrum Multiple Access (SSMA). This technique is popularly used in digital cellular communications.
- Reducing the unintentional interference arising from other users of the channel.
- Suppressing self-interference due to multipath propagation.
- Hiding a signal by transmitting it at low power and, thus, making it difficult for an unintended listener to detect in the presence of background noise. It is also called a Low Probability of Intercept (LPI) signal.
- Achieving message privacy in the presence of other listeners.
- Obtaining accurate range (time delay) and range rate (velocity) measurements in radar and navigation.

Key takeaway

There are two spread-spectrum approaches called Transmitted Reference (TR) and Stored Reference (SR).

(i) In a TR system, the transmitter sends two versions of truly random spreading signal (wideband carrier) – one modulated by data and the other unmodulated. The receiver used the unmodulated carrier as the reference signal for despreading (correlating) the data modulated carrier.

(ii) (ii)In a SR system, the spreading code signal is independently generated at both the transmitter and the receiver. Since the same code must be generated independently at two locations, the code sequence must be deterministic, even though it should appear random to unauthorized listeners. Such random appearing deterministic signals are called pseudo noise (PN), or pseudorandom signals.

(iii) Modern spread spectrum systems use Stored Reference (SR) approach which uses a Pseudo noise (PN) or pseudorandom code signal.

4.2 Direct sequence (DS) spread spectrum

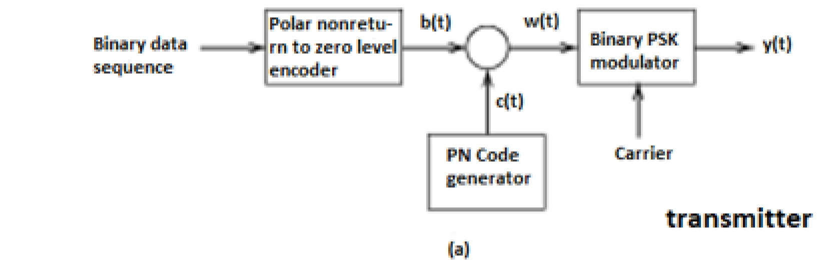
Direct Sequence Spread Spectrum (DSSS) using BPSK modulation, so the first reversing switch introduces 180 degree phase reversals according to a pseudo-random code, while the second introduces the same reversals to reconstitute the original, narrow-band signal. The output is a "recompressed" narrow-band signal.

DSSS is a spread spectrum modulation technique used for digital signal transmission over airwaves. It was originally developed for military use, and employed difficult-to-detect wideband signals to resist jamming attempts.

It is also being developed for commercial purposes in local and wireless networks.
The stream of information in DSSS is divided into small pieces, each associated with a frequency channel across spectrums. Data signals at transmission points are combined with a higher data rate bit sequence, which divides data based on a spreading ratio. The chipping code in a DSSS is a redundant bit pattern associated with each bit transmitted.

This helps to increase the signal's resistance to interference. If any bits are damaged during transmission, the original data can be recovered due to the redundancy of transmission.
The entire process is performed by multiplying a radio frequency carrier and a pseudo-noise (PN) digital signal. The PN code is modulated onto an information signal using several modulation techniques such as

quadrature phase-shift keying (QPSK), binary phase-shift keying (BPSK), etc. A doubly-balanced mixer then multiplies the PN modulated information signal and the RF carrier. Thus, the TF signal is replaced with a bandwidth signal that has a spectral equivalent of the noise signal. The demodulation process mixes or multiplies the PN modulated carrier wave with the incoming RF signal. The result produced is a signal with a maximum value when two signals are correlated. Such a signal is then sent to a BPSK demodulator. Although these signals appear to be noisy in the frequency domain, bandwidth provided by the PN code permits the signal power to drop below the noise threshold without any loss of information.



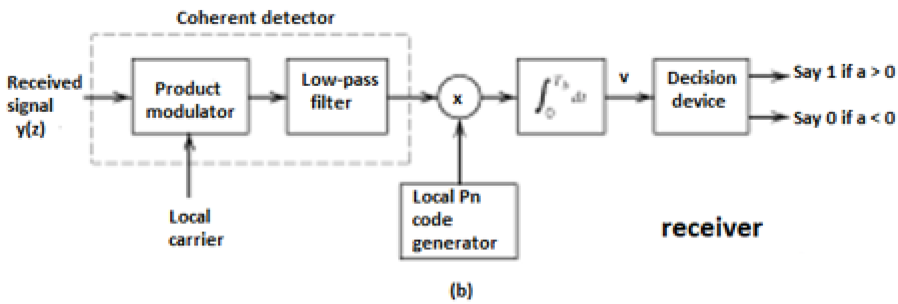


Fig 2(a) Transmitter (b) Receiver of DS-SS Technique

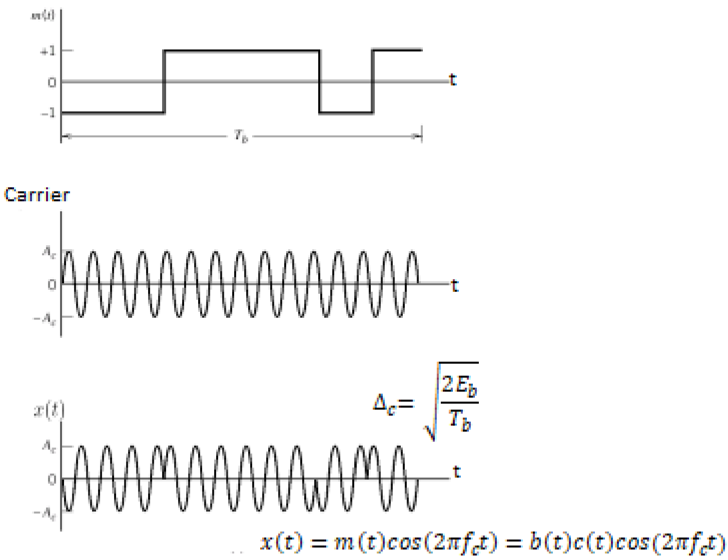


Fig 3 Wave forms for DSSS

Key takeaway

- In practice, the transmitter and receiver of Figures shown above are followed. In the transmitter spectrum spreading is performed prior to phase modulation. Also phase demodulation is done first and then despreading is done second, in the receiver.

- In the model of DS spread spectrum BPSK system used for analysis, the order of these two operations is interchanged. In the transmitter, BPSK is done first and spectrum spreading is done subsequently. Similarly, at the receiver also, spectrum de spreading is done first and then phase demodulation is done second.
- This is possible, because the spectrum spreading and BPSK are both linear operations.

4.3 Spread Spectrum and Code Division Multiple Access (CDMA)

The most important application of spread spectrum technique is the Digital cellular CDMA system. Here, we shall explain in detail about this CDMA digital cellular system based on Direct Sequence (DS) spread spectrum. This digital cellular communication system was proposed and developed by Qualcomm corporation. It has been standardized and designated as Interim Standard 95 (IS-95) by the Telecommunications Industry Association (TIA) for use in the 800 MHz and in the 1900 MHz frequency bands. The nominal bandwidth used for transmission from a base station to the mobile receivers (Forward link or channel) is 1.25 MHz. A separate channel, also with a bandwidth of 1.25 MHz is used for signal transmission from mobile receivers to a base station (reverse link or channel). The signals transmitted in both the forward and reverse links are DS Spread spectrum signals having a chip rate of 1.288 x 106 chips per second (Mchips/s).

A) Forward link or channel

The signal transmission from a base station to the mobile receivers is referred as the Forward link or channel. The figure below shows the block diagram of IS-95 forward link.

Source coding

The speech (source) coder is a code-excited linear predictive (CELP) coder. It generates data at the variable rates of 9600, 4800, 2400 and 1200bits/s. The data rate is a function of the speech activity of the user, in frame intervals of 20ms.

Channel coding

The data from the speech coder is encoded by a rate 1/2, constraint length K = 9 convolutional code. For lower speech activity, the output symbols from the convolutional encoder are repeated. If the data rate is 4800 bits/s, then the output symbols are repeated twice, so as to maintain a constant bit rate of 9600 bits/s.

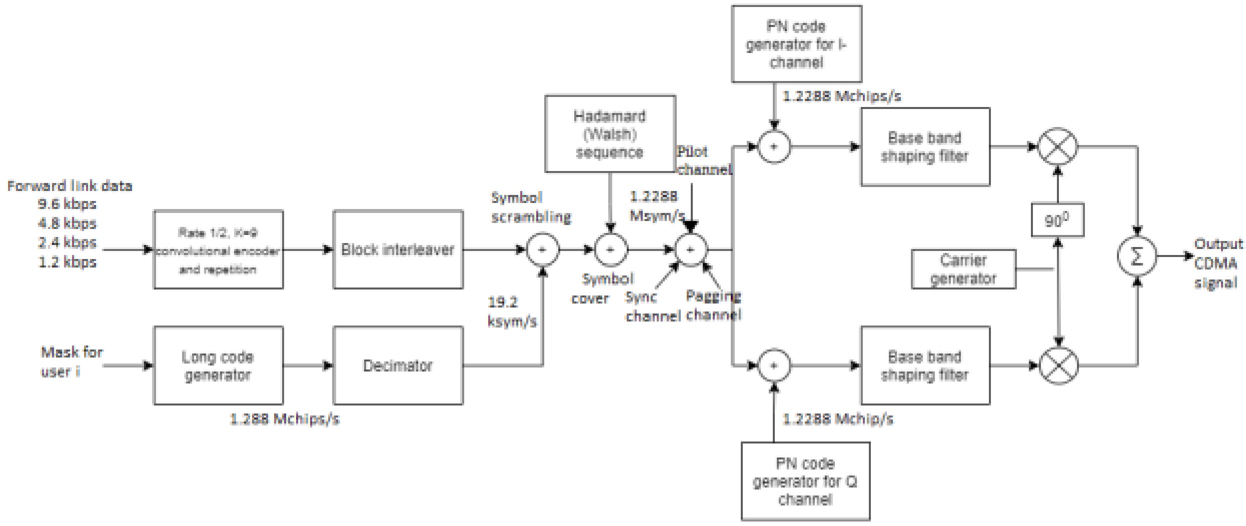


Fig 4 IS-95 Forward Link

Block interleaver:

The encoded bits for each frame are passed through a block interleaver. It is needed to overcome the effects of burst errors that may occur in transmission through the channel. The data bits at the output of the block interleaver occur at a rate of 19.2kbits/s.

Symbol scrambler

The data bits from the block interleaver are scrambled by multiplication with the output of a long code (period N=242 -1) generator. This generator is running at the chip rate of 1.288M chips/s, but the output is decimated by a factor of 64 to 19.2 kchips/s. The long code is used to uniquely identify a call of a mobile station on the forward and reverse links.

Hadamard Sequence

Each user of the channel is assigned a Hadamard (or Walsh) sequence of length 64. There are 64 orthogonal Hadamard sequences assigned to each base station. Thus, there are 64 channels available. One Hadamard sequence is used to transmit a pilot signal. The pilot signal is used for measuring the channel characteristics, including the signal strength and the carrier phase offset. Another Hadamard sequence is used for providing time synchronization. Another one sequence may be used for messaging (paging) service. Hence there are 61 channels left for allocation to different-users. The data sequence is now multiplied by the assigned Hadamard sequence of each user.

Modulator

The resulting binary sequence is now spread by multiplication with two PN sequences of length 215 and rate 1.2288 Mchips/s. This operation creates in-phase and quadrature signal components. Thus, the binary data signal is converted to a four-phase signal. Then, both I and Q signals are filtered by baseband spectral shaping filters. Different base stations are identified by different offsets of these PN sequences. The signals for all the 64 channels are transmitted synchronously. Finally, heterodyning of a carrier wave with BPSK modulation and QPSK spreading, is done. The summed output is the CDMA signal.

Mobile receiver

At the receiver, a RAKE demodulator is used to resolve the major multipath signal components. Then, they are phase-aligned and weighted according to their signal strength using the estimates of phase and signal strength derived from the pilot signal. These components are combined and passed to the Viterbi Soft decision decoder.

B) Reverse link or channel

The signal transmission from mobile transmitters to a base station is referred as the Reverse link or channel. The Figure below shows the block diagram of IS-95 reverse link.

Limitations

In the reverse link, the signals transmitted from various mobile transmitters to the base station are asynchronous. Hence, there is significantly more interference among users. Also, the mobile transmitters are usually battery operated and therefore, these transmissions are power limited. We have to design the reverse link in order to compensate for these two limitations.

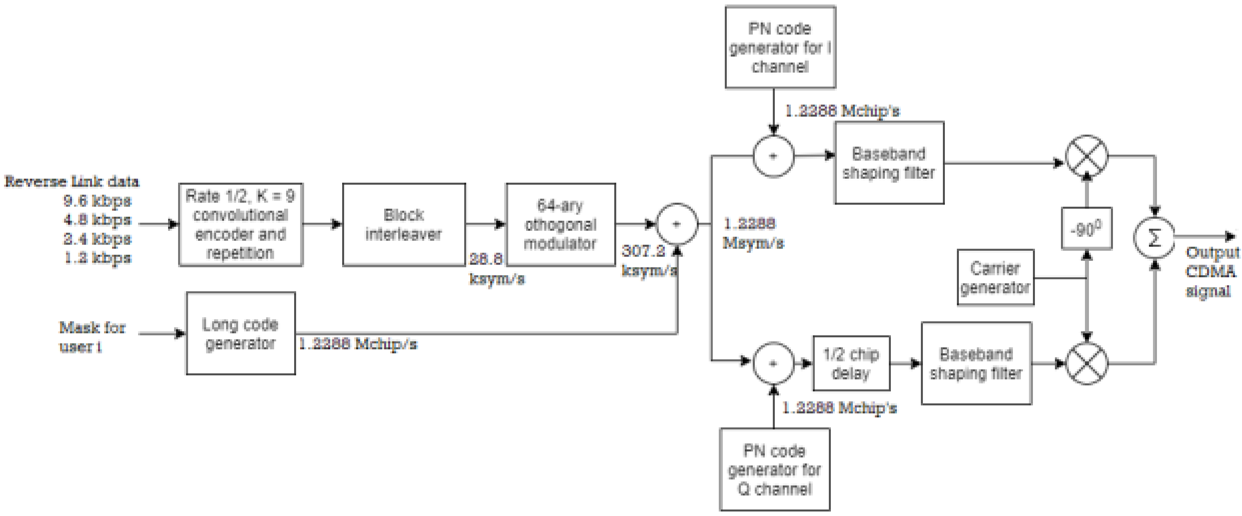


Fig 5 IS-95 Reverse link

Source coding

The reverse link data may also be at variable rates of 9600, 4800, 2400 and 1200 bits/s. The data rate is a function of the speech activity of the user, in frame intervals of 20ms.

Channel coding

The data from the speech coder is encoded by a rate 1/3, constraint length K=9 convolutional code. This coder has higher coding gain in a fading channel. This compensates for the above-mentioned limitations. For lower speech activity, the output bits form the convolutional encoder are repeated either two, or four, or eight times.

Block interleaver

The encoded bits for each frame are passed through a block interleaver. It is needed to overcome the effects of burst errors. For each 20ms frame, the 576 encoded bits are block-interleaved. However, the coded bit rate is 28.2 kbits/s.

Hadamard sequence

The data is modulated using an M=64 orthogonal signal set using Hadamard sequences of length 64. Thus, a 6-bit block of data is mapped into one of the 64 Hadamard sequences. The result is a bit (or chip) rate of 307.2 kbits/s at the output of the modulator.

Symbol scrambler

To reduce interference to other users, the time position of the transmitted code symbol repetitions is randomized. Hence, at the lower speech activity, consecutive bursts do not occur evenly spaced in time. The signal is also spread by the output of the long code generator running at a rate of 1.2288 Mchips/s. This is done for channelization (addressing), for privacy, scrambling, and spreading.

Modulator

The resulting 1.2288 Mchips/s binary sequence at the output of the multiplier is then further multiplied by two PN sequences of length N=215 with rate 1.2288

Mchips/s. This operation creates in phase and quadrature signals. Both the I and Q signals are filtered by baseband spectral shaping filters. The Q-channel signal is delayed in time by one half PN chip time relative to the I-channel signal prior to the base band filter. The signal at the output of the two baseband filters is an offset QPSK signal. Finally, the filtered signals are passed to quadrature mixers. The summed output is the CDMA signal.

Base station Receiver

The base station dedicates a separate channel in order to receive the transmissions of each active user in the cell. Although the chips are transmitted as an offset QPSK signal, the demodulator at the base station receiver employs noncoherent demodulation. A fast Hadamard transform is used to reduce the computational complexity in the demodulation process. The output of the demodulator is then fed to the Viterbi detector, whose output is used to synthesize the speech signal.

Key takeaway

Approach	SDMA	TDMA	FDMA	CDMA
Idea	Segment space into cells/sectors	Segment sending time into disjoint time-slots, demand driven or fixed patters	Segment the frequency band into disjoint sub-bands	Spread the spectrum using orthogonal codes
Terminals	Only one terminal can be active in one cell/one sector	All terminals are active for short periods of time on the same frequency	Every terminal has its own frequency. Uninterrupted	All terminals can be active at the same place at the same moment

Approach	SDMA	TDMA	FDMA	CDMA
				uninterrupted.
Signal separation	Cell structure, directed antennas	Synchronization in the time domain	Filtering in the frequency domain	Code plus special receivers
Advantages	Very simple, increases capacity per km2	Established, fully digital, flexible	Simple, established robust	Flexible, less frequency planning needed, soft handover
Disadvantages	Inflexible, antennas typically fixed	Guard space needed (multipath propagation), synchronization difficult	Inflexible, frequencies are a scarce resource	Complex receivers, needs more complicated power control for senders
Comment	Only in combination with TDMA, FDMA or CDMA useful	Standard in fixed networks, together with FDMA/SDMA used in many mobile networks	Typically combined with TDMA (frequency hopping patterns) and SDMA (frequency reuse)	Still faces some problems, higher complexity, lowered expectations: will be integrated with TDMA/FDMA

4.4 Ranging Using DS Spread Spectrum

The second widely used method is the direct-sequence spread spectrum ranging technique, as shown in figure below. This technique has been used in GPS C/A code and P code.

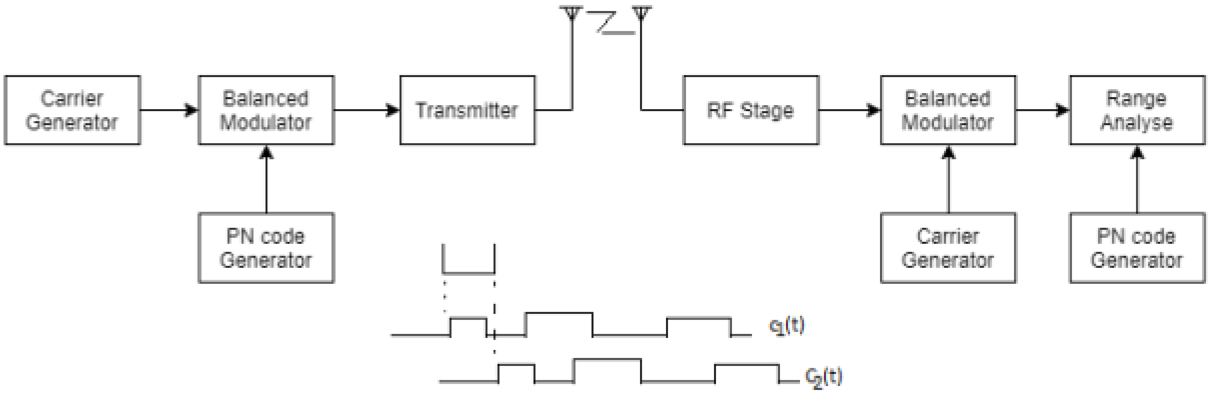


Fig:6 DS/SS System

The difference between DS/SS and CW ranging systems is that with DS/SS the carrier is phase-shift modulated by a pseudo-random signal. At the receiver, the received signal is demodulated. By measuring the number of chips of the code delay between the signals being transmitted and received, one can determine uniquely the range from the transmitter to the receiver.

The basic principle of PN ranging is the following. Two identical PN codes are involved. One code propagates between the transmitter and the receiver. The other code is maintained in the receiver. Since a PN code is completely deterministic, the sequence of ones and zeros in the code is completely known. Both codes are initiated in synchronism at a particular time, t_o. The code that propagates between the transmitter and the receiver is received by the receiver, delayed by the propagation time. The receiver then shifts the local version of the PN code, keeping track of the amount of time delay until its sequence exactly matches the sequence being received from the transmitter. When the receiver observes that the two versions of the same PN code are "matched" or synchronized at the observation time, t_r, $^{\Delta}T = t_r - t_o$ is the range delay time.

The difference between DS/SS and CW ranging systems is that with DS/SS the carrier is phase-shift modulated by a pseudo-random signal. At the receiver, the received signal is demodulated. By measuring the number of chips of the code delay between the signals being transmitted and received, one can determine uniquely the range from the transmitter to the receiver. The basic principle of PN ranging is the following. Two identical PN codes are involved. One code propagates between the transmitter and the receiver. The other code is maintained

in the receiver. Since a PN code is completely deterministic, the sequence of ones and zeros in the code is completely known.

Both codes are initiated in synchronism at a particular time, to. The code that propagates between the transmitter and the receiver is received by the receiver, delayed by the propagation time. The receiver then shifts the local version of the PN code, keeping track of the amount of time delay until its sequence exactly matches the sequence being received from the transmitter. When the receiver observes that the two versions of the same PN code are "matched" or synchronized at the observation time, t_r, $\Delta T = t_r - t_o$ is the range delay time.

The matching of the local code with the received code is essentially matching the zero-one sequence and then "exactly" matching the leading and trailing edges of each chip with the sequence. Any error in time delay translates into an error in determining the range separating the transmitter and receiver. It is clear that the smaller the chip duration, the smaller the code matching error. The spread spectrum technique has its advantage in that its phase is easily resolved, as shown in figure below. By matching received PN code and the local PN code, a high time resolution can be achieved. If the repetition cycle of the PN code is long enough, the range between transmitter and the receiver cm be determined without ambiguity. The spread spectrum technique has the ability to discriminate interference signals.

A sharp correlation peak occurs when local PN code exactly matches the received PN code. The later correlation peaks due to multipath propagation are rejected. In addition, DS/SS systems minimize interference to other services in the same frequency band by transmitting at low power densities. For a spread spectrum system to operate properly, the received code must be synchronized with the transmitter code to within 1 chip. Range measurement is made by counting chips of offset or fractions of chips and is therefore a discrete measurement. In practice measurements are commonly made to within a fraction of a chip period. The highest-resolution spread spectrum systems known can measure range to approximately 1/1000th of a chip period.

Therefore, DS/SS ranging technique has its limitations when used in high-resolution ranging applications. The range resolution of a DS/SS ranging system depends on the chosen chip rate and accuracy of synchronization of the local PN code with the received PN code. For example, to design a DS/SS ranging system with range resolution of 5 centimetres, and assuming the accuracy of PN code synchronization of the transmitter and the receiver can discriminate to one-hundredth of a chip. The chip wavelength should be less than 5 meters (= 5 x 10^{-2} x 100), which gives a chip rate of:

$R_c = c/\lambda_c$= 2.997925x108/5= 0.6x108 cps

Where, c= speed of light

R_c = chip rate

λ_c= wavelength of chip

Obviously, the above result is impractical for the applications with lower carrier frequencies. Increasing the resolution of a DS/SS system requires increasing the chip rate or increasing synchronization accuracy of the system or both. Unfortunately, both methods are often limited by other factors in practical applications. Another limitation with DÇ/SS ranging systems is that only the phase resolvability of the modulating pseudorandom signals is used. The carrier phase information, which cm provide higher range resolution, is discarded.

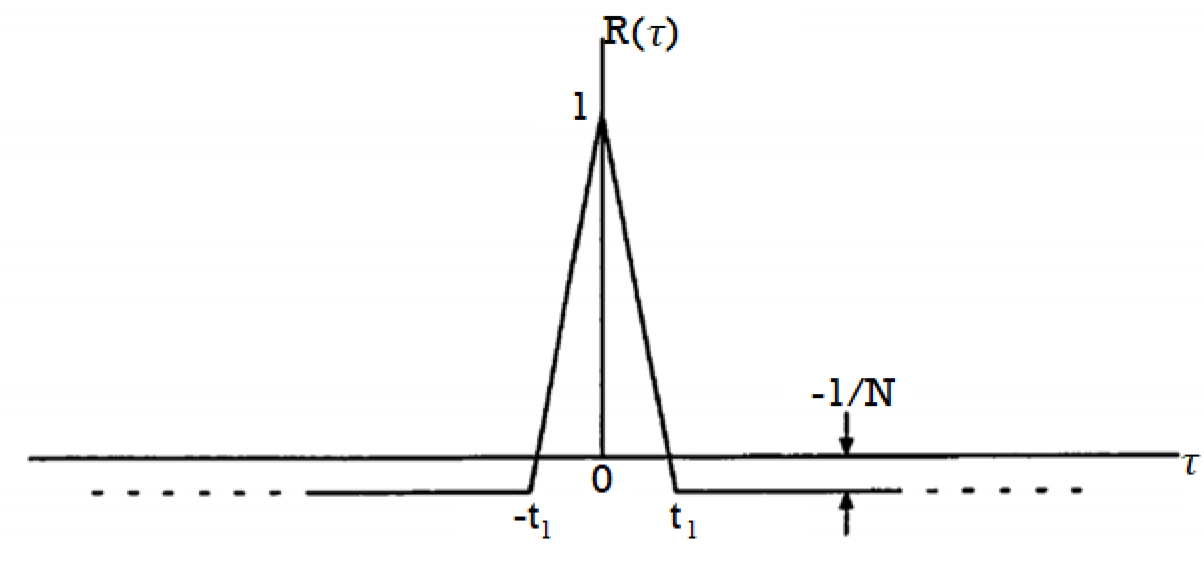


Fig:7 DS/SS Correlation; N - Length of PN sequence, t_1 - Chip duration

4.5 Frequency Hopping (FH) Spread Spectrum

This is frequency hopping technique, where the users are made to change the frequencies of usage, from one to another in a specified time interval, hence called as **frequency hopping**. For example, a frequency was allotted to sender 1 for a particular period of time. Now, after a while, sender 1 hops to the other frequency and sender 2 uses the first frequency, which was previously used by sender 1. This is called as **frequency reuse**.

The frequencies of the data are hopped from one to another in order to provide a secure transmission. The amount of time spent on each frequency hop is called as **Dwell time**.

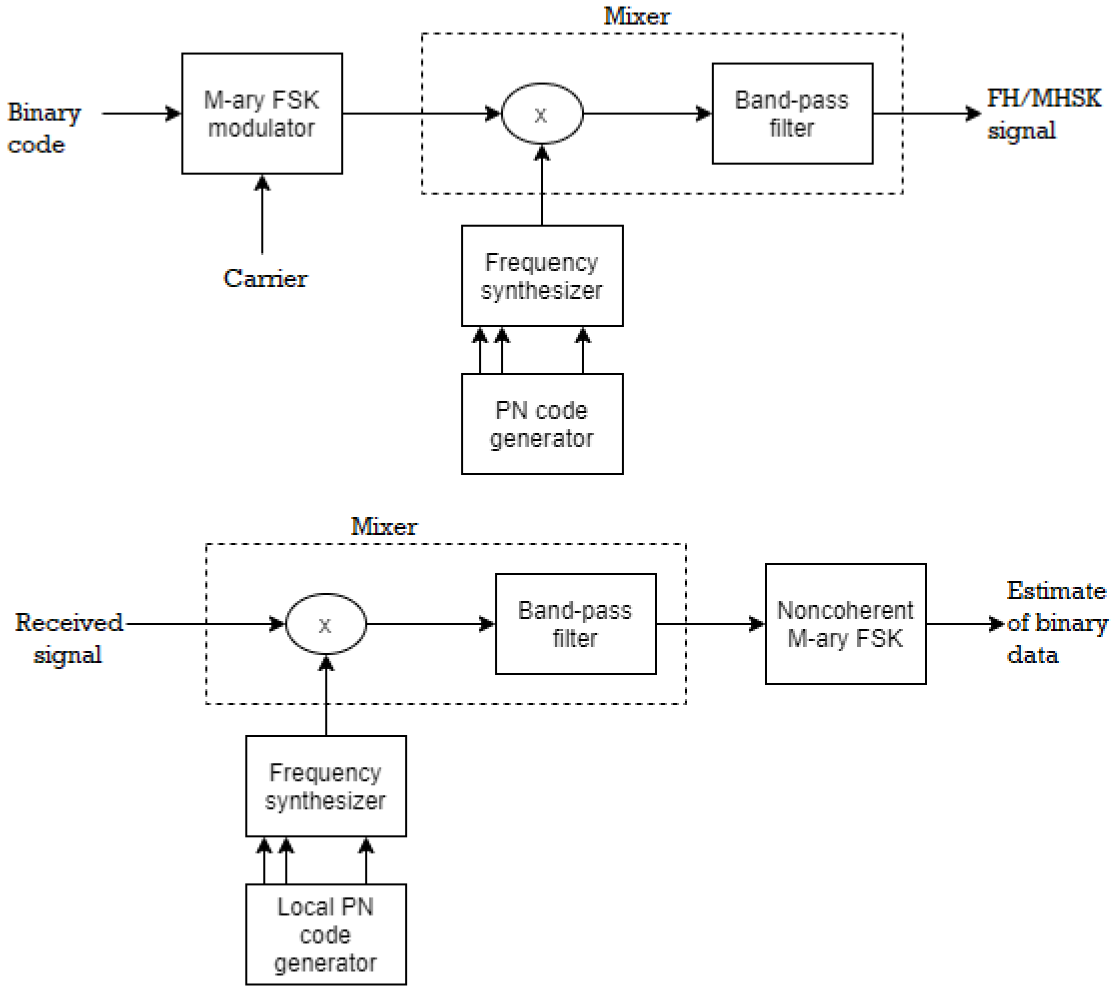


Fig 8 FHSS

Transmitter and Receiver

Advantages of FH-SS system:

1. The processing gain PG is higher than that of DS-SS system

2. Synchronization is not greatly dependent on the distance.

3. The serial search system with FH-SS needs shorter time for acquisition.

Disadvantages of FH-SS system:

1. The bandwidth of FH-SS system is too large (in GHz).

2. Complex and expensive digital frequency synthesizers are required.

Applications of FHSS system:

1) CDMA systems based on FH spread spectrum signals are particularly attractive for mobile communication.

2) Wireless local area networks (WLAN) standard for Wi-Fi.

3) Wireless Personal area network (WPAN) standard for Bluetooth.

Comparison between FHSS and DSSS

FHSS	**DSSS**
Multiple frequencies are used	Single frequency is used
Hard to find the user's frequency at any instant of time	User frequency, once allotted is always the same
Frequency reuse is allowed	Frequency reuse is not allowed

Sender need not wait	Sender has to wait if the spectrum is busy
Power strength of the signal is high	Power strength of the signal is low
Stronger and penetrates through the obstacles	It is weaker compared to FHSS
It is never affected by interference	It can be affected by interference
It is cheaper	It is expensive
This is the commonly used technique	This technique is not frequently used

4.6 Pseudorandom (PN) Sequences: Generation and Characteristics

- Pseudo-Noise (PN) also known Pseudo Random Binary Sequence (PRBS).
- A Pseudo-Noise code (PN-code) or Pseudo Random Noise Code (PRN code) is a spectrum which generated deterministically by random sequence.
- PN sequence is random occurrence of 0's and 1's bit stream.
- Directly sequence spread spectrum (DS-SS) system is most popular sequence in DS-SS system bits of PN sequence is known as chips and inverse of its period is known as chip rate.

In frequency hopping spread spectrum (FH-SS) sequence, channel number are pseudo random sequences and hop rate are inverse of its period.

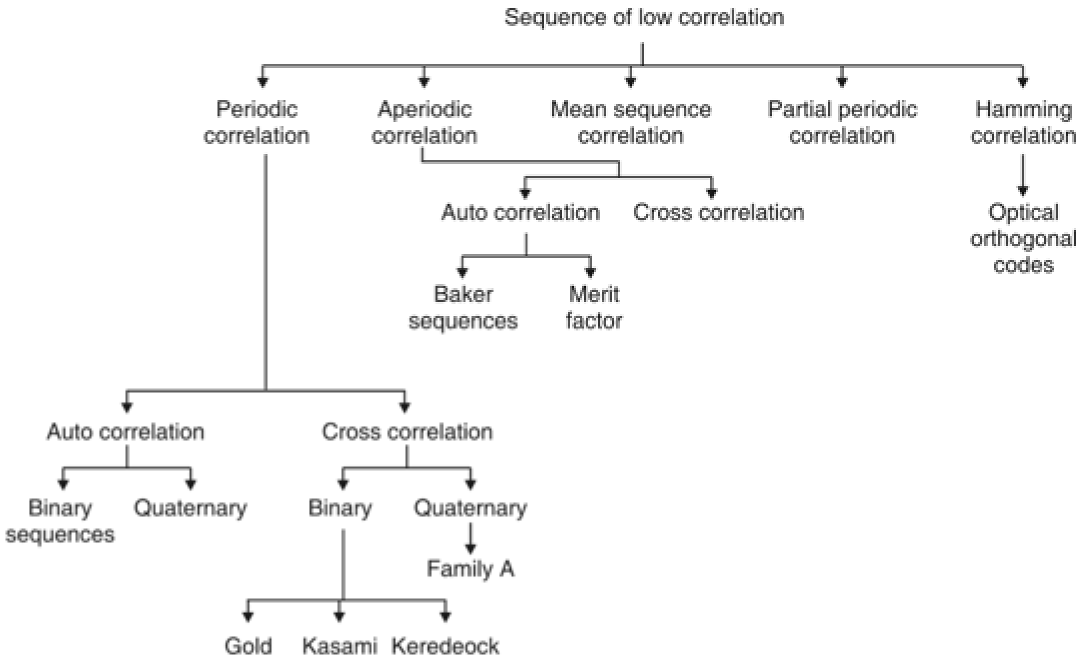


Fig.9: Overview of PN sequence

Properties

(i) Balance property:

Total no. Of 1's is more than no. Of 0's in maximum length sequence.

(ii) Run property 1's and 0's stream shows length sequence, every fraction relates some meaning.

Rum	Length
1/2	1
1/4	2
1/8	3

(iii) Correlation property:

If compared sequences are found similar then it is auto correlation.

If compared sequences are found mismatched then it is cross correlation.

Let (K) and y (K) are two sequences then correlation R (m) will be:

R (m)xy= x (k) y (k + m)

Correlation R (m) in pattern of digital bit sequence will be:

R (m) =

$y1 = P1 \oplus q1$

$y1 = 0$ if $P1 \neq q1$

$y1 = 1$ if $P1 = q1$

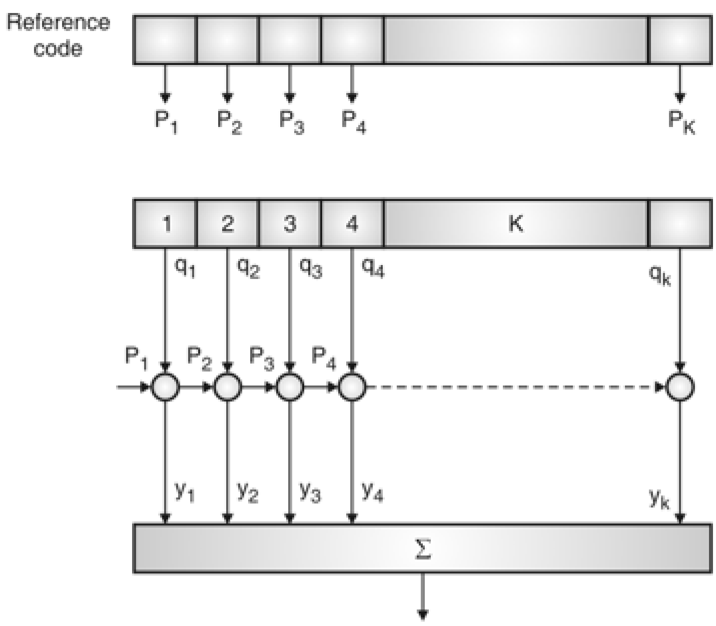


Fig 10: Correlator

Fig. Above shows Pi is a sequence which shifts through K bit shift register. K is length of correlate. Output achieved by K XNOR gate after comparison.

(iv) Shift and add:

By using X-OR gate, shift sequence modulo-2 added to upshifted sequence.

PN Sequence Generation

The class of sequences used in spread spectrum communications is usually periodic in that a sequence of 1s and 0s repeats itself exactly with a known period. The maximum length sequence, a type of cyclic code represents a commonly used periodic PN sequence. The maximum length sequences or PN sequences can be generated easily using shift register circuits with feedback from one or more stages. A PN sequence generator using a 3-stage shift register is shown in Figure below.

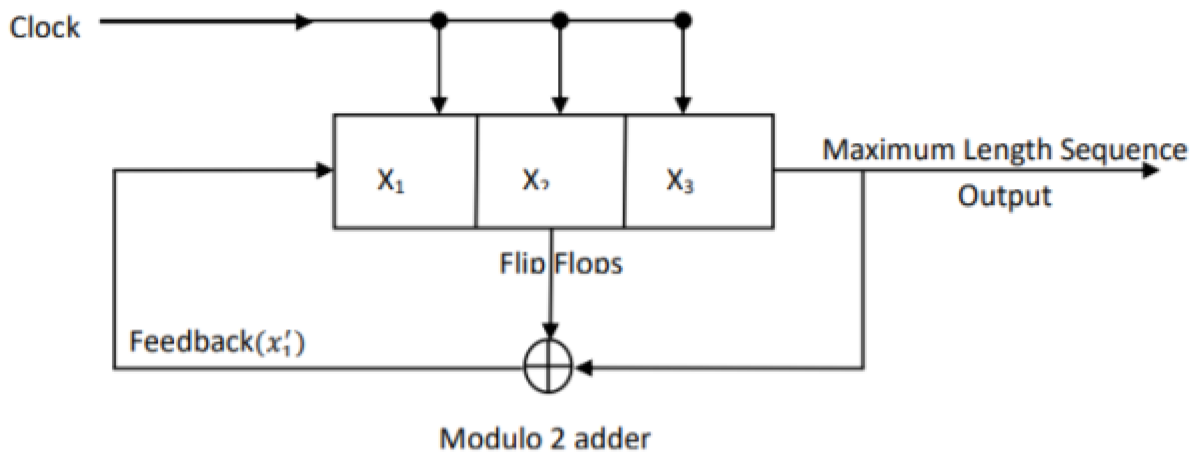


Fig 11 PN sequence or Maximum Length Sequence Generator

The 3-stage shift register consists of 3 flipflops regulated by a single timing clock. At each pulse of the clock, the state of each flipflop is shifted to the next one. The feedback function is obtained by using modulo-2 addition of the outputs of flipflops x_2 and x_3. The feedback term is applied to the input of the first flipflop x_1. The maximum length sequence output is obtained by noting the contents of flipflop x_3 at each clock pulse. The maximum-length sequence so generated is always periodic with a period of $N = 2^{m}-1$

Where m is the length of the shift register.

Here, m = 3 and so N = 23 -1 = 7. For the PN sequence generator of Figure above, if we assume that the shift register contents are initially 111, then with each clocking pulse, the contents will change as shown in the following table

Shifts	$x'_1 = X_2 \oplus X_3$	Shift register contents		
		X_1	X_2	X_3
0		1	1	1
1	$1 \oplus 1 = 0$	0	1	1
2	$1 \oplus 1 = 0$	0	0	1
3	$0 \oplus 1 = 1$	1	0	0
4	$0 \oplus 0 = 0$	0	1	0
5	$1 \oplus 0 = 1$	1	0	1
6	$0 \oplus 1 = 1$	1	1	0
7	$1 \oplus 0 = 1$	1	1	1

Hence for one period, the output PN sequence is 1 1 1 0 0 1 0, with a sequence length of 7. Thereafter, the sequence will be repeated.

Key takeaway

- The length of the PN sequence is N = 2m-1, where m is the number of shift register stages.
- The PN sequence repeats itself after every 'N' clock cycles.
- The PN sequence is an NRZ type pulse signal with logic 1 represented by + 1 and logic 0 represented by -1, as shown in Figure

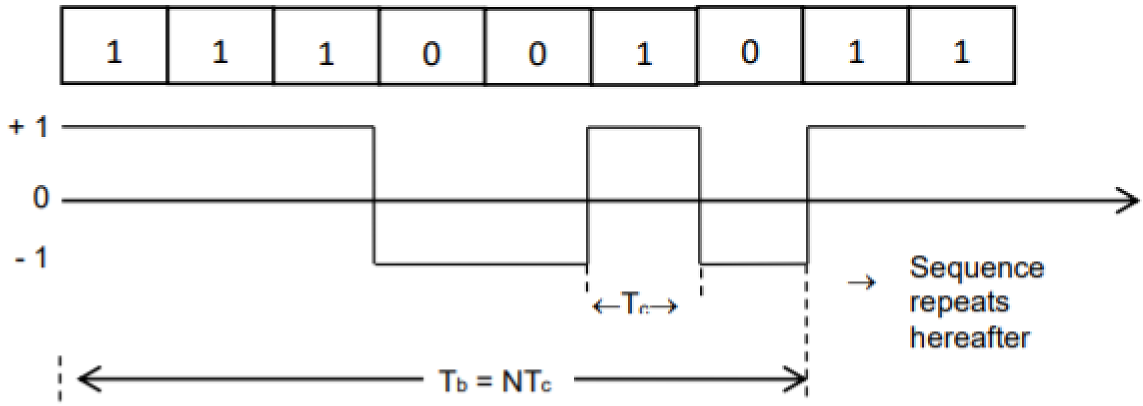


Fig 12 PN Sequence waveform

- The duration of every bit is known as the chip duration Tc. The chip rate Rc is defined as the number of bits (chips) per second. Tc = $1/Rc$ (or) Rc = $1\ Tc$
- The period of the PN sequence is Tb = NTc
- The autocorrelation function R(τ) is a periodic function of time and it is a two valued function.

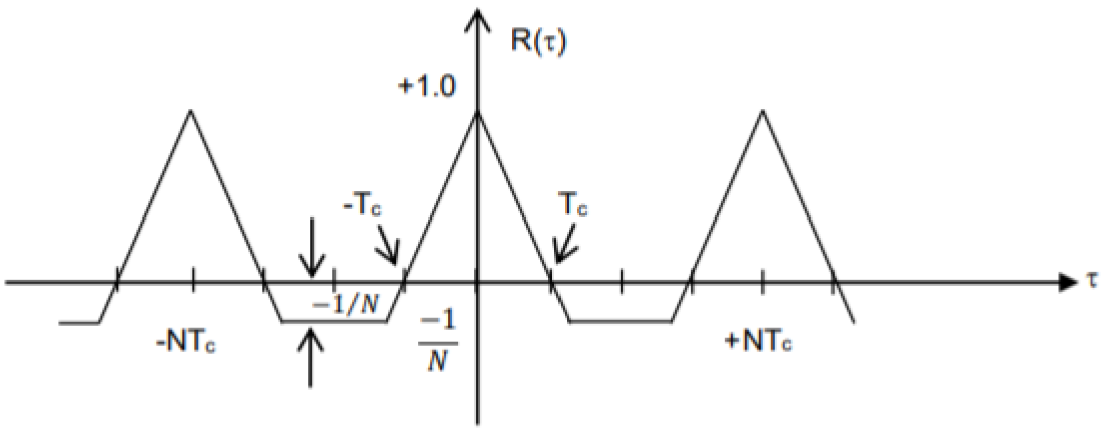


Fig 13

Autocorrelation function of a PN sequence

Example

Q1) A four stage shift register with feedback connections taken from the outputs of stages 4 and 1 through a modulo– 2adder, is used for PN sequence generation. Assuming the initial contents of the shift register to be 0100, determine the output sequence. What is the length of the sequence?

A1) The PN sequence generator is shown in Figure

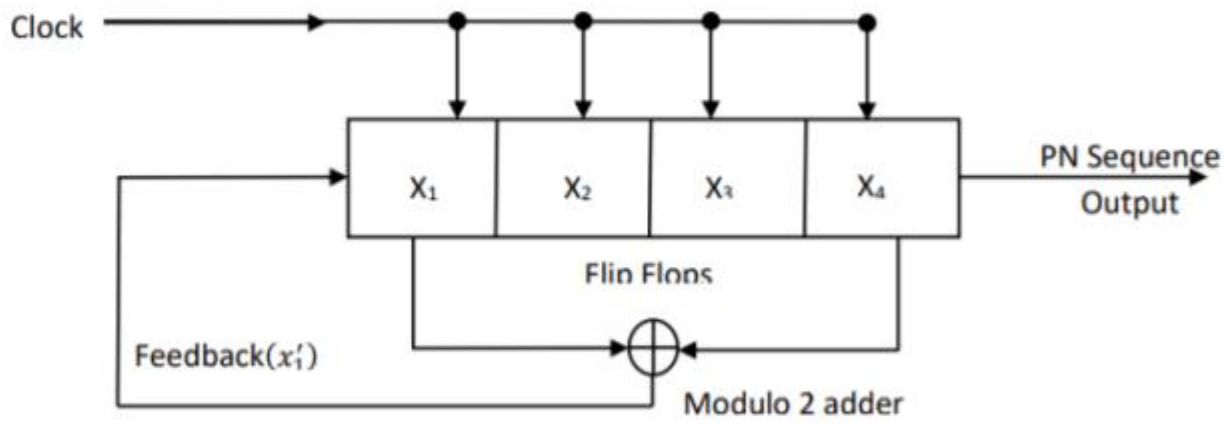


Fig 14 PN Sequence Generator

If the initial contents of the shift register are 0100, then with each clocking pulse, the contents will change as shown in the following table

Shifts	Feedback $x_1' = X_4 \oplus X_1$	Shift register contents X_1	X_2	X_3	X_4
0		0	1	0	0
1	$1 \oplus 1 = 0$	0	0	1	0
2	$1 \oplus 1 = 0$	0	0	0	1
3	$0 \oplus 1 = 1$	1	0	0	0
4	$0 \oplus 0 = 0$	1	1	0	0
5	$1 \oplus 0 = 1$	1	1	1	0
6	$0 \oplus 1 = 1$	1	1	1	1
7	$1 \oplus 0 = 1$	0	1	1	1

8	1 $\oplus 0 = 1$	1	0	1	1
9	1 $\oplus 1 = 0$	0	1	0	1
10	1 $\oplus 0 = 1$	1	0	1	0
11	0 $\oplus$ 1=1	1	1	0	1
12	1 $\oplus$ 1=0	0	1	1	0
13	0 $\oplus$ 0=0	0	0	1	1
14	1 $\oplus$ 0=1	1	0	0	1
15	1 $\oplus$ 1=0	0	1	0	0

The output PN sequence is 0 0 1 0 0 0 1 1 1 1 0 1 0 1 1 After 15 shifting's, the initial contents of the shift registers are once again obtained. For further shifting's, the same cycle of events will repeat. Thus, the length of one period of the PN sequence is, $N = 2^m - 1 = 24 - 1 = 15$. Hence the sequence is a maximal length sequence.

Q2) A spread spectrum communication system is characterised by the following parameters Information bit duration, Tb = 4.095 ms PN chip duration, Tc = 1μs Determine the processing gain and jamming margin if Eb/No =10 and the average probability of error, Pe=0.5 x 10^{-5}.

A2)

i) the processing gain, PG $= \frac{T_b}{T_c} = 4.095\frac{ms}{1\mu s}$

$$PG = \frac{4.095 \times 10^{-3}}{1 \times 10^{-6}} = 4.095 \times 10^{3} = 4095$$

Hence PG=4095. Since PG=Spread factor, N, we have PG=N=4095

Ii) The jamming margin is

$$(\text{Jamming margin})_{dB} = (\text{Processing gain})_{dB}$$
$$= 10\log_{10}\left(\frac{E_b}{N_0}\right)_{min} \quad 10\log_{10}(4095) - 10\log_{10} 10$$
$$= 36.1225 - 10 = 26.1225$$
$$(\text{Jamming margin})_{dB} = 26.1225$$

4.7 Synchronization in Spread Spectrum Systems

Need for Synchronization

The process in which the locally generated carrier at the receiver must be in frequency and phase synchronism with the carrier at the transmitter is called synchronization. In spread spectrum communication systems, there should be perfect alignment between the transmitted and received PN codes, for satisfactory operation. Because

(i) Carrier frequency as well as the PN clock may drift with time.

(ii) If there is relative motion between the transmitter and receiver, as in the case of mobile and satellite spread spectrum systems, the carrier and PN clock will suffer Doppler frequency shift.

Hence, synchronization of the PN sequence of the receiver with that of the transmitter is essential.

Acquisition: Acquisition schemes can be classified into three types. They are

1) Serial search acquisition

2) Parallel search acquisition

3) Sequential search acquisition

1. Serial search acquisition:

A) DS Spread spectrum systems: Figure below shows the serial search scheme for Direct Sequence spread spectrum systems. There is always an initial timing uncertainty between the receiver and the transmitter. Let us suppose that the transmitter has N chips and the chip duration is Tc. If initial synchronization is to take place in the presence of additive noise and other interference, it is necessary to dwell for Td=NTc in order to test synchronism at each time instant. We search over the time uncertainty interval in(coarse) time steps of 1/2 Tc.

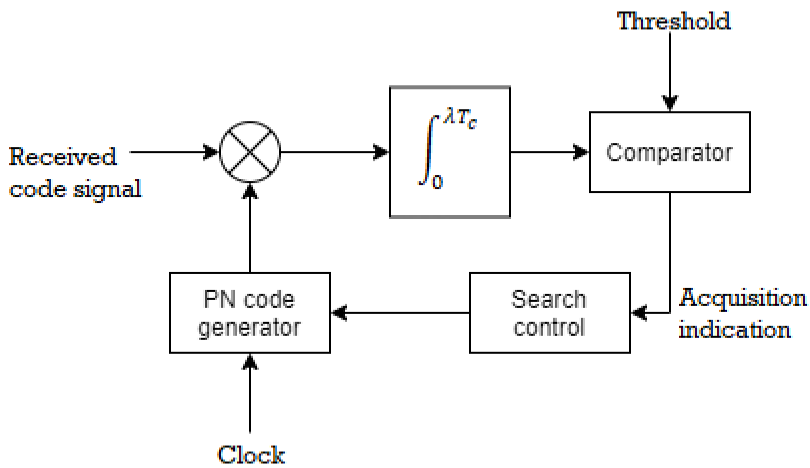


Fig 15 Direct Sequence spread spectrum systems – Serial Search Acquisition

The locally generated PN signal is correlated with the incoming PN signal. At fixed search intervals of N T_c (search dwell time), the output signal is compared to a preset threshold. If the output is below the threshold, the locally generated code signal is advanced in time by 1/2Tc seconds. The correlation process is repeated again. These operations are performed until a signal is detected or the threshold is exceeded. Then the PN code is assumed to have been acquired. Thus, if initially the misalignment between the two codes was n chips, the total time taken for acquisition is given by $T_{acq} = 2nNT_c$ seconds

B) FH spread spectrum systems

Figure below shows the serial search scheme for frequency hopping spread spectrum systems. Here the non-coherent matched filter consists of a mixer followed by a bandpass filter (BPF) and a square law envelope detector. The PN code generator controls the frequency hopper. Acquisition is accomplished when the local hopping is aligned with that of the received signal. Let fi be the frequency of the frequency synthesizer at the transmitter. Suppose fj be the frequency of the signal produced by the frequency synthesizer in the acquisition circuit of the receiver. If fi ≠ fj, then only a small voltage less than the threshold will be produced at the output of BPF. At a later instant of time during searching, if fi = fj, then a large voltage exceeding the threshold will be produced at the output of BPF. This indicates the alignment of local hopping with that of the received signal.

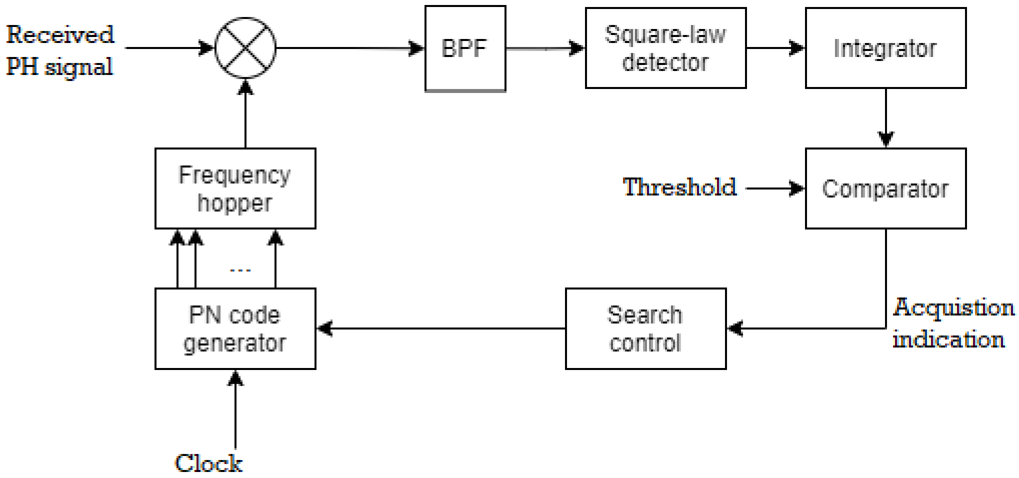


Fig 16

Frequency hopping serial search acquisition

2. Parallel search acquisition

The parallel search acquisition scheme introduces some degree of parallelism by having two or more correlators operating in parallel. They will search over nonoverlapping time slots. In this scheme, the search time is reduced at the expense of a more complex and costly implementation.

3. Sequential search acquisition

In this scheme, the dwell time at each delay in the search process is made variable by employing a correlator with a variable integration period whose (biased) output is compared with two thresholds. Hence the sequential search method results in a more efficient search in the sense that the average search time is minimised.

Tracking

Once the signal is acquired, the initial search process is stopped and fine synchronization and tracking begins. The tracking maintains the PN Code generator at the receiver in synchronism with the incoming signal. Tracking includes both fine chip synchronization and, for coherent demodulation, carrier phase tracking.

A) DS Spread spectrum system:

The commonly used tracking loop for a Direct sequence spectrum signal is the Delay-locked loop (DLL) as shown in the Figure

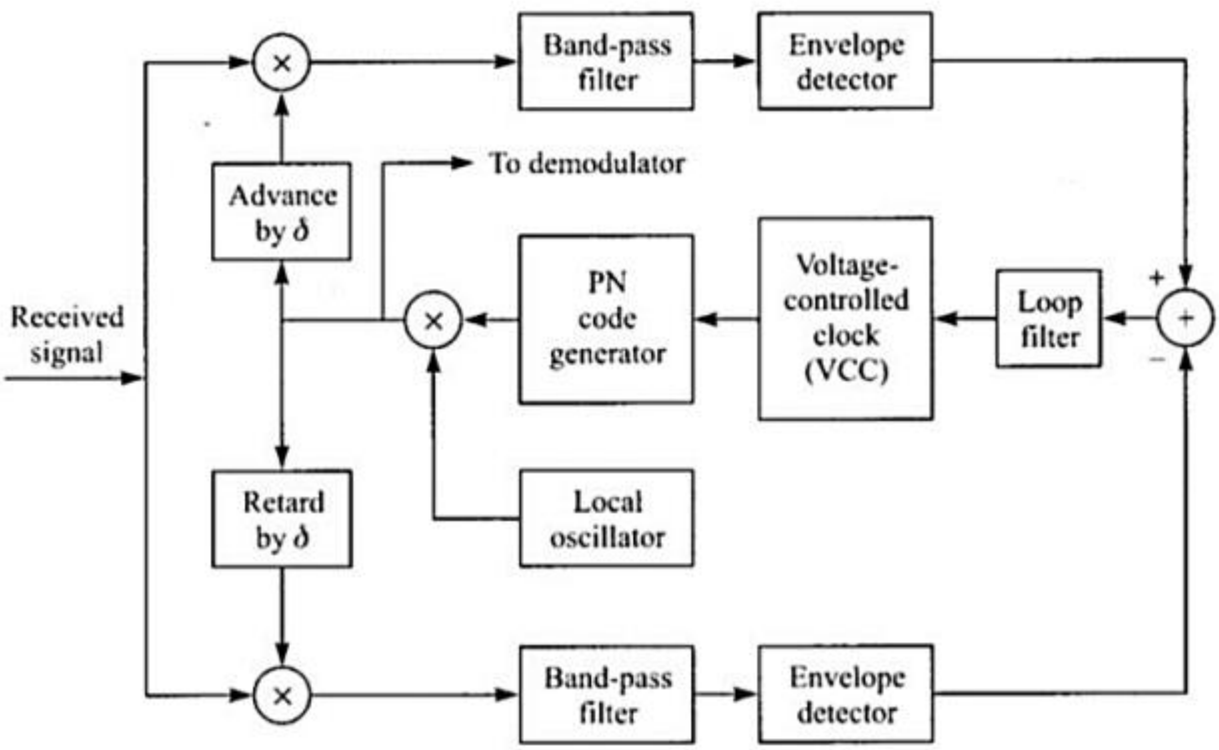


Fig 17 Delay-Locked Loop (DLL) for PN code tracking

The received DS spread spectrum signal is applied simultaneously to two multipliers. One of the multipliers is fed with PN code delayed by δ, a fraction of the chip interval. The other multiplier is fed with the same PN code advanced by δ. The output from each multiplier is fed to a BPF centred on f0. The output of each BPF is envelope detected and then subtracted. This difference signal is applied to the loop filter that drives the voltage controlled oscillator.

The VCO serves as the clock for the PN Code generator. If the synchronization is not exact, the filtered output from one correlator will exceed the other. Hence the VCO will be appropriately advanced or delayed. At the equilibrium point, the two filtered correlator outputs will be equally displaced from the peak value. Then the PN code generator output will be exactly synchronized to the received signal that is fed to the demodulator.

B) FH Spread spectrum system:

A typical tracking technique for FH spread spectrum signals is illustrated in Figure below. Although initial acquisition has been achieved, there is a small timing error between the received signal and the receiver clock. The BPF is tuned to a single intermediate frequency and its bandwidth is of the order of $1/T_c$, where Tc is the chip interval. Its output is envelope detected and then multiplied by the clock signal to produce a three-level signal. This drives the loop filter.

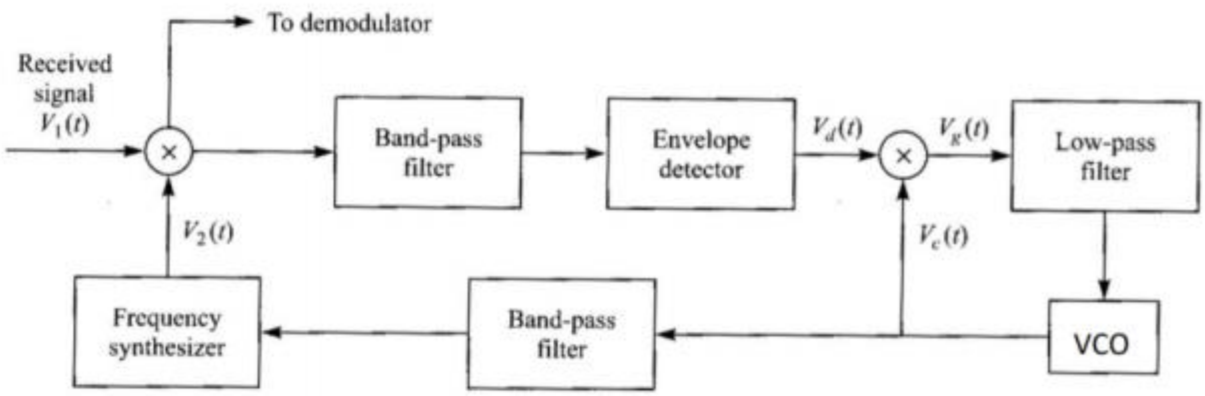


Fig 18 Tracking loop for FH signals

Suppose that the chip transitions from the locally generated sinusoidal waveform do not occur at the same time as the transitions in the incoming signal. Then the output of the loop filter will be either positive or negative, depending on whether the VCO is lagging or advanced relative to the timing of the input signal. This error signal from the loop filter will provide the control signal for adjusting the VCO timing signal so as to drive the frequency synthesizer output to proper synchronism with the received signal.

Short Answer Type Questions

Q1) What is spread spectrum communication?

A1) A system is defined to be a spread spectrum communication system if it fulfills the following requirements.

1. The signal occupies a bandwidth much in excess of the minimum bandwidth necessary to send the information.

2. Spreading is accomplished by means of a spreading signal, often called a code signal, which is independent of the data.

3. At the receiver, despreading (recovering the original data) is done by the correlation of the received spread signal with a synchronized replica of the spreading signal used to spread the information.

Q2) How a pseudo noise (PN) sequence can be generated?

A2) The class of sequences used in spread spectrum communications is usually periodic in that a sequence of 1's and 0's repeats itself exactly with a known period. The maximum length sequence, a type of cyclic code represents a commonly used periodic PN sequence. The maximum length sequences or PN sequences can be generated easily using shift register circuits with feedback from one or more stages. The length of the PN sequence is $N = 2^m-1$, where m is the number of shift register stages.

Q3) Define Direct Sequence Spread Spectrum (DS-SS) system. What are the advantages and disadvantages of Direct Sequence Spread Spectrum (DS-SS) system?

A3) In the Direct sequence spread spectrum (DS-SS) systems, the use of a PN sequence to modulate a phase shift keyed signal achieves instantaneous spreading of the transmission bandwidth.

Advantages

1. This system combats the intentional interference (jamming) most effectively.

2. It has a very high degree of discrimination against the multipath signals. Therefore, the interference caused by the multipath reception is minimized successfully.

3. The performance of DS-SS system in the presence of noise is superior to other systems.

Disadvantages

1. The PN code generator output must have a high rate. The length of such a sequence needs to be long enough to make the sequence truly random.

2. With the serial search system, the acquisition time is too large. This makes the DS-SS system be slow.

Q4) List the commercial applications of spread spectrum techniques?

A4) Spread spectrum signals are used for

1) Combating or suppressing the detrimental effects of interference due to jamming (Intentional interference). It can be used in military applications also.

2) Accommodating multiple users to transmit messages simultaneously over the same channel bandwidth. This type of digital communication in which each user (transmitter-receiver pair) has a distinct PN code for transmitting over a common channel bandwidth is called as Code Division Multiple Access (CDMA) or Spread Spectrum Multiple Access (SSMA). This technique is popularly used in digital cellular communications.

3) Reducing the unintentional interference arising from other users of the channel.

4) Suppressing self-interference due to multipath propagation.

5) Hiding a signal by transmitting it at low power and, thus, making it difficult for an unintended listener to detect in the presence of background noise. It is also called a Low Probability of Intercept (LPI) signal. 6) Achieving message privacy in the presence of other listeners.

7) Obtaining accurate range (time delay) and range rate (velocity) measurements in radar and navigation.

Q5) Define synchronization. State and define the synchronization steps?

A5) The process in which the locally generated carrier at the receiver must be in frequency and phase synchronism with the carrier at the transmitter is called synchronization. In spread spectrum communication systems, there should be perfect alignment between the transmitted and received PN codes, for satisfactory operation.

The process of synchronizing the locally generated spreading signal with the received spread spectrum signal is usually done in two steps. They are

1) Acquisition: The first step called acquisition consists in bringing the two spreading signals into coarse alignment with one another.

2) Tracking: Once the received spread spectrum signal has been acquired, the second step, called tracking, takes over for fine alignment.

Case Study

Decision-Directed Channel Estimation for Multi-User OFDM

In a multi-user OFDM scenario the signal received by each antenna is constituted by the superposition of the signal contributions associated with the different users or transmit antennas. Note that in terms of the multiple-input multiple-output (MIMO) structure of the channel the multi-user single transmit antenna scenario is equivalent, for example, to a single-user space-time coded (STC) scenario using multiple transmit antennas. For the latter a Least-Squares (LS) error channel estimator was proposed by Li et al, which aims at recovering the different transmit antennas' channel transfer functions on the basis of the output signal of a specific reception antenna element and by also capitalising on the remodulated received symbols associated with the different users. The performance of this estimator was found to be limited in terms of the mean-square estimation error in scenarios, where the product of the number of transmit antennas and the number of CIR taps to be estimated per transmit antenna approaches the total number of subcarriers hosted by an OFDM symbol. As a design alternative, in a DDCE was proposed by Jeon et al. For a space-time coded OFDM scenario of two transmit antennas and two receive antennas.

Specifically, the channel transfer function1 associated with each transmit-receive antenna pair was estimated on the basis of the output signal of the specific receive antenna upon subtracting the interfering signal contributions associated with the remaining transmit antennas. These interference contributions were estimated by capitalising on the knowledge of the channel transfer functions of all interfering transmit antennas predicted during the $(n - 1)$-th OFDM symbol period for the n-th OFDM symbol, also invoking the corresponding remodulated symbols associated with the n-th OFDM symbol. To elaborate further, the difference between the subtraction-based channel transfer function estimator and the LS estimator proposed by Li et al. In is that in the former the channel transfer functions predicted during the previous, i.e., the $(n - 1)$-th OFDM symbol period for the current, i.e., the n-th OFDM symbol are employed for both symbol detection as well as for obtaining an updated channel estimate for employment during the $(n + 1)$-th OFDM symbol period. In the approach advocated in the subtraction of the different transmit antennas' interfering signals is performed in the frequency domain.

By contrast, in a similar technique was proposed by Li with the aim of simplifying the DDCE approach of, which operates in the time domain. A prerequisite for the operation of this parallel interference cancellation (PIC)-assisted DDCE is the availability of a reliable estimate of the various channel transfer functions for the current OFDM symbol, which are employed in the cancellation process in order to obtain updated channel transfer function estimates for the demodulation of the next OFDM symbol. In order to compensate for the channel's variation as a function of the OFDM symbol index, linear prediction techniques can be employed, as it was also proposed for example. However, due to the estimator's recursive structure, determining the optimum predictor coefficients is not as straightforward as for the transversal FIR filter-assisted predictor.

References:

1. Bernard Sklar, Prabitra Kumar Ray, "Digital Communications Fundamentals and Applications", Pearson Education, 2nd Edition

2. Wayne Tomasi, "Electronic Communications System", Pearson Education, 5th Edition

3. A.B Carlson, P B Crully, J C Rutledge, "Communication Systems", Tata McGraw Hill Publication, 5th Edition

4. Simon Haykin, "Communication Systems", John Wiley & Sons, 4th Edition

5. Simon Haykin, "Digital Communication Systems", John Wiley & Sons, 4th Edition.

DC

Unit - 5

Information Theoretic Approach to Communication System

5.1 Introduction to information theory

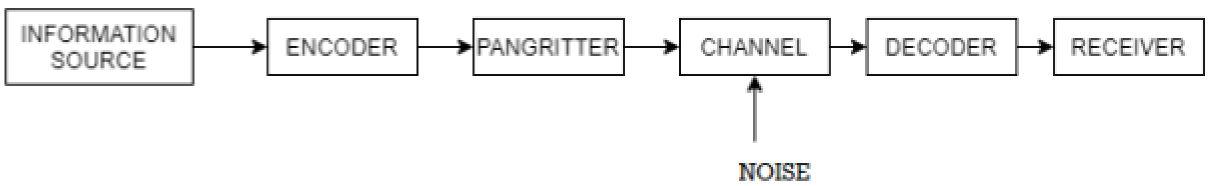
The meaning of the word "information" in Information Theory is the message or intelligence. The message may be an electrical message such as voltage, current or power or speech message or picture message such as facsimile or television or music message.

A source which produces these messages is called **Information source.**

Basics of Information System

Information system can be defined as the message that is generated from the information source and transmitted towards the receiver through transmission medium.

The block diagram of an information system can be drawn as shown in the figure.



Information sources can be classified into

- Analog Information Source
- Digital Information Source

Analog Information source such as microphone actuated by speech or a TV camera scanning a scene, emit one or more continuous amplitude electrical signals with respect to time.

Discrete Information source such as teletype or numerical output of computer consists of a sequence of discrete symbols or letters.

An analog information source can be transformed into discrete information source through the process of sampling and quantizing.

Discrete Information source are characterized by

- Source Alphabet
- Symbol rate
- Source alphabet probabilities

- Probabilistic dependence of symbol in a sequence.

In the block diagram of the Information system as shown in the figure, let us assume that the information source is a discrete source emitting discrete message symbols S1, S2………………. Sq with probabilities of occurrence given by P1, P2…………. Pq respectively. The sum of all these probabilities must be equal to 1.

Therefore P1+ P2 + ………………. + Pq = 1---------------------(1)

Or $\sum_{i=1}^{q} Pi = 1$--------------------(2)

Symbols from the source alphabet S = {s1, s2……………….sn} occurring at the rate of "rs" symbols/sec.

The source encoder converts the symbol sequence into a binary sequence of 0's and 1s by assigning code-words to the symbols in the input sequence.

Transmitter

The transmitter couples the input message signal to the channel. Signal processing operations are performed by the transmitter include amplification, filtering and modulation.

Channel

A communication channel provides the electrical connection between the source and the destination.The cahnnel may be a pair of wires or a telephone cable or a free space over the information bearing symbol is radiated.

When the binary symbols are transmitted over the channel the effect of noise is to convert some of the 0's into 1's and some of the 1's into 0's. The signals are then said to be corrupted by noise.

Decoder and Receiver

The source decoder converts binary output of the channel decoder into a symbol sequence. Therefore the function of the decoder is to convert the corrupted sequence into a symbol sequence and the function of the receiver is to identify the symbol sequence and match it with the correct sequence.

5.2 Entropy and its properties

Let us consider a zero memory space producing independent sequence of symbols with source alphabet

S = {S1,S2,S3………………Sq} with probabilities P = {P1,P2,P3……..Pq} respectively.

Let us consider a long independent sequence of length L symbols . This long sequence then contains

P1 L number of messages of type S1

P2L number of messages of type S2

……

……

PqL number of messages of type Sq.

We know that the self information of S1 = log 1/P1 bits.

Therefore

P1L number of messages of type S1 contain P1L log 1/P1 L bits of information.

P2L number of messages of type S2 contain P2L log 1/P2L bits of information.

Similarly PqL number of messages of type Sq contain PqL log 1/PqL bits of information.

The total self-information content of all these message symbols is given by

I total = P1L log 1/P1 + P2L log 1/P2 +……….+ PqL log 1/Pq bits.

I total = L $\sum_{i=1}^{q} Pi$ log 1/Pi

Average self-information = I total /L

= $\sum_{i=1}^{q} Pi$ log 1/Pi bits/msg symbol.

Average self-information is called **ENTROPY** of the source S denoted by H(s).

H(s)= $\sum_{i=1}^{q} Pi$ log 1/Pi bits/message symbol.

Thus H(s) represents the average uncertainity or the average amount of surprise of the source.

Example 1:

Let us consider a binary source with source alphabet S={S1,S2} with probabilities

P= {1/256, 255/256}

Entorpy H(s) = $\sum_{i=1}^{q} Pi$ log 1/Pi

= 1/256 log 256 + 255/256 log 256/255

H(s) = 0.037 bits / msg symbol.

Example 2:

Let S = {S3,S4} with P' = {7/16,9/16}

Then Entropy H(s') = 7/16 log 16/7 + 9/16 log 16/9

H(s) = 0.989 bits/msg symbol.

Example 3:

Let S" = {S5,S6} with P" = {1/2,1/2}

Then Entropy H(S") = ½ $\log_2 2$ + ½ $\log_2 2$

H(S") = 1 bits/msg symbol.

In this case the uncertainity is maximum for a binary source and becomes impossible to guess which symbol is transmitted.

Zero memory space:

It represents a model of discrete information source emitting a sequence of symbols from S= {S1,S2………………….. Sq} successive symbols are selected according to some fixed probability law and are statistically independent of one another. This means that there is no connection between any two symbols and that the source has no memory.Such type of source are memoryless or zero memory sources.

Information rate

Let us suppose that the symbols are emitted by the source at a fixed time rate "rs" symbols/sec. The average source information rate "Rs" in bits/sec is defined as the product of the average information content per symbol and the message symbol rate "rs"

Rs = rs H(s) bits/sec

Consider a source S={S1,S2,S3} with P={1/2,1/4,1/4}

Find

a) Self information

b) Entropy of source S

a) Self information of S1 = $I_1 = \log_{2 1/P1}$ = $\text{Log}_2 2 = 1$ bit

Self information of S2 = $I_2 = \log_2 1/P_2 = \text{Log}_2 4 = 2$ bits

Self information of S3 = $I_3 = \log_2 1/P_3 = \text{Log}_2 4 = 2$ bits

b) Average Information content or Entropy =

H(s) = $\sum_{i=1}^{3} Pi$ Ii

= $I_1 P_1 + I_2 P_2 + I_3 P_3$

= ½ (1) + ¼ (2) + ¼ (2)

= 1.5 bits/msg symbol.

Q. The collector volatge of a certain circuit lie between -5V and -12V.The voltage can take only these values -5,-6,-7,-9,-11,-12 volts with the respective probabilities 1/6,1/3,1/12,1/12,1/6,1/6. This voltage is recorded with a pen recorder. Determine the average self information associated with the record in terms of bits/level.

Given Q = 6 levels. P= {1/6,1/3,1/12,1/12,1/6,1/6}

Average self-information H(s) = $\sum_{i=1}^{6} Pi$ log $_2$ 1/Pi

= 1/6 log $_2 6$ + 1/3 Log $_2 3$ + 1/12 log $_2$ 12 + 1/12 log $_2$ 12 + 1/6 Log $_2 6$ + 1/6 Log $_2 6$

H(s) = 2.418 bits/level

Q. A binary source is emitting an independent sequence of 0s and 1s with probabilities p and 1-p respectively. Plot entopy of source versus p.

The entropy of the binary source is given by

H(s) = $\sum_{i=1}^{2} Pi$ log 1/Pi

= P1 log 1/P1 + P2 log 1/P2

= P log (1/P) + (1-P) log 1/(1- P)

P=0.1 = 0.1 log(1/0.1) + 0.9 log(1/0.9)

= 0.469 bits/symbol.

At p=0.2

0.2 log(1/0.2) + 0.8 log(1/0.8)

= 0.722 bits/symbol.

At p=0.3

0.3 log(0.3) + 0.7 log(1/0.7)

= 0.881 bits/symbol.

At p=0.4

0.4 log(0.4) + 0.6 log(1/0.6)

= 0.971 bits/symbol.

At p=0.5

0.5 log(0.5) + 0.5 log(1/0.5)

= 1 bits/symbol.

At p=0.6

H(s) = 0.6 log (1/0.6) + 0.4 log(1/0.4)

H(s) = 0.971 biys/symbol

At p=0.7

0.7 log(1/0.7) + 0.3 log(1/0.3)

= 0.881 bits/symbol.

At p=0.8

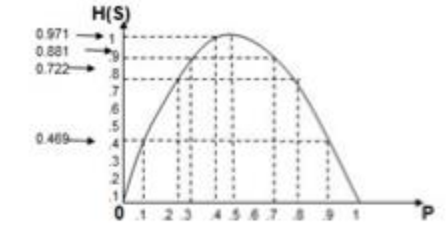
0.8 log (1/0.8) +0.2 log(1/0.2)

0.722 bits/symbol

At p=0.9

0.9 log(1/0.9) + 0.1 log(1/0.1)

=0.469 bits /symbol.



Q. Let X represent the outcome of a single roll of a fair die. What is the entropy of X?

Since a die has six faces , Probability of getting any number is 1/6.

Px = 1/6.

So entropy of X = H(X) = Px log 1/Px = 1/6 log 6 = 0.431 bits / msg symbol.

Q. Find the entopy of the source in bits/symbol of the source that emits one out of four symbols A,B,C and D in a statistically independent sequences with probabilities ½,1/4,1/8 and 1/8 .

The entropy of the source is given by

H(s) = $\sum_{i=A}^{D} Pi$ log 1/Pi

= Pa log 1/Pa + Pb log 1/Pb + Pc log 1/Pc + Pd log 1/Pd

= ½ $\log_2 2$ + ¼ $\log_2 4$ + 1/8 $\log_2 8$ + 1/8 $\log_2 8$

H(s) = 1.75 bits/msg symbol.

1 nat = 1.443 bits

1bit = 1/1.443 = 0.693 nats.

Entropy of the souce H(s) = 1.75 X 0.693 =

H(s) = 1.213 nats/symbol.

Properties

- The entropy function for a source alphabet S = $\{S_1,S_2,S_3 \ldots\ldots\ldots\ldots\ldots\ldots\ldots Sq\}$ with probability

P = { P1,P2,P3…………………Pq} where q = no of source symbols as

H(s) = $\sum_{i=1}^{q} Pi$ log (1/Pi) = $\sum_{i=1}^{q} Pi\, Ii$ bits/symbol.

The properties can be observed as :

The entropy function is continuous for every independent variable Pk in the interval (0,1) that is if Pk varies continuously from 0 to 1 , so does the entropy function.

The entropy function vanishes at both Pk = 0 and pk =1 that is H(s) = 0.

- The entropy function is symmetrical function of its arguments

H[Pk,(1-Pk)]= H[(1-Pk),Pk]

The value of H(s) remains the same irrespective of the location of the probabilities.

If $P_A = \{ P_1,P_2,P_3\}$

$P_B = \{ P_2, P_3,P_1\}$

$P_C = \{ P_3,P_2,P_1\}$

Then $H(S_A) = H(S_B) = H(S_C)$

- **Extremal Property**

Let us consider the source with "q " symbols S={S1,S2……… Sq} with probabilities P = {P1,P2,………Pq}.Then the entropy is given by

H(s) = $\sum_{i=1}^{q} Pi$ log 1/Pi -------------------------(1)

And we know that

P1 +P2 + P3 ……………. Pq = $\sum_{i=1}^{q} Pi$ = 1

Let us now prove that

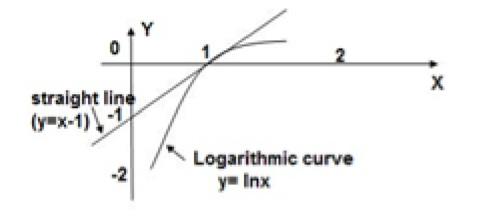
Log q – H(s) = [$\sum_{i=1}^{q} Pi$] log q - $\sum_{i=1}^{q} Pi$ log (1/Pi)]-------------------------(2)

=$\sum_{i=1}^{q} Pi$ [log q - log (1/Pi)]---------------------------------(3)

=$\sum_{i=1}^{q} Pi$ [log qPi]----------------------------------(4)

= $\sum_{i=1}^{q} Pi$ [Log_e q Pi] / [Log_e 2]-------------------------(5)

Log q – H(s) = $\log_2 e$ $\sum_{i=1}^{q} Pi$ ln qPi ------------------------------(6)



From the graph it is evident that the straight line y=x-1 always lies above the logarithmic curve y = ln x except at x=1. Thus the straight line forms the tangent to the curve at x=1.

Therefor ln x ≤ x-1

Multipling (-1) on both sides we get

-ln x ≥ 1-x or ln 1/x ≥ 1-x ------------------(7)

If X = 1/qp then ln qp ≥ 1 – 1/qp----------------------(8)

Multipling equation 2 both sides by Pi and taking summation we get for all i=1 to q we get

$\sum_{i=1}^{q} Pi$ ln q Pi ≥ $\sum_{i=1}^{q} Pi$ [1 – 1/qPi] --------------------------(9)

Multiplying eq(3) by log 2 e we get

$\text{Log}_2\, e \sum_{i=1}^{q} Pi \;\; \ln q\, Pi \geq \text{Log}_2\, e\, [\sum_{i=1}^{q} Pi - \sum_{i=1}^{q} 1/q$ ------------------------(10)

From equation (6)

LHS of log q – H(s) and RHS of eq(6) is zero [Since [$\sum_{i=1}^{q} Pi - \sum_{i=1}^{q} 1/q$ =0]

Therefore log q – H(s) = 0

Or H(s) ≤ $\log_2 q$; Suppose Pi -1/q =0 ; Pi = 1/q for all i=1,2,3………… q which satisfies then

H(s) max = $\log_2 q$ bits/msg symbol.

Property of Additivity

Suppose that we split the symbol sq into n sub symbols such that sq = sq1,sq2……….sqn with probabilities Pq1,Pq2,Pq3…………………………. Pqn such that

Pqj = Pq1 + Pq2 + Pq3 +………………….. +Pqn = $\sum_{j=1}^{n} Pqj$

Then the splitted symbol entropy is

$H(s') = H(P_1,P_2,P_3\ldots\ldots\ldots Pq\text{-}1, Pq1, Pq2\ldots\ldots\ldots Pqn)$

$= \sum_{i=1}^{n} Pi$ log (1/Pi) + $\sum_{j=1}^{n} Pqj$ log 1/Pqj

H'(s) = $\sum_{i=1}^{n} Pi$ log (1/Pi) – $\sum_{j=1}^{n} Pq$ log 1/Pq + $\sum_{j=1}^{n} Pqj$ log 1/Pqj

H'(s) = $\sum_{i=1}^{n} Pi$ log (1/Pi) + $\sum_{j=1}^{n} Pqj$ [log 1/Pqj- [log 1/Pq]

$= \sum_{i=1}^{n} Pi$ log (1/Pi) + Pq $\frac{\sum_{j=1}^{n} Pqj}{Pq}$ [log Pq/Pqj]

H(s) = H(s)

Therefore H'(s) ≥ H(s)

That is partitioning of symbols into sub symbols cannot decrease entropy

The source effeciency

$$\eta_s = \frac{H(s)}{H(s)_{max}}$$

Source Redunancy

$$R_{\eta s} = 1 - \eta_s$$

Verify that the rule of additivity for the following source S= {S1,S2,S3,S4} with P={1/2,1/3,1/12,1/12}

The entropy H(s) is given by

H(s) = $\sum_{i=1}^{4} Pi$ log 1/Pi

= ½ $\log_2 2$ + 1/3 $\log_2 3$ + 2 x 1/12 $\log_2 12$

H(s) = 1.6258 bits/msg symbol.

From additive property we have

H'(s) = $P_1 \log (1/P_1) + (1-P_1) \log 1/(1-P_1)$

= (1-P1){ P2/(1-P1) log (1-P1)/P2 + P3/(1-P1) log(1-P1)/P3 + P4/(1-P1) log (1-P1)/P4

= 1/2 log 2 + ½ Log 2 + ½ { (1/3)/(1/2) log(1/2)/(1/2) + (1/12)/1/2 log(1/12)/(1/2) + (1/12)/(1/2) log (1/2) /(1/12)

H'(s) = 1.6258 bits/symbol.

Q. A discreter message source S emits two independent symbols X and Y with probabilities 0.55 and 0.45 respectively. Calculate the effeciency of the source and its redundancy.

Given H(s) = $\sum_{i=x}^{Y} Pi$ log 1/Pi

Px log 1/Px + Py log 1/Py

= 0.55 log 1/0.55 + 0.45 log 1/0.45

= 0.9928 bits/msg symbol.

The maximum entropy is given by

H(s) max = log $_2$ q = log $_2$ 2 = 1 bits/symbol.

The source effeciency is given by

η_s = H(s) max/ H(s) = 0.9928/1 = 0.9928

η_s = 99.28 %

The source redundacy is given by

Rs = 1-η_s = 0.0072

Rns = 0.72%

5.3 Source coding theorem

Source emcoding is a process in which the output of an information source is converted into a r-ary sequence where r is the number of different symbols used in this transformation process . The functional block that performs this task in a communication system is called source encoder.

The input to the encoder is the symbol sequence emitted by the information source and the output of the encoder is r-ary sequence.

If r=2 then it is called the binary sequence

r=3 then it is called ternary sequence

r=4 then it is called quarternary sequence..

Let the source alphabet S consist of "q" number of source symbols given by S = {S1,S2.......Sq} . Let another alphabet X called the code alphabet consists of "r" number of coding symbols given by

$X = \{X_1, X_2 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots X_r\}$

The term coding can now be defined as the transformation of the source symbols into some sequence of symbols from code alphabet X.

Shannon's encoding algorithm

The following steps indicate the Shannon's procedure for generating binary codes.

Step 1:

List the source symbols in the order of non-increasing probabilities

$S = \{S_1, S_2, S_3 \ldots\ldots\ldots\ldots\ldots\ldots\ldots S_q\}$ with $P = \{P_1, P_2, P_3 \ldots\ldots\ldots\ldots\ldots\ldots\ldots P_q\}$

$P_1 \geq P_2 \geq P_3 \geq \ldots\ldots\ldots\ldots\ldots\ldots\ldots P_q$

Step 2

Compute the sequences $\propto 1 = 0$

$\propto 2 = P1 = P1 + \propto 1$

$\propto 3 = P2 + P1 = P2 + \propto 2$

$\propto 4 = P3+P2+P1 = P3 + \propto 3$

….. ……..

$\propto q+1 = Pq + Pq\text{-}1 + \ldots\ldots\ldots + P2 + P1 = Pq + \propto q$

Step 3:

Determine the smallest integer value of "li using the inequality

$2^{li} \geq 1/Pi$ for all i=1,2,……………………q.

Step 4:

Expand the decimal number αi in binary form upto li places neglecting expansion beyond li places.

Step 5:

Remove the binary point to get the desired code.

Example :

Apply Shannon's encoding (binary algorithm) to the following set of messages and obtain code efficiency and redundancy.

m 1	m 2	m 3	m 4	m 5
1/8	1/16	3/16	¼	3/8

Step 1:

The symbols are arranged according to non-incraesing probabilities as below:

m 5	m 4	m 3	m 2	m 1

3/8	¼	3/16	1/16	1/8
P 1	P 2	P 3	P 4	P 5

The following sequences of $\propto$ are computed

$\propto 1 = 0$

$\propto 2 =$ P1 + $\propto 1$ = 3/8 + 0 = 0.375

$\propto 3$ = P2 + $\propto 2$ = ¼ + 0.375 = 0.625

$\propto 4$ = P3 + $\propto 3$ = 3/16 + 0.625 = 0.8125

$\propto 5$ = P4 + $\propto 4$ = 1/16 + 0.8125 = 0.9375

$\propto 6$ = P5 + $\propto 5$ = 1/8 + 0.9375 = 1

Step 3:

The smallest integer value of li is found by using

$2^{li} \geq 1/Pi$

For i=1

$2^{l1} \geq 1/P1$

$2^{l1} \geq 1/3/8 = 8/3 = 2.66$

The samllest value ofl1 which satisfies the above inequality is 2 therefore $l_1 = 2$.

For i=2 $2^{l2} \geq 1/P2$ Therefore $2^{l2} \geq 1/P2$

$2^{l2} \geq 4$ therefore l2 = 2

For i =3 $2^{l3} \geq 1/P3$ Therefore $2^{l3} \geq 1/P3$ $2^{l3} \geq 1/\ 3/16 = 16/3$ l3 =3

For i=4 $2^{l4} \geq 1/P4$ Therefore $2^{l4} \geq 1/P4$

$2^{l4} \geq 8$ Therefore l4 = 3

For i=5

$2^{l5} \geq 1/P5$ Therefore

$2^{l5} \geq 16$ Therefore l5=4

Step 4

The decimal numbers αi are expanded in binary form upto li places as given below.

$\propto 1 = 0$

$\propto 2 =$ 0.375

0.375 x2 = 0.750 with carry 0

0.75 x 2 = 0.50 with carry 1

0.50 x 2 = 0.0 with carry 1

Therefore

$(0.375)_{10} = (0.11)_2$

$\propto 3$ = 0.625

0.625 X 2 = 0.25 with carry 1

0.25 X 2 = 0.5 with carry 0

0.5 x 2 = 0.0 with carry 1

$(0.625)_{10} = (0.101)_2$

$\propto 4$ = 0.8125

0.8125 x 2 = 0.625 with carry 1

0.625 x 2 = 0.25 with carry 1

0.25 x 2 = 0. 5 with carry 0

0.5 x 2 = 0.0 with carry 1

$(0.8125)_{10} = (0.1101)_2$

$\propto 5$ = 0.9375

0.9375 x 2 = 0.875 with carry 1

0.875 x 2 = 0.75 with carry 1

0.75 x 2 = 0.5 with carry 1

0.5 x 2 = 0.0 with carry 1

$(0.9375)_{10} = (0.1111)_2$

$\propto 6$ = 1

Step 5:

$\propto 1 = 0$ and l1 = 2 code for S1 ----→00

$\propto 2$ =$(0.011)_2$ and l2 =2 code for S2 ----→ 01

$\propto 3$ = $(.101)_2$ and l3 =3 code for S3 -------→ 101

$\propto 4$ = $(.1101)_2$ and l4 =3 code for S -----→ 110

$\propto 5$ = $(0.1111)_2$ and l5 = 4 code for s5 ---→ 1111

The average length is computed by using

L = $\sum_{i=1}^{5} Pi\ li$

= 3/8 x 2 + ¼ x 2 + 3/16 x 3 + 1/8 x 3 + 1/16 x 4

L= 2.4375 bits/msg symbol

The entropy is given by

$H(s) = 3/8 \log (8/3) + ¼ \log 4 + 3/16 \log 16/3 + 1/8 \log 8 + 1/16 \log 16$

H(s) =2.1085 bits/msg symbol.

Code efficiency is given by

$\eta_c = H(s)/L = 2.1085/2.4375 \quad = 0.865$

Percentage code effeciency = 86.5%

Code redundancy

$R\eta_c = 1 - \eta_c = 0.135$ or 13.5%.

5.4 Huffman coding

First the source symbols are listed in the non-increasing order of probabilities

Consider the equation

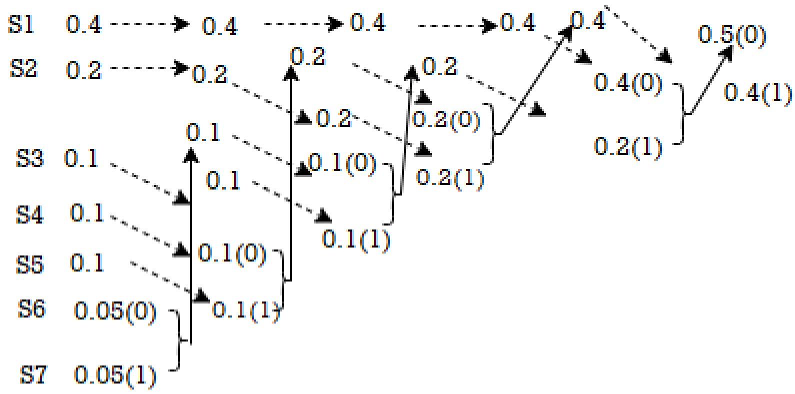
$q = r + (r-1)\ \alpha$

Where q = number of source symbols

r = no of different symbols used in the code alphabet

The quantity α is calculated and it should be an integer . If it is not then dummy symbols with zero probabilities are added to q to make the quantity) α an integer. To explain this procedure consider some examples.

Given the messages X1,X2,X3,X4,X5 and X6 with respective probabilities of 0.4,0.2,0.2,0.1,0.07 and 0.03. Construct a binary code by applying Huffman encoding procedure . Determine the efficiency and redundancy of the code formed.



Now Huffmann code is listed below:

Source Symbols	Pi	Code	Length (li)
X1	0.4	1	1
X2	0.2	01	2
X3	0.2	000	3
X4	0.1	0010	4
X5	0.07	00110	5
X6	0.03	00111	5

Now Average length (L) = $\sum_{i=1}^{6} Pi$ li

L = 0.4 x 1 + 0.2 x 2 + 0.2 x 3 + 0.1 x 4 + 0.07x 5 + 0.03 x 5

L = 2.3 bits/msg symbol.

Entropy H(s) = = $\sum_{i=1}^{6} Pi$ log 1/Pi

0.4 log 1/0.4 + 0.2 log 1/0.2 + 0.1 log 1/0.1 + 0.07 log 1/0.07 + 0.03 log 1/0.03

H(s) = 2.209 bits /msg symbol.

Code effeciency = nc = H(s) /L = 2.209/2.3 = 96.04%.

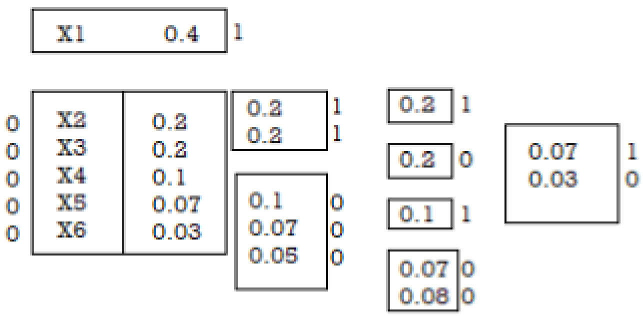
5.5 Shannon-Fano coding

Shanon-Fano Encoding procedure for getting a compact code with minimum redundancy is given below:

- The symbols are arranged according to non-increasing probabilities
- The symbols are divided into two groups so that the su of probabilities in each group is approximately equal.
- All the symbols in the first group are designated by "1" and the second group by "0.
- The first group is again subduivided into two subgrous such that rach subgroup probabilities are approximately same.
- All the symbols of the first subgroup are designated by 1 and second subgroup by 0
- The second subgroup is subdivided into two or more subgropus and step 5 is repeated.
- This process is continued till further sub-division is impossible.

Given are the messages X1,X2,X3,X4,X5,X6 with respective probabilities 0.4,0.2,0.2,0.1,0.07 and 0.03. Construct a binary code by applying Shannon-fano encoding procedure . Determine code effeciency and redundancy of the code.

Applying Shannon-fano encoding procedure



CODE	LN IN BURST
1	1
011	2
010	2
001	3
0001	4
0000	4

Average length L is given by

$$L = \sum_{i=1}^{6} Pi\ li$$

= 0.4 x 1 + 0.2 x 3 + 0.2 x 3 + 0.1 x 3 + 0.07 x4 + 0.03 X 4

L = 2.3 bits/msg symbol.

Entropy

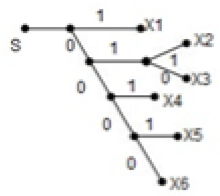
H(s) = $\sum_{i=1}^{6} Pi$ log 1/Pi

0.4 log 1/0.4 + 2x 0.2 log 1/0.2 + 0.1 log 1/0.1 + 0.07 log 1/0.07 + 0.03 log 1/0.03

H(s) = 2.209 bits/msg symbol.

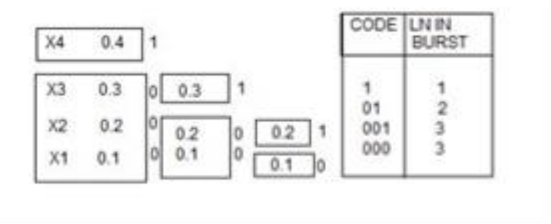
Code Efficiency = ηc = H(s) / L = 2.209/2.03 = 96.04%

Code Redundancy = $R_{\eta c}$ = 1- ηc = 3.96%



Q. You are given four messages X1 , X2 , X3 and X4 with respective probabilities 0.1,0.2,0.3,0.4

- Device a code with pre-fix property (Shannon fano code) for these messages and draw the code.
- Calculate the efficiency and redundancy of the code.
- Calculate the probabilities of 0's and 1's in the code.



Average length L = $\sum_{i=1}^{4} Pi$ li

= 0.4 x1 + 0.3 x 2 + 0.2 x 3 + 0.1 x3

L = 1.9 bits/symbol.

Entropy H(s) = $\sum_{i=1}^{4} Pi$ log 1/Pi

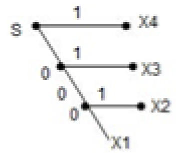
= 0.4 log 1/0.4 + 0.3 log 1/0.3 + 0.2 log 1/0.2 + 0.1 log 1/0.1

H(s) = 1.846 bits/msg symbol.

Code efficiency ηc = H(s) / L = 1.846/1.9 = 97.15%

Code Redundancy = $R_{\eta c}$ = 1- ηc = 2.85%

The code tree can be drawn



The probabilities of 0's and 1's in the code are found by using the

P(0)= 1/L $\sum_{i=1}^{4}$(no of 0's in code xi) (Pi)

= 1/1.9 [3 x 0.1 + 2x 0.2 + 1x 0.3 + 0(0.4)]

P(0) = 0.5623

P(1) = 1/L $\sum_{i=1}^{4}$(no of 1's in code xi) (Pi)

= 1/1.9 [0x0.1+1x0.2+1x0.+1x0.4]

P(1) = 0.4737

Shanon –Fano Ternary Code:

The seven steps observed for getting a binary code are slightly modified for ternary code as given below:

- The symbols are arranged according to non-increasing probabilities.
- The symbols are divided into three groups so that the sum of probbilities in each group is approximately equal.
- All the symbols in the first group designated by a "2" , second group by "1" and third group by "0".
- The first group is again subdivdided into three more sub groups such that each subgroup probabilities are approximately same.
- All the symbols of first subgroup are designated by "2" , the second subgroup by 1 and third subgroup by 0.
- The 2^{nd} and 3^{rd} group are subdivided into three more subgroups each and step 5 is repeated.
- This process is continued till further sub division is impossible.

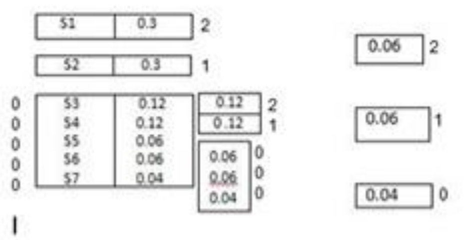
Q. Construct a Shannon-fano ternary code for the following ensenble and find the code efficiency and redundancy.Also draw the corresponding code tree.

S = { S1,S2,S3,S4,S5,S6,S7}

P={ 0.3,0.3,0.12,0.12,0.06,0.06,0.04} with

X= {0,1,2}

Solution



CODE	LI IN BURST
2	1
1	1
02	2
01	2
002	3
001	3
000	3

The average length of the ternary tree is given by

$L = \sum_{i=1}^{7} Pi$ li -----------------(1)

= 0.3 x 1 + 0.3 x 1 + 0.12 x 2 +0.12 x 2 + 0.06 x3 +0.06x 3 +0.04 x 3

L = 1.56 trinits/msg symbol.

Entropy in bits/msg symbolis given by

$H(s) = \sum_{i=1}^{7} Pi$ log 1/Pi

= 0.3 log 1/0.3 x 2 + 0.12 log1/0.12 x2 + 0.06 log 1/0.06 x 2 +0.04 log 1/0.04

= 2.4491 bits/msg symbol.

Now entropy in r-ary units per message symbol is given by

$H_r(s) = H(s) / \log_2 r$

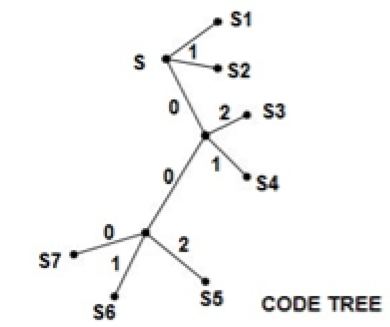
r=3

H3 (s) = H(s) / $\log_2$ 3 = 2.4419/ $\log_2$ 3 = 1.5452.

The terenary coding effeciency is given by

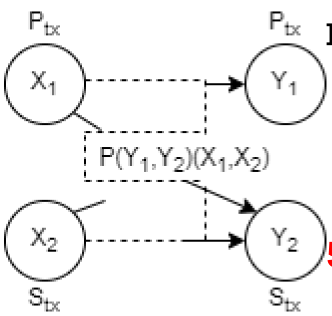
ηc = H3(s) / L = 1.5452/ 1.56 = 0.99 bits/msg symbol.

ηc = 99.05%



5.6 Discrete memory less channel

- Communication over a discrete memory less channel takes place in a discrete number of "channel uses," indexed by the natural number $n \in \mathbb{N}$.
- The concepts use the simple two-transmitter, two-receiver channel shown in Figure.
- The primary (secondary) transmitter P_{tx} (S_{tx}) wishes to communicate a message to a single primary (secondary) receiver P_{rx} (S_{rx}).
- The transmitters communicate their messages by transmitting codewords, which span n channel uses (one input symbol per channel use). The receivers independently decode the received signals, often corrupted by noise according to the statistical channel model, to obtain the desired message.
- One quantity of fundamental interest in such communication is the maximal rate, typically cited in bits/channel use at which communication can take place.
- Most information theoretic results of interest are asymptotic in the number of channel uses; that is, they hold in the limit as $n \to \infty$.



Discrete memory less channel

5.7 Mutual information

Mutual information is a quantity that measures a relationship between two random variables that are sampled simultaneously.

It measures how much information is communicated, on average, in one random variable about another.

For example, suppose X represents the roll of a fair 6-sided die, and Y represents whether the roll is even (0 if even, 1 if odd). Clearly, the value of Y tells us something about the value of X and vice versa. That is, these variables share mutual information.

On the other hand, if X represents the roll of one fair die, and Z represents the roll of another fair die, then X and Z share no mutual information. The roll of one die does not contain any information about the outcome of the other die.

An important theorem from information theory says that the mutual information between two variables is 0 if and only if the two variables are statistically independent. The formal definition of the mutual information of two random variables X and Y, whose joint distribution is defined by

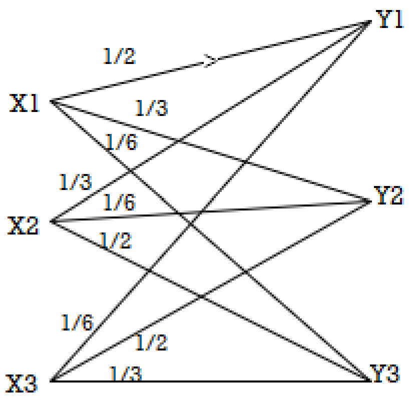
$$I(X;Y) = \sum_{x \in X}\sum_{y \in Y} P(x,y) \log P(x.y)/ P(x)\, P(y)$$

5.8 Channel capacity

A channel is said to be symmetric or uniform channel if the second and subsequent rows of the channel matrix contain the same elements as that of the first row but in different order.

For example

$$P(Y/X) = \begin{matrix} 1/2 & 1/3 & 1/6 \\ 1/3 & 1/6 & 1/2 \\ 1/6 & 1/2 & 1/3 \end{matrix}$$



Channel matrix and diagram of symmetric/uniform channel

The equivocation H(B/A) is given by

$$H(B/A) = \sum_{i=1}^{r}\sum_{j=1}^{s} P(ai,bj) \quad \log 1/P(bi/ai)$$

But p (ai, bj) = P(ai) P(bj/ai)

$$H(B/A) = \sum_{i=1}^{r} P(ai) \sum_{j=1}^{s} P(bj/ai) \quad \log 1/P(bj/ai)$$

Since $\sum_{j=1}^{s} P(bj/ai) \quad \log 1/P(bj/ai) = h$

H(B/A) = 1.h

H= $\sum_{j=1}^{s} P(j)$ log 1/P(j)

Therefore, the mutual information I (A, B) is given by

I (A, B) is given by H(B) –H(B/A)

I (A, B) = H(B) –h

Now the channel capacity is defined as

C = Max [I (A, B)]

= Max[H(B) – h]

C = Max[H(b) –h] -----------------------(X)

But H(B) the entropy of the output symbol becomes maximum if and only if when all the received symbols become equiprobable and since there are 'S' number of output symbols we have H(B) max = max[H(B) = log s---------------------(Y)

Substituting (Y) in (X) we get

C = $\log_2$ s – h

For the channel matrix shown find the channel capacity

P(bj/ai) = $\begin{matrix} 1/2 & 1/3 & 1/6 \\ 1/3 & 1/6 & 1/2 \\ 1/6 & 1/2 & 1/3 \end{matrix}$

The given matrix belongs to a symmetric or uniform channel

H(B/A) = h = $\sum_{i=1}^{3} Pj$ log 1/Pj

= ½ log 2 + 1/3 Log 3 + 1/6 log 6

H(B/A) = 1.4591 bits/msg symbol.

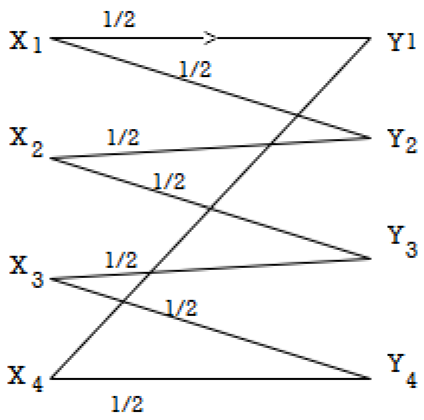
The channel capacity is given by

C = (log S – h) rs rs = 1 msgsymbol/sec

C = log 3 – 1.4591

C = 0.1258 bits/sec

Q. Determine the capacity of the channel as shown in the figure.



The channelmatrix can be written as

P(Y/X) = ½ ½ 0 0

0 ½ ½ 0

0 0 ½ ½

½ 0 0 ½

Now h = H(Y/X) = $\sum_{j=1}^{4} P_j$ log 1/Pj

H = ½ log 2 + ½ log 2 +0+ 0 = 1 bits/Msg symbol.

Channelcapacity

C = log s – h

= log 4 – 1

= 1 bits/msg symbol.

5.9 Channel coding theorem

Theorem 1 :

Consider a discrete memoryless channel capacity C . For all R<C there exists a sequence of block codes (Cn(M,n) $_{n>0}$ of rate Rn together with a decoding algorithm such that

Lim n-> ∞ Rn =R and lim n→ ∞ sup $_{x} \in _{Cn}$ Pe (Cn,x) = 0

Theorem 2:

Consider a discrete memoryless channel capacity C . Any code C of rate R>C satisfies

$1/M \sum_{x\in C} Pe\,(C,x) > K(C,R)$ where $K(C,R) > 0$ depends on the channel and the rate but is independent of the code.

5.10 Differential entropy and mutual Information for continuous ensembles

The differential entropy of a continuous random variable X with pdf f(x) is

$h(x) = -\int f(x)\log f(x)dx = -E[\log(f(x))]$ --------------------(1)

Definition: Consider a pair of continuous random variable (X,Y) distributed according to the joint pdf (x,y) . The joint entropy is given by

$h(X,Y) = -\iint f(x,y)\log f(x,y)dx\,dy$ ------------------------------(2)

While the conditional entropy is

$h(X|Y) = -\iint f(x,y)\log f(x|y)\,dx\,dy$ --(3)

5.11 Information Capacity theorem

Shannon-Hartley law states that the capacity of a band limited Gaussian channel with AWGN is given by

C = B log(1+B/N) bits/sec

Where B ----- channel bandwidth in Hz

S ----------------------- signal power in watts

N ------------------------- noise power in watts.

Proof: Consider the channel capacity ‘C’ given by

$C = [H(Y) – H(N)]\ max$------------------------(1)

But

$H(N)\ max = B \log 2\pi e N$ bits/sec

$H(Y)\ max = B \log 2\pi e\ \sigma \gamma^2$ bits/sec

But $\sigma \gamma^2 = S + N$

$H(Y)\ max = B \log 2\pi e (S+N)$ bits/sec

C = = B log 2π e (S+N) - = B log 2π e N

$C = B \log 2\pi e (S+N)/ 2\pi e N$

$C = B \log (S+N)/N$

$C = B \log (1+S/N)$ bits/sec

1. A voice grade channel of the telephone network has a bandwidth of 3.4KHz

- Calculate the channel capacity of the telephone channel for a signal to noise ratio of 30db
- Calculate the minimum signalto noise ratio required to support information through the telephone channel at the rate of 4800 bits/sec

Given :

Bandwidth = 3.4 KHz

$10 \log_{10} S/N = 30$ db

$S/N = 10^3 = 1000$

Channel capacity

$C = B \log 2\ (1+S/N) = 3400 \log (1 + 1000)$

$C = 33889$ bits/sec

Given $C = 4800$ bits/sec

$4800 = 3400 \log 2(1+S?N)$

$S/N = 2\ 48/34 - 1$

$S/N = 1.66$

$S/N = 10 \log 1.66 = 2.2$ db

$S/N = 2.2$ db

2. An analog signal has a 4 Khz bandwidth . The signal is sampled at 2.5 times the Nyquist rate and each sample quantized into 256 equally likely levels. Assume that the successive samples are statistically independent.

- Find the information rate of the source
- Can the output of the source be transmitted without error over the Gaussian channel of bandwidth 50Hz and S/N ratio of 20db
- If the output of the source is to be transmitted without errors over an analog channel having S/N of 10 db , Compute bandwidth requirement of the channel.

Given $B = 4000$Hz Therefore Nyquist rate $= 2B = 8000$Hz

Sampling rate $rs = 2.5$ Nyquist rate

$= 2.5 \times 8000 = 20{,}000$ samples/sec

Since all the quantization levels are equal likely the maximum information content

$I = \log_e q = \log_2 256 = 8$ bits /sample.

Information rate $Rs = rs\ .I = 20{,}000 \times 8 = 1{,}60{,}000$ bits/sec

From Shannon's Hartley law

$C = B \log (1+ S/N)$

$C = 332.91 \times 10^{3}$ bits/sec

Iii)

$C = B \log (1+S/N) = Rs$

$B = Rs/\log(1+S/N) = 46.25$ KHz

B = 46.25 KHz.

References:

1. Bernard Sklar, Prabitra Kumar Ray, "Digital Communications Fundamentals and Applications", Pearson Education, 2nd Edition

2. Wayne Tomasi, "Electronic Communications System", Pearson Education, 5th Edition

3. A.B Carlson, P B Crully, J C Rutledge, "Communication Systems", Tata McGraw Hill Publication, 5th Edition

4. Simon Haykin, "Communication Systems", John Wiley & Sons, 4th Edition

5. Simon Haykin, "Digital Communication Systems", John Wiley & Sons, 4th Edition.

DC

Unit - 6

Error-Control Coding

6.1 Linear Block Codes: Coding

In channel encoder a block of k-msg bits is encoded into a block of n bits by adding (n-k) number of check bits as shown in figure.

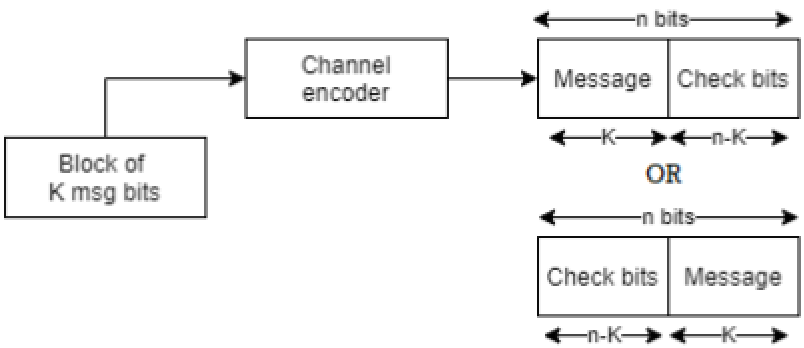


Fig: Linear Block Code

A (n,k) block code is said to be a (n,k) linear block code if it satisfies the condition

Let C1 and C2 be any two code words belonging to a set of (n,k) block code

Then C1⊕C2 is also an n-bit code word belonging to the same set of (n,k) block code.

A (n,k) linear block code is said to be systematic if k-msg bits appear either at the beginning or end of the code word.

Matrix Description of linear block codes:

Let the message block of k-bits be represented as row vector called as message vector given by

[D] = { d1,d2………………dk}

Where d1,d2………………dk are either 0's 0r 1's

The channel encoder systematically adds(n-k) number of check bits toform a (n,k) linear block code. Then the 2k code vector can be represented by

C = {c1,c2……………cn) ---------------------------(2)

In a systematic linear block code the msg bits appear at the beginning of the code vector

Therefore ci=di for all I = 1 to k.--------------------------------(3)

The remaining n-k bits are check bits . Hence eq(2) and (3) can be combined into

[C] = { C1,C2,C3……………………Ck, Ck+1,Ck+2 ……………………Cn}

These (n-k) number of check bits ck+1,Ck+2 ………………. Cn are derived from k-msg bits using predetermined rule below:

$C_{k+1} = P_{11}d_1 + P_{21} d_2 + \ldots\ldots\ldots\ldots\ldots + P_{k1}d_k$

$C_{k+2} = P_{21} d_1 + P_{22} d_2 + \ldots\ldots\ldots\ldots\ldots + P_{k2}d_k$

…………………………………………………….

$C_{k+n} = P_{1,n-k}d_1 + P_{2,n-k} d_2 + \ldots\ldots\ldots\ldots\ldots + P_{k,n-k}d_k$

Where P_{11}, P_{21} …………. Are either 0's or 1's addition is performed using modulo 2 arithmetic

In matrix form

$[c_1 \;\; c_2 \ldots\ldots\ldots c_n] = [d_1, d_2 \ldots\ldots\ldots\ldots d_n]$ [1 0 0 0 P_{11} P_{12} ……….P_{1n-k}]

----

0 0 0 P_{k1}…………………… $P_{k,n-k}$

0r [C] = [D] [G]

$[G] = [I_k \mid P](k \times n)$

Where I_k is the unit matrix of order 'k'

[P] = parity matrix of order kx (n-k)

I = denotes demarkation between I_k and P

The generator matrix can be expressed as

$[G] = [P \mid I_k]$

Associted with the generator matrix [G] another matrix order (n-k) x n matrix is called the parity check matrix.

Given by

$[H] = [P^T \mid I_{n-k}]$

Where P^T represents parity transpose matrix.

Examples

Q1) The generator matrix for a block code is given below. Find all code vectors of this code

$$G = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

A1) The Generator matrix G is generally represented as

$$[G]_{k \times n} = \left[I_{k \times k} \vdots P_{k \times q} \right]_{k \times n}$$

$$\left[I_{k \times k} \vdots P_{k \times q} \right]_{k \times n} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

On comparing,

The number of message bits, k = 3

The number of code word bits, n = 6

The number of check bits, q = n – k = 6 – 3 = 3

Hence, the code is a (6, 3) systematic linear block code. From the generator matrix, we have Identity matrix

$$I_{k \times k} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The coefficient or submatrix,

$$P_{k \times q} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Therefore, the check bits vector is given by,

$$|C|_{1\times q} = [M]_{1\times k}[P]_{k\times q}$$

On substituting the matrix form,

$$[C_1\ C_2\ C_3] = [M_1\ M_2\ M_3]\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 1\\ 1 & 0 & 1\end{bmatrix}$$

From the matrix multiplication, we have

$$C_1 = (1\times m_1) \oplus (0\times m_2) \oplus (1\times m_3)$$
$$C_2 = (0\times m_1) \oplus (1\times m_2) \oplus (0\times m_3)$$
$$C_3 = (0\times m_1) \oplus (1\times m_2) \oplus (1\times m_3)$$

On simplifying, we obtain

$$C_1 = m_1 \oplus m_3$$
$$C_2 = m_2$$
$$C_3 = m_2 \oplus m_3$$

Hence the check bits (c_1 c_2 c_3) for each block of (m_1 m_2 m_3) message bits can be determined.

(i) For the message block of (m1 m2 m3) = (0 0 0), we have

$$C_1 = m_1 \oplus m_3 = 0 \oplus 0 = 0$$
$$C_2 = m_2 = 0$$
$$C_3 = m_2 \oplus m_3 = 0 \oplus 0 = 0$$

(ii) For the message block of (m1 m2 m3) = (0 0 1), we have

$$C_1 = m_1 \oplus m_3 = 0 \oplus 1 = 1$$
$$C_2 = m_2 = 0$$
$$C_3 = m_2 \oplus m_3 = 0 \oplus 1 = 1$$

(iii) For the message block of (m1 m2 m3) = (0 1 0), we have

$C_1 = m_1 \oplus m_3 = 0 \oplus 0 = 0$
$C_2 = m_2 = 1$
$C_3 = m_2 \oplus m_3 = 1 \oplus 0 = 1$

Similarly, we can obtain check bits for other message blocks.

6.2 Syndrome and error detection

Let us suppose that C=(C1,C2………………Cn) be a valid vector transmitted over a noisy communication channel belonging to a (n,k) linear block code.

Let R = { r1,r2,…………………….. Rn} be the received vector . Due to noise in the channel vector r1,r2……………..rn may be different from c1,c2……………………….cn . The error vector or error pattern E is defined as the distance between R and C

E = R-C or E = R+C

Example

For a systematic (6,3) linear block code the parity matrix P is given by

$$P = \begin{matrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{matrix}$$

- Find the possible code vector
- Draw the encoding circuit.
- Draw the syndrome calculation circuit for recorded vector

R = [r1 r2 r3 r4 r5 r6]

Given n=6 and k=3 for (6,3) block code

Since k=3 there are 23 = 8 message vectors given by

(000),(001),(010),(011),(100),(101),(110),(111)

The code vector is found by

[C]= [D][G]

[G] = {Ik|P]= [I3|P]

1 0 0 | 1 0 1

0 1 0 |0 1 1

0 0 1|1 1 0

C = [d1 d2 d3] 1 0 0 | 1 0 1

0 1 0 |0 1 1

0 0 1|1 1 0

C= [d1,d2,d3,(d1+d3),(d2+d3),(d1+d2)]

Code name	Msg Vector			Code Vector For (6,3) liner block code					
	d 2	1 d3	d	d 1	d 2	d 3	d 1+d 3	d2 + d3	d1 + d2
Ca	0	0	0	0	0	0	0	0	0
Cb	0	0	1	0	0	1	1	1	0
Cc	0	1	0	0	1	0	0	1	1

Cd	0	1	1	0	1	1	1	0	1
Ce	1	0	0	1	0	0	1	0	1
Cf	1	0	1	1	0	1	0	1	1
Cg	1	1	0	1	1	0	1	1	0
Ch	1	1	1	1	1	1	0	0	0

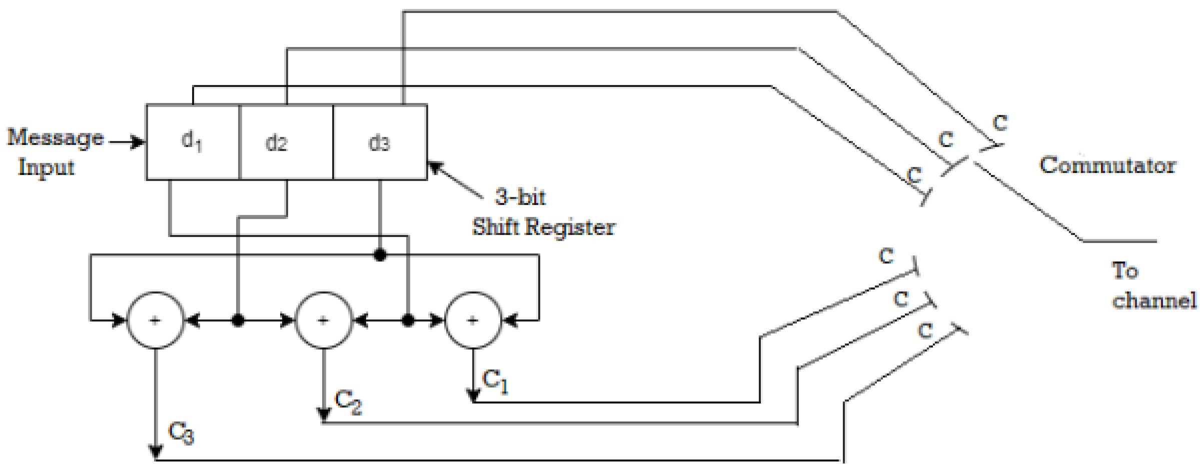
The code vector is given by

$c1 = d1$, $c2 = d2$, $c3 = d3$ $c4 = d1 + d3$ $c5 = d2 + d3$ $c6 = d1 + d2$

Since k=3 we require 3bit shift registers to move the message bits into it.

We have (n-k) = 6-3 = 3.

Hence we require 3-modulo 2 adders and 6 segement commutator.



For the (6,3) code matrix H^T is given by

H^T =

$$\begin{matrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}$$

$H^T = [P/I_{n-k}] = [P/I_3]$

S= [s1 s2 s3] = [r1 r2 r3 r4 r5 r6]

$$\begin{matrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}$$

[S] = [S1 s2 s3] = (r1+r3+r4) , (r2+r3+r5),(r1+r2+r6)

S1 = r1+r3+r4

S2 = r2+r3+r5

S3 = r1 +r2 +r6

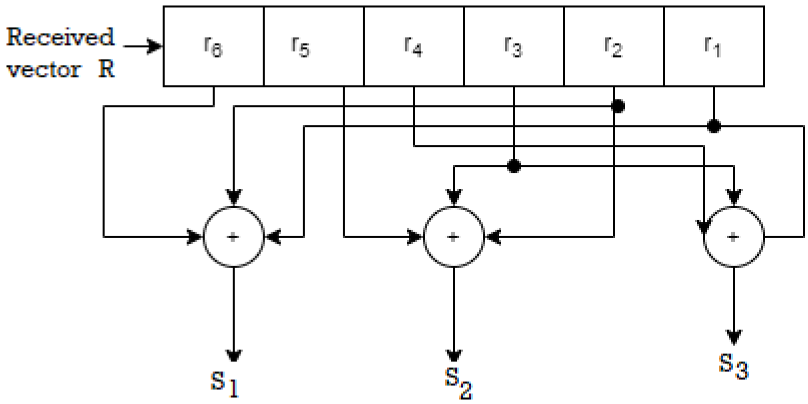


Fig: Syndrome calculation circuit

6.3 Error detection and correction capability

For a systematic (6,3) linear block code the parity marix P is given by

P = 1 0 1

0 1 1

1 1 0

For the recorded code-vector R= [1 1 0 0 1 0] Detect and correct the single error that has occurred due to noise.

Given

P = 1 0 1

0 1 1

1 1 0

P^T = 1 0 1

0 1 1

1 1 0

Parity check matrix [H] = [P^T | I_{n-k}] = [P^T | I_3]

1 0 1 1 0 0

0 1 1 0 1 0

1 1 0 0 0 1

H^T = 1 0 1

0 1 1

1 1 0

1 0 0

0 1 0

0 0 1

S = [s1 s2 s3] = R H^T= [11 00 10] 1 0 1

0 1 1

1 1 0

1 0 0

0 1 0

0 0 1

[S]= [100] since s≠0 it represents error. Since [100] present in the 4^{th} row of H^T . So the error vector [E] = [0 0 0 1 0 0] the the corrected vector is given by

C = R + E = [11 00 10][000100]

C = [110110] which is the valid code

6.4 Standard array and syndrome decoding

Syndrome decoding

An (8,4) binary linear block code is defined by systematic matrices

H =

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

G =

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

Consider two possible messages m1 = [0 1 1 0]

C1 = [0 1 1 0 0 1 10]

m2 = [1 0 1 1]

c2 = [01001011]

Suppose the error pattern e = [00000100] is added to both codewords

r1 = [01100010]

s1 = [1011]

r2 = [01001111]

s2 = [1011]

Both syndromes equal column 6 of H so decoder correct bit 6.

Standard array

Syndrome table decoding can also be described using the standard array. The standard array of a group code C is the coset decomposition of Fth with respect to the subgroup C .

0	c 2	c 3	……	c M
e 2	c 2 + e 2	c 3 + e 2		c M + e2
e 3	c 2 + e3	c 3 + e 3		CM + e3
…….				
e N	c 2 + eN	c 3 + eN		c M + e N

The first row is the code C with zero vector in the first column

Every other row is coset

The n-tuple in the first column is called the coset leader

We usually choose the coset leader o be the most plausible error pattern example the error pattern of smallest weight.

Example

G =

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

1	1	0	0	0	1

H

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

The standard array has 6 coset leaders of weight 1 and weight 2.

000000	001110	010101	011011	100011	101101	110110	111000
000001	001111	010100	011010	100010	101100	110111	111001
000010	001100	010111	111001	100001	101111	110100	111010
000100	001010	010001	011111	100111	101001	110010	111011
001000	000110	011101	010011	101011	100101	111110	110000
010000	011110	000101	001011	110011	111101	100110	101000
100000	101110	110101	111011	000011	001101	010110	011000
001001	000111	011100	010010	101010	100100	111111	110001

Stanadard array decoding

An (n,k) LBC over GF (Q) has $M = Q^k$ codewords.

Every n-tuple appears exactly once in the standard array . Therefore the number of rows N satisfies

$MN = Q^{n} \dashrightarrow N = Q^{n-k}$

All vectors in a row of the standard array have the same syndrome.

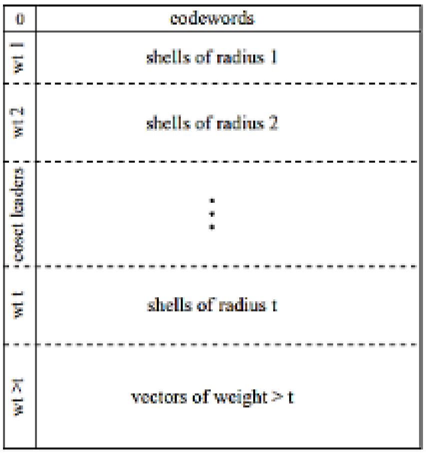
Thus there is one to one correspondence between the rows of the stanadard array and Q^{n-k} syndrome values.

Decoding using the stanadard array is simple

Decode the senseword r to the codeword at the top of the column that contains r.

The decoder subtarcts the coset leader from the received vector to obtain estimated codeword.

The decoded region for codeword is the column headed by that codeword.



6.5 Cyclic Codes: Coding & Decoding

Binary cyclic codes are a subclass of the linear block codes. They have very good features which make them extremely useful. Cyclic codes can correct errors caused by bursts of noise that affect several successive bits. The very good block codes like the Hamming codes, BCH codes and Golay codes are also cyclic codes. A linear block code is called as cyclic code if every cyclic shift of the code vector produces another code vector. A cyclic code exhibits the following two properties.

(i) Linearity Property: A code is said to be linear if modulo-2 addition of any two code words will produce another valid codeword.

(ii) Cyclic Property: A code is said to be cyclic if every cyclic shift of a code word produces another valid code word. For example, consider the n-bit code word, X = (xn-1, xn-2, ….. x1, x0).

If we shift the above code word cyclically to left side by one bit, then the resultant code word is

$X' = (x_{n-2}, x_{n-3}, \ldots\ldots x_1, x_0, x_{n-1})$

Here X' is also a valid code word. One more cyclic left shift produces another valid code vector X''.

$X'' = (x_{n-3}, x_{n-4}, \ldots\ldots x_1, x_0, x_{n-1}, x_{n-2})$

Representation of codewords by a polynomial

• The cyclic property suggests that we may treat the elements of a code word of length n as the coefficients of a polynomial of degree (n-1).

• Consider the n-bit code word, $X = (x_{n-1}, x_{n-2}, \ldots\ldots x_1, x_0)$

This code word can be represented in the form of a code word polynomial as below:

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

Where X (p) is the polynomial of degree (n-1). p is an arbitrary real variable. For binary codes, the coefficients are 1s or 0s.

- The power of 'p' represents the positions of the code word bits. i.e., p^{n-1} represents MSB and p0 represents LSB.
- Each power of p in the polynomial X(p) represents a one-bit cyclic shift in time. Hence, multiplication of the polynomial X(p) by p may be viewed as a cyclic shift or rotation to the right, subject to the constraint that pn = 1.
- We represent cyclic codes by polynomial representation because of the following reasons.

1. These are algebraic codes. Hence algebraic operations such as addition, subtraction, multiplication, division, etc. becomes very simple.

2. Positions of the bits are represented with the help of powers of p in a polynomial.

A) Generation of code vectors in non-systematic form of cyclic codes

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

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

The codeword polynomial X(P) is given as X (P) = M(P). G(P)

Where G(P) is called as the generating polynomial of degree 'q' (parity or check bits q = n – k). The generating polynomial is given as $G(P) = p^q + g_{q-1}p^{q-1} + \ldots\ldots + g_1p + 1$

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

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

$X_1 (P) = M_1 (P) G (P)$

$X_2 (P) = M_2 (P) G (P)$

$X_3 (P) = M_3 (P) G (P)$ and so on

All the above code vectors $X_1, X_2, X_3 \ldots\ldots$ Are in non-systematic form and they satisfy cyclic property.

The generator polynomial of a (7, 4) cyclic code is G(p) = p3 + p + 1. Find all the code vectors for the code in non-systematic form

Here n = 7, k = 4

Therefore, q = n – k = 7 – 4 = 3

Since k = 4, there will be a total of 2^k = 24 = 16 message vectors (From 0 0 0 0 to 1 1 1 1). Each can be coded in to a 7 bits codeword.

(i) Consider any message vector as

$M = (m_3\ m_2\ m_1\ m_0) = (1\ 0\ 0\ 1)$

The general message polynomial is $M(p) = m_3p^3 + m_2p^2 + m_1p + m_0$, for k = 4

For the message vector (1 0 0 1), the polynomial is

$M(p) = 1.\ p^3 + 0.\ p^2 + 0.\ p + 1$

$\Rightarrow M(p) = P^3 + 1$

The given generator polynomial is $G(p) = p^3 + p + 1$

In non-systematic form, the codeword polynomial is $X\ (p) = M\ (p).\ G\ (p)$

On substituting,

$X\ (p) = (p^3 + 1).\ (p^3 + p +1)$

$= p^6 + p^4 + p^3 + p^3 + p +1$

$= p^6 + p^4 + (1 \oplus 1)\ p^3 + p +1$

$= p^6 + p^4 + p + 1$

$= 1.p^6 + 0.p^5 + 1.p^4 + 0.p^3 + 0.p^2 + 1.p + 1$

The code vector corresponding to this polynomial is

$X = (x_6\ x_5\ x_4\ x_3\ x_2\ x_1\ x_0)$

$X = (1\ 0\ 1\ 0\ 0\ 1\ 1)$

(ii) Consider another message vector as

$M = (m_3\ m_2\ m_1\ m_0) = (0\ 1\ 1\ 0)$

The polynomial is $M(p) = 0.p^3 + 1.p^2 + 1.p^1 + 0.1$

$M(p) = p^2 + p$

The codeword polynomial is

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

$\Rightarrow X(p) = (p^2 + p).\ (p^3 + p + 1)$

$= p^5 + p^3 + p^2 + p^4 + p^2 + p$

$= p^5 + p^4 + p^3 + (1 \oplus 1)\, p^2 + p$

$= p^5 + p^4 + p^3 + p$

$= 0.p^6 + 1.p^5 + 1.p^4 + 1.p^3 + 0.p^2 + 1.p + 0.1$

The code vector, X = (0 1 1 1 0 1 0)

Similarly, we can find code vector for other message vectors also, using the same procedure.

B) Generation of code vectors in systematic form of cyclic codes

The code word for the systematic form of cyclic codes is given by

X = (k message bits ⁝q check bits)

$X = (m_{k-1}\, m_{k-2} \,........\, m_1\, m_0 ⁝\, c_{q-1}\, c_{q-2} \,.....\, c_1\, c_0)$

In polynomial form, the check bits vector can be written as

$C(p) = c_{q-1}p^{q-1} + c_{q-2}p^{q-2} + ... + c_1p + c_0$

In systematic form, the check bits vector polynomial is obtained by

$C(p) = \text{rem}\,[p^q.\, M\,(p)/G(p)\,]$

Where:

M(p) is message polynomial

G(p) is generating polynomial

'rem' is remainder of the division

The generator polynomial of a (7, 4) cyclic code is G(p) = p3 + p + 1. Find all the code vectors for the code in systematic form.

Here n = 7, k = 4

Therefore, q = n - k = 7 – 4 = 3

Since k = 4, there will be a total of 2^k = 24 = 16 message vectors (From 0 0 0 0 to 1 1 1 1). Each can be coded into a 7 bits codeword.

(i) Consider any message vector as

$M = (m_3\ m_2\ m_1\ m_0) = (1\ 1\ 1\ 0)$

By message polynomial,

$M(p) = m_3p^3 + m_2p^2 + m^1p + m^0$, for k = 4.

For the message vector (1 1 1 0), the polynomial is

$M(p) = 1.p^3 + 1.p^2 + 1.p + 0.1$

$M(p) = p^3 + p^2 + p$

The given generator polynomial is $G(p) = p^3 + p + 1$

The check bits vector polynomial is

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

$$= rem\left[\frac{p^3(p^3+p^2+p)}{(p^3+p+1)}\right]$$

$$= rem\left[\frac{(p^6+p^5+p^4)}{(p^3+p+1)}\right]$$

We perform division as per the following method.

$$\begin{array}{rl}
 & p^3 + p^2 \\
(p^3+p+1) & \overline{)\ p^6 + p^5 + p^4 + 0.p^3 + 0.p^2 + 0.p + 0.1} \\
 & p^6 + 0.p^5 + p^4 + p^3 \\
\text{(Mod-2 addition)} & \overline{0.p^6 + p^5 + 0.p^4 + p^3 + 0.p^2 + 0.p + 0.1} \\
 & p^5 + 0.p^4 + p^3 + p^2 \\
 & \overline{0.p^5 + 0.p^4 + 0.p^3 + \underbrace{p^2 + 0.p + 0.1}_{\text{remainder}}}
\end{array}$$

Thus, the remainder polynomial is $p^2 + 0.p + 0.1$.

This is the check bits polynomial C(p)

$\therefore c(p) = p^2 + 0.p + 0.1$

The check bits are c = (1 0 0)

Hence the code vector for the message vector (1 1 1 0) in systematic form is

$X = (m_3\ m_2\ m_1\ m_0 \vdots\ c_2\ c_1\ c_0) = (1\ 1\ 1\ 0\ 1\ 0\ 0)$

(ii) Consider another message vector as

$M = (m_3\ m_2\ m_1\ m_0) = (1\ 0\ 1\ 0)$

The message polynomial is $M(p) = p^3 + p$

Then the check bits vector polynomial is

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

$$= rem\left[\frac{p^3(p^3+p)}{(p^3+p+1)}\right]$$

$$= rem\left[\frac{(p^6+p^4)}{(p^3+p+1)}\right]$$

The division is performed as below.

$$\begin{array}{rl}
 & p^3 + 1 \\
(p^3 + p + 1) & \overline{)\ p^6 + 0.p^5 + p^4 + 0.p^3 + 0.p^2 + 0.p + 0.1} \\
 & p^6 + 0.p^5 + p^4 + p^3 \\
\text{(Mod-2 addition)} & \overline{0.p^6 + 0.p^5 + 0.p^4 + p^3 + 0.p^2 + 0.p + 0.1} \\
 & p^3 + 0.p^2 + p + 1 \\
 & \overline{\underbrace{0.p^3 + 0.p^2 + p + 1}_{\text{remainder}}}
\end{array}$$

Thus, the check bits polynomial is $C(p) = 0.p^2 + 1.p + 1.1$

The check bits are C = (0 1 1)

Hence the code vector is X= (1 0 1 0 0 1 1)

Similarly, we can find code vector for other message vectors also, using the same procedure.

Cyclic Redundancy Check Code (CRC)

Cyclic codes are extremely well-suited for error detection. Because they can be designed to detect many combinations of likely errors. Also, the implementation of both encoding and error detecting circuits is practical. For these reasons, all the error detecting codes used in practice are of cyclic code type. Cyclic Redundancy Check (CRC) code is the most important cyclic code used for error detection in data networks & storage systems. CRC code is basically a systematic form of cyclic code.

CRC Generation (encoder)

The CRC generation procedure is shown in the figure below.

- Firstly, we append a string of 'q' number of 0s to the data sequence. For example, to generate CRC-6 code, we append 6 number of 0s to the data.
- We select a generator polynomial of (q+1) bits long to act as a divisor. The generator polynomials of three CRC codes have become international standards. They are
    - CRC – 12 code: $p^{12} + p^{11} + p^3 + p^2 + p + 1$
    - CRC – 16 code: $p^{16} + p^{15} + p^2 + 1$
    - CRC – CCITT Code: $p^{16} + p^{12} + p^5 + 1$

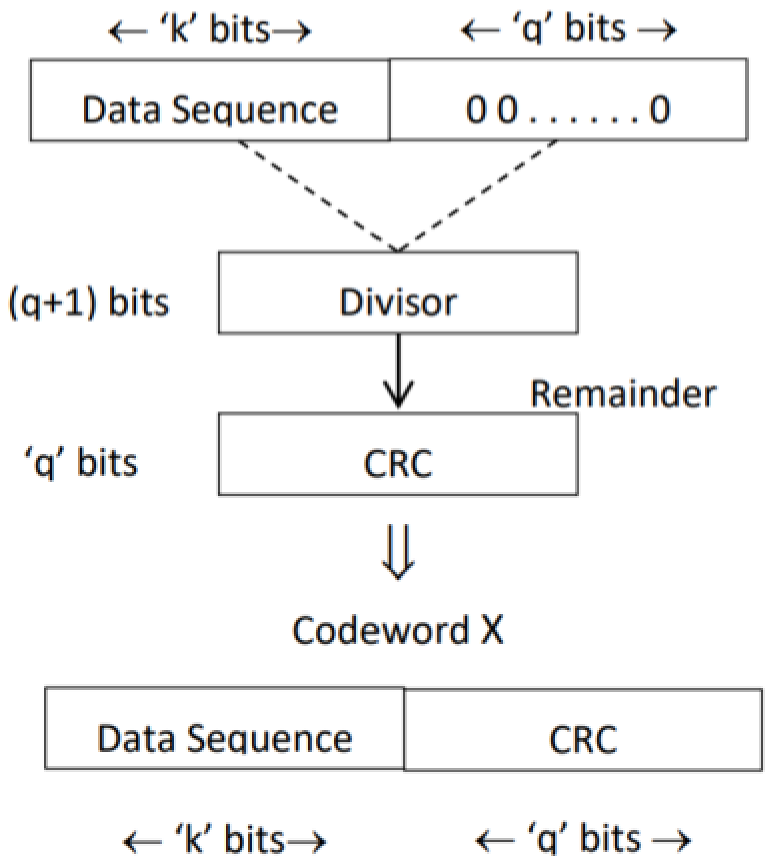


Fig: CRC Generation

- We divide the data sequence appended with 0s by the divisor. This is a binary division.
- The remainder obtained after the division is the 'q' bit CRC. Then, this 'q' bit CRC is appended to the data sequence. Actually, CRC is a sequence of redundant bits.
- The code word generated is now transmitted.

CRC checker

The CRC checking procedure is shown in the figure below. The same generator polynomial (divisor) used at the transmitter is also used at the receiver.

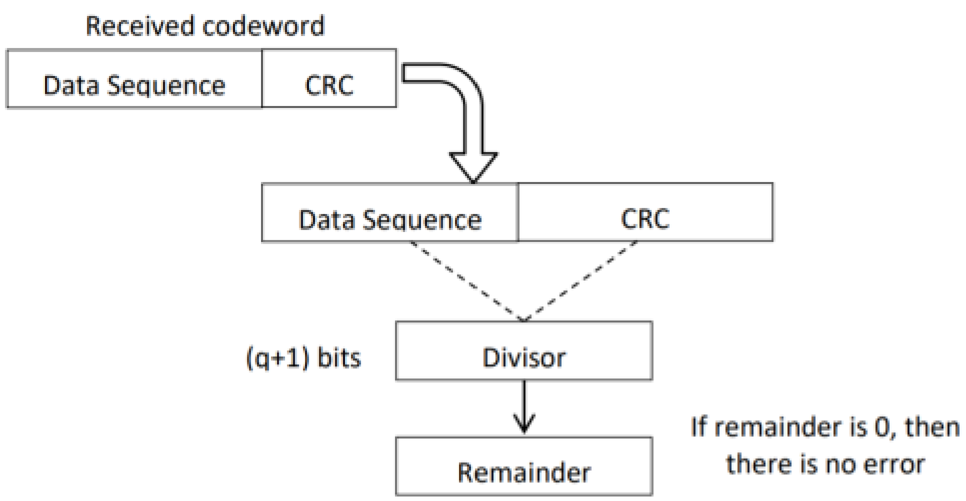


Fig: CRC checker

- We divide the received code word by the divisor. This is also a binary division.

- If the remainder is all 0s, then there are no errors in the received codeword, and hence must be accepted.
- If we have a non-zero remainder, then we infer that error has occurred in the received code word. Then this received code word is rejected by the receiver and an ARQ signaling is done to the transmitter.

1. Generate the CRC code for the data word of 1 1 1 0. The divisor polynomial is p3 + p + 1

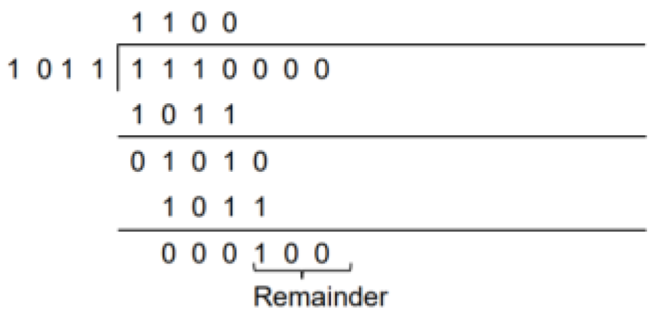
Data Word (Message bits) = 1 1 1 0

Generator Polynomial (divisor) = $p^3 + p + 1$

Divisor in binary form = 1 0 1 1

The divisor will be of (q + 1) bits long. Here the divisor is of 4 bits long. Hence q = 3. We append three 0s to the data word.

Now the data sequence is 1 1 1 0 0 0 0. We divide this data by the divisor of 1 0 1 1. Binary division is followed.



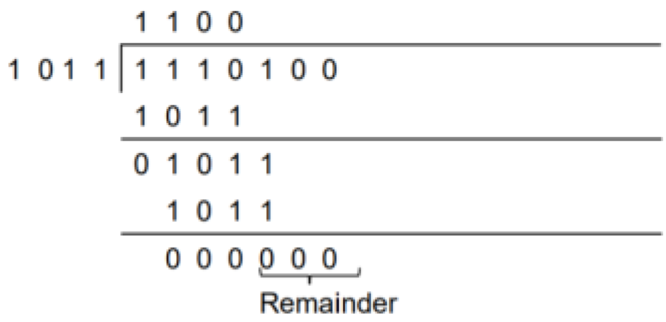
The remainder obtained from division is 100. Then the transmitted codeword is 1 1 1 0 1 0 0.

2. A codeword is received as 1 1 1 0 1 0 0. The generator (divisor) polynomial is $p^3 + p + 1$. Check whether there is error in the received codeword.

Received Codeword = 1 1 1 0 1 0 0

Divisor in binary form = 1 0 1 1

We divide the received codeword by the divisor.



The remainder obtained from division is zero. Hence there is no error in the received codeword.

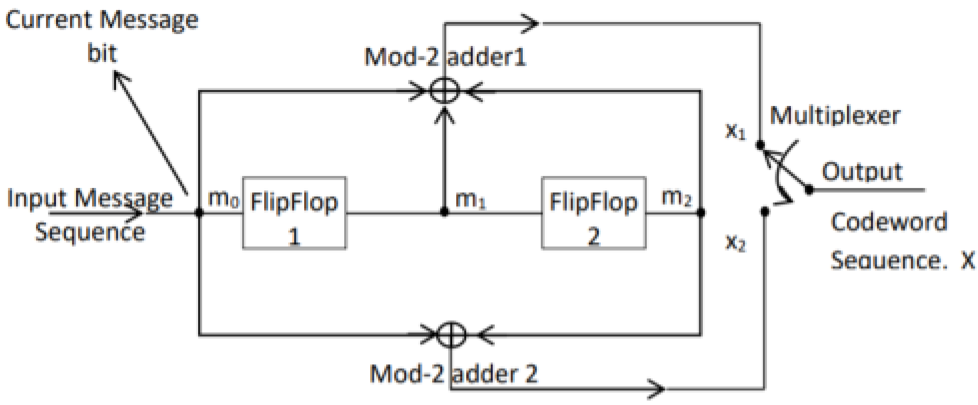
Key takeaway

- Cyclic codes can correct burst errors that span many successive bits.
- They have an excellent mathematical structure. This makes the design of error correcting codes with multiple-error correction capability relatively easier.
- The encoding and decoding circuits for cyclic codes can be easily implemented using shift registers.
- The error correcting and decoding methods of cyclic codes are simpler and easy to implement. These methods eliminate the storage (large memories) needed for lookup table decoding. Therefore, the codes become powerful and efficient.

6.6 Convolutional Codes: Coding & Decoding

In block coding, the encoder accepts a k-bit message block and generates an n-bit code word. Thus, code words are produced on a block-by-block basis. Therefore, a buffer is required in the encoder to place the message block. A subclass of Tree codes is convolutional codes. The convolutional encoder accepts the message bits continuously and generates the encoded codeword sequence continuously. Hence there is no need for buffer. But in convolutional codes, memory is required to implement the encoder.

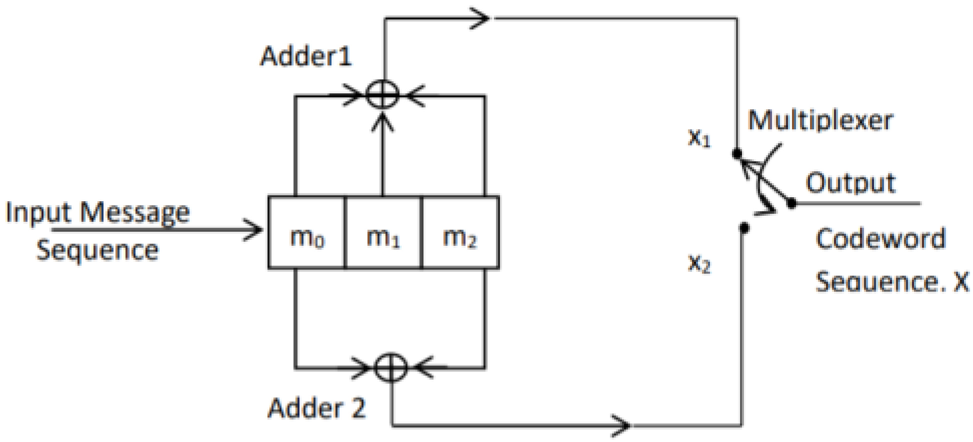
Fig:



Convolutional Encoder

In a convolutional code, the encoding operation is the discrete-time convolution of the input data sequence with the impulse response of the encoder. The input message bits are stored in the fixed length shift register and they are combined with the help of mod-2 adders. This operation is equivalent to binary convolution and hence it is called convolutional coding. The figure above shows the connection diagram for an example convolutional encoder.

Fig:



Convolutional Encoder redrawn alternatively

The encoder of a binary convolution code may be viewed as a finite-state machine. It consists of M-stage shift register with prescribed connections to modulo2 adders. A multiplexer serializes the outputs of the adders. The convolutional codes generated by these encoders of Figure above are non-systematic form. Consider that the current message bit is shifted to position m0. Then m_1 and m_2 store the previous two message bits. Now, by mod-2 adders 1 and 2 we get the new values of X_1 and X_2. We can write

$X_1 = m_0 \oplus m_1 \oplus m_2$ and

$X_2 = m_0 \oplus m_2$

The multiplexer switch first samples X_1 and then X_2. Then next input bit is taken and stored in m_0. The shift register then shifts the bit already in m_0 to m_1. The bit already in m_1 is shifted to m_2. The bit already in m_2 is discarded. Again, X_1 and X_2 are generated according to this new combination of m_0, m_1 and m_2. This process is repeated for each input message bit. Thus, the output bit stream for successive input bits will be,

$X = x_1\ x_2\ x_1\ x_2\ x_1\ x_2$…….. And so on

In this convolutional encoder, for every input message bit, two encoded output bits X1 and X2 are transmitted. Hence number of message bits, k = 1. The number of encoded output bits for one message bit, n = 2.

Code rate: The code rate of this convolutional encoder is given by

Code rate, r = $Message\ bits\ (k)/encoder\ output\ bits\ (n) = k/n$ = 1/2 where $0 < r < 1$

Constraint Length: The constraint length (K) 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 Figure above, constraint length is K = 3 bits. Because, 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. For the encoder of Figure above, whenever a particular message bit enters the shift register, it remains in the shift register for three shifts i.e.,

First Shift → Message bit is entered in position m_0.

Second Shift → Message bit is shifted in position m_1.

Third Shift → Message bit is shifted in position m_2.

Constraint length ‘K’ is also equal to one plus the number of shift registers required to implement the encoder.

Dimension of the Code

The code dimension of a convolutional code depends on the number of message bits ‘k’, the number of encoder output bits, ‘n’ and its constraint length ‘K’. The code dimension is therefore represented by (n, k, K). For the encoder shown in figure above, the code dimension is given by (2, 1, 3) where n = 2, k = 1 and constraint length K = 3.

Graphical representation of convolutional codes

Convolutional code structure is generally presented in graphical form by the following three equivalent ways.

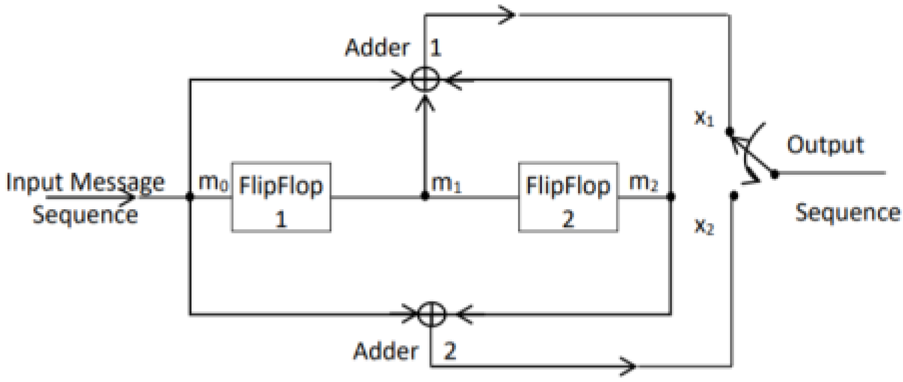
1. By means of the state diagram

2. By drawing the code trellis

3. By drawing the code tree

These methods can be better explained by using an example.

For the convolutional encoder given below in Figure, determine the following. a) Code rate b) Constraint length c) Dimension of the code d) Represent the encoder in graphical form.



a) Code rate:

The code rate, r $= k/n$. The number of message bits, k = 1.

The number of encoder output bits, n = 2.

Hence code rate, r = 1/2

b) Constraint length:

Constraint length, k = 1 + number of shift registers.

Hence k = 1 + 2 = 3

c) Code dimension:

Code dimension = (n, k, K) = (2, 1, 3)

Hence the given encoder is of ½ rate convolutional encoder of dimension (2, 1, 3).

d) Graphical form representation

The encoder output is X = (x_1 x_2 x_1 x_2 x_1 x_2 ….. And so on)

The Mod-2 adder 1 output is $x_1 = m_0 \oplus m_1 \oplus m_2$

The Mod-2 adder 2 output is $x_2 = m_0 \oplus m_2$.

We can represent the encoder output for possible input message bits in the form of a Logic table.

	Input message bit	Present state		Next state		Encoder output	
	m_0	m_1	m_2	m_1	m_2	x_1	x_2
A	0	0	0	0	0	0	0

	1	0	0	1	0	1	1
B	0	1	0	0	1	1	0
	1	1	0	1	1	0	1
C	0	0	1	0	0	1	1
	1	0	1	1	0	0	0
D	0	1	1	0	1	0	1
	1	1	1	1	1	1	0

Output $x_1 = m_0 \oplus m_1 \oplus m_2$ and $x_2 = m_0 \oplus m_2$

- The encoder output depends on the current input message bit and the contents in the shift register i.e., the previous two bits.
- The present condition of the previous two bits in the shift register may be in four combinations. Let these combinations 00, 10, 01 and 11 be corresponds to the states A, B, C and D respectively.
- For each input message bit, the present state of the m1 and m2 bits will decide the encoded output.
- The logic table presents the encoded output x1 and x2 for the possible '0' or '1' bit input if the present state is A or B or C or D.

1. State diagram representation

- The state of a convolutional encoder is defined by the contents of the shift register. The number of states is given by 2^k-1 = 23-1 = 22 = 4. Here K represents the constraint length.
- Let the four states be A = 00, B = 10, C = 01 and D = 11 as per the logic table. A state diagram as shown in the figure below illustrates the functioning of the encoder.
- Suppose that the contents of the shift register is in the state A = 00. At this state, if the incoming message bit is 0, the encoder output is X = (x1 x2) = 00. Then if the m0, m1 and m2 bits are shifted, the contents of the shift

register will also be in state A = 00. This is represented by a solid line path starting from A and ending at A itself.

- At the 'A' state, if the incoming message bit is 1, then the encoder output is X = 11. Now if the m0, m1 and m2 bits are shifted, the contents of the shift register will become the state B = 10. This is represented by a dashed line path starting from A and ending at B.
- Similarly, we can draw line paths for all other states, as shown in the Figure below.

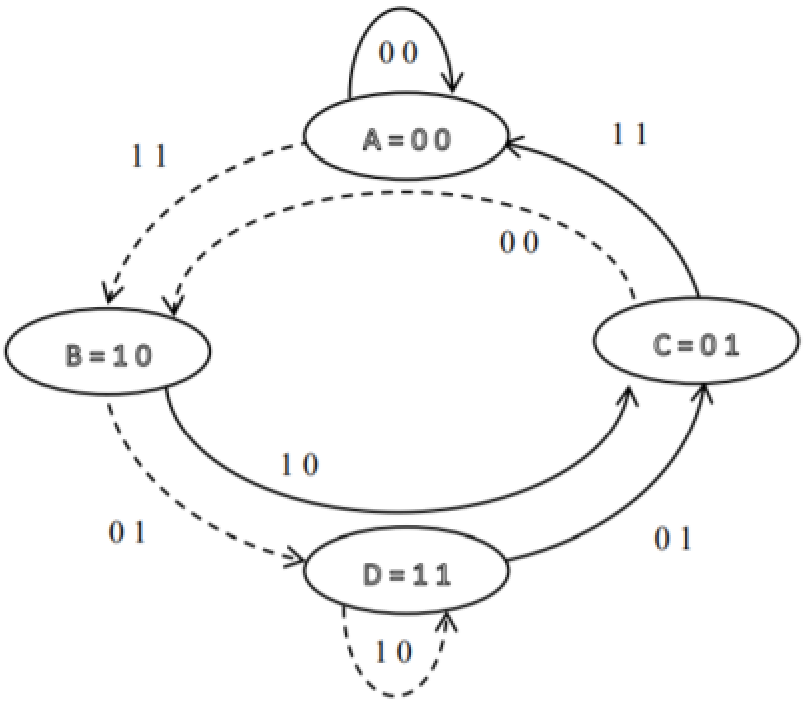


Fig: State Diagram

2. Code tree representation

- The code tree diagram is a simple way of describing the encoding procedure. By traversing the diagram from left to right, each tree branch depicts the encoder output codeword.
- Figure below shows the code representation for this encoder.
- The code tree diagram starts at state A =00. Each state now represents the node of a tree. If the input message bit is m0 = 0 at node A, then path of the tree goes upward towards node A and the encoder output is 00. Otherwise, if the input message bit is m_0 = 1, the path of the tree goes down towards node B and the encoder output is 11.
- Similarly depending upon the input message bit, the path of the tree goes upward or downward. On the path between two nodes the outputs are shown.
- In the code tree, the branch pattern begins to repeat after third bit, since particular message bit is stored in the shift registers of the encoder for three shifts.

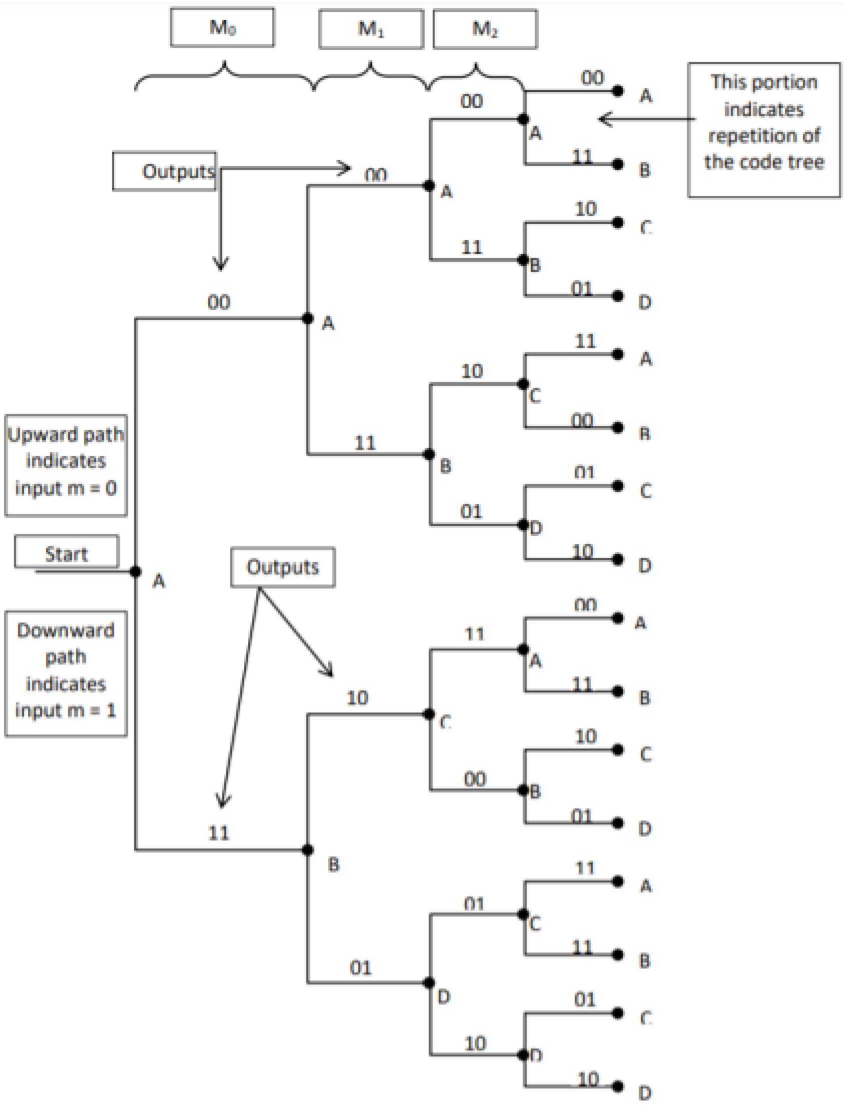


Fig: Code Tree Representation

3. Code trellis representation

- Code trellis is the more compact representation of the code tree. In the code tree there are four states (or nodes). Every state goes to some other state depending upon the input message bit.
- Code trellis represents the single and unique diagram for such steady state transitions. The Figure below shows the code trellis diagram.
- The nodes on the left denote four possible current states and those on the right represents next state. The solid transition line represents for input message $m_0 = 0$ and dashed line represents input message $m_0 = 1$.
- Along with each transition line, the encoder output x_1 x_2 is represented during that transition.

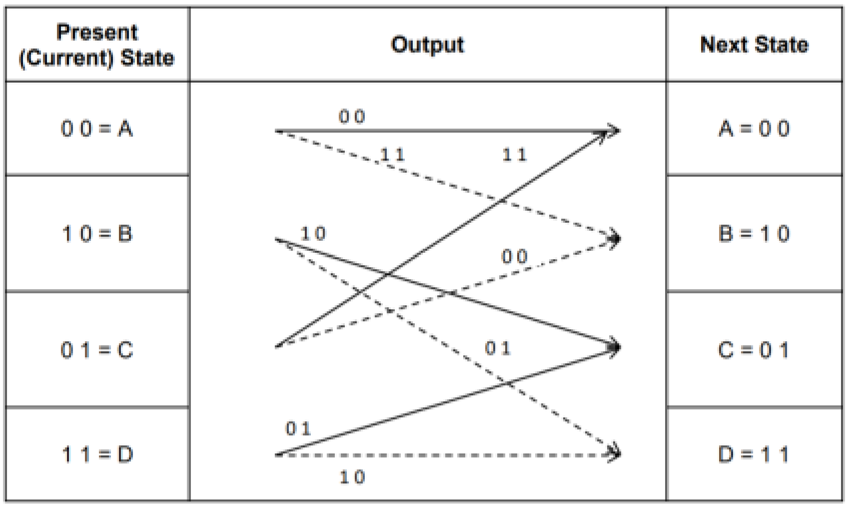


Fig: Code Trellis

Advantages of convolutional codes

- The convolutional codes operate on smaller blocks of data. Hence decoding delay is small.
- The storage hardware required is less.

Disadvantages of convolutional codes

- Due to complexity, the convolutional codes are difficult to analyze.
- These codes are not developed much as compared to block codes.

Key takeaway

Comparison between Linear Block codes and Convolutional codes

Sr. No.	**Linear Block Codes**	**Convolutional codes**
1.	Block codes are generated by $X = MG \;\; or$ $X(p) = M(p)G(p)$	Convolutional codes are generated by convolution between message sequencing and generating sequence.
2.	For a block of message bits, encoded block (code vector) is generated	Each message bits are encoded separately. For every message bit, two or more encoded bits are generated.
3.	Coding is block by block.	Coding is bit by bit

4.	Syndrome decoding is used for most likelihood decoding.	Viterbi decoding is used for most likelihood decoding.
5.	Generator matrices, parity check matrices and syndrome vectors are used for analysis.	Code tree, code trellis and state diagrams are used for analysis.
6.	Generating polynomial and generator matrix are used to get code vectors.	Generating sequences are used to get code vectors.
7.	Error correction and detection capability depends upon minimum distance d_{min} .	Error correction and detection capability depends upon free distance d_{min} .

6.7 Introduction to Turbo Codes & LDPC Codes

Introduction to Turbo Codes

The use of a good code with random-like properties is basic to turbo coding. In the first successful implementation of turbo codes11, Berrouet al. Achieved this design objective by using concatenated codes. The original idea of concatenated codes was conceived by Forney (1966). To be more specific, concatenated codes can be of two types: parallel or serial. The type of concatenated codes used by Berrou et al. Was of the parallel type, which is discussed in this section

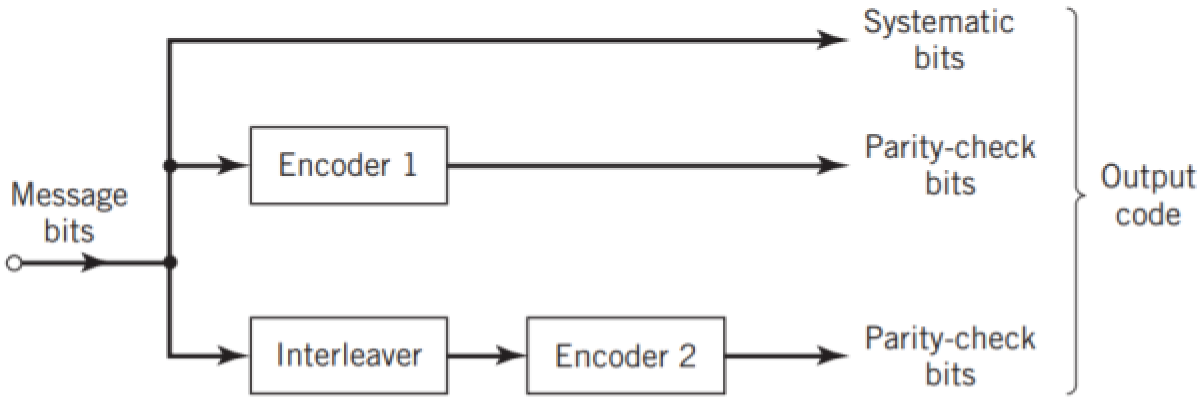


Fig: Block diagram of turbo encoder of the parallel type.

Figure above depicts the most basic form of a turbo code generator that consists of two constituent systematic encoders, which are concatenated by means of an interleaver. The interleaver is an input–output mapping device that permutes the ordering of a sequence of symbols from a fixed alphabet in a completely deterministic manner; that is, it takes the symbols at the input and produces identical symbols at the output but in a different temporal order. Turbo codes use a pseudo-random interleaver, which operates only on the systematic (i.e., message) bits. The size of the interleaver used in turbo codes is typically very large, on the order of several thousand bits. There are two reasons for the use of an interleaver in a turbo code:

1. The interleaver ties together errors that are easily made in one half of the turbo code to errors that are exceptionally unlikely to occur in the other half; this is indeed one reason why the turbo code performs better than a traditional code.

2. The interleaver provides robust performance with respect to mismatched decoding, a problem that arises when the channel statistics are not known or have been incorrectly specified.

Ordinarily, but not necessarily, the same code is used for both constituent encoders in Figure above. The constituent codes recommended for turbo codes are short constraint length RSC codes. The reason for making the convolutional codes recursive (i.e., feeding one or more of the tap outputs in the shift register back to the input) is to make the internal state of the shift register depend on past outputs. This affects the behavior of the error patterns, with the result that a better performance of the overall coding strategy is attained.

Two-State Turbo Encoder Figure below shows the block diagram of a specific turbo encoder using an identical pair of two-state RSC constituent encoders. The generator matrix of each constituent encoder is given by

G(D)= (1, 1/1+D)

The input sequence of bits has length, made up of three message bits and one termination bit. The input vector is given by

m= (m_0 m_1 m_2 m_3)

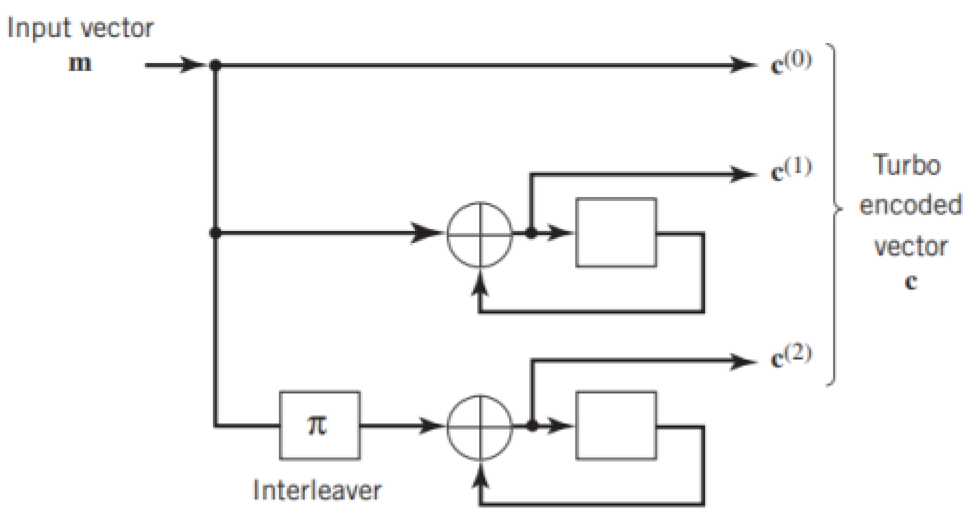


Fig: Two-state turbo encoder for Example

The parity-check vector produced by the first constituent encoder is given by

$$b^{(1)} = \left(b_0^{(1)}, b_1^{(1)}, b_2^{(1)}, b_3^{(1)}\right)$$

Similarly, the parity-check vector produced by the second constituent encoder is given by

$$b^{(2)} = \left(b_0^{(2)}, b_1^{(2)}, b_2^{(2)}, b_3^{(2)}\right)$$

The transmitted code vector is therefore defined by

$$c = \left(c^{(0)}, c^{(1)}, c^{(2)}\right)$$

With the convolutional code being systematic, we thus have

$$c^{(0)} = m$$

As for the remaining two sub-vectors constituting the code vector c, they are defined by $c^{(1)} = b^{(1)}$ and $c^{(2)} = b^{(2)}$.

The transmitted code vector c is therefore made up of 12 bits. However, recalling that the termination bit m_3 is not a message bit, it follows that the code rate of the turbo code

R=3/12=1/4

Turbo Decoder

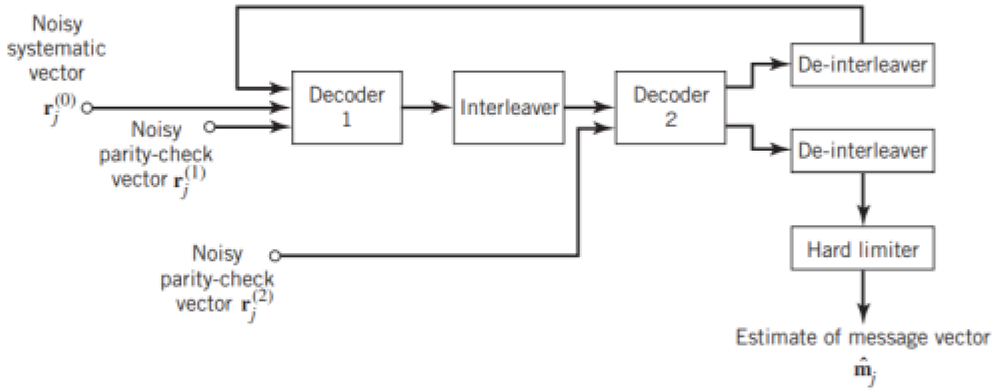
Figure 11(a) below shows the block diagram of the two-stage turbo decoder. The decoder operates on noisy versions of the systematic bits and the two sets of parity-check bits in two decoding stages to produce an estimate of the original message bits. A distinctive feature of the turbo decoder that is immediately apparent from the block diagram of Figure11(a) is the use of feedback, manifesting itself in producing extrinsic information from one decoder to the next in an iterative manner. In a way, this decoding process is analogous to the feedback of exhaust gases experienced in a turbo-charged engine; indeed, turbo codes derive their name from this analogy. In other words, the term "turbo" in turbo codes has more to do with the decoding rather than the encoding process. In operational terms, the turbo encoder in Figure11(a) operates on noisy versions of the following inputs, obtained by demultiplexing the channel output, r_j

- Systematic (i.e., message) bits, denoted by $r_j^{(0)}$
- Parity-check bits corresponding to encoder 1 in Figure encoder, denoted by $r_j^{(1)}$
- Parity-check bits corresponding to encoder 2 in Figure encoder, denoted by $r_j^{(2)}$

The net result of the decoding algorithm, given the received vector r_j, is an estimate of the original message vector, namely, which is delivered at the decoder output to the user. Another important point to note in the turbo decoder of Figure 11(a) is the way in which the interleaver and de-interleaver are positioned inside the feedback loop. Bearing in mind the fact that the definition of extrinsic information requires the use of intrinsic information, we see that decoder 1 operates on three inputs:

- The noisy systematic (i.e., original message) bits,
- The noisy parity-check bits due to encoder 1, and
- De-interleaved extrinsic information computed by decoder 2

(a)



(b)

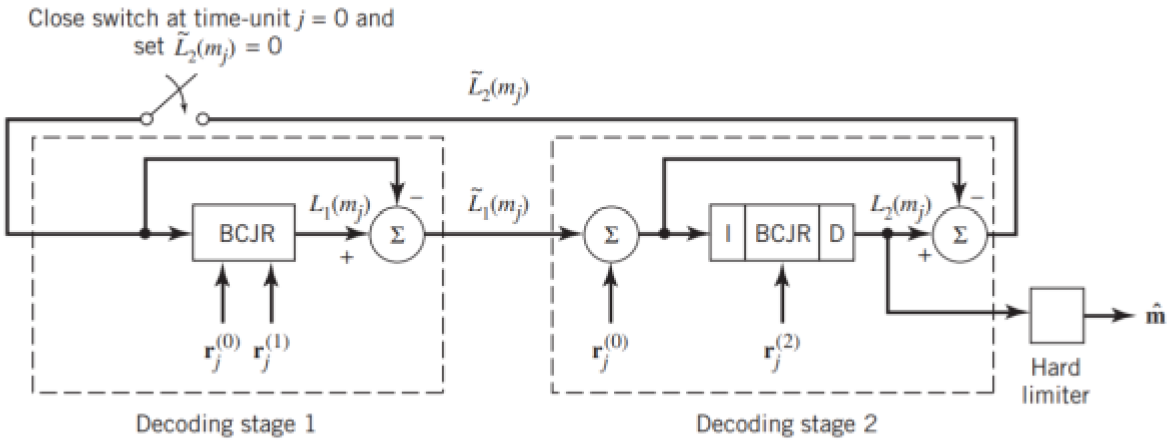


Fig: (a) Block diagram of turbo decoder. (b) Extrinsic form of turbo decoder, where I stand for interleaver, D for deinterleaver, and BCJR for BCJR for BCJR algorithm for log-MAP decoding.

In a complementary manner, decoder 2 operates on two inputs of its own:

- The noisy parity-check bits due to encoder 2 and
- The interleaved version of the extrinsic information computed by decoder 1.
- For this iterative exchange of information between the two decoders inside the feedback loop to continuously reinforce each other, the de-interleaver and interleaver would have to separate the two decoders in the manner depicted in Figure a. Moreover, the structure of the decoder in the receiver is configured to be consistent with the structure of the encoder in the transmitter.

LDPC Codes

The two most important advantages of LDPC codes over turbo codes are:

- Absence of low-weight codewords and
- Iterative decoding of lower complexity.

With regard to the issue of low-weight codewords, we usually find that a small number of codewords in a turbo codeword are undesirably close to the given codeword. Owing to this closeness in weights, once in a while the channel noise causes the transmitted codeword to be mistaken for a nearby code.

In contrast, LDPC codes can be easily constructed so that they do not have such low-weight codewords and they can, therefore, achieve vanishingly small BERs. Turning next to the issue of decoding complexity, we note that the computational complexity of a turbo decoder is dominated by the MAP algorithm, which operates on the trellis for representing the convolutional code used in the encoder. The number of computations in each recursion of the MAP algorithm scales linearly with the number of states in the trellis. Commonly used turbo codes employ trellises with 16 states or more. In contrast, LDPC codes use a simple parity-check trellis that has just two states. Consequently, the decoders for LDPC codes are significantly simpler to design than those for turbo decoders. However, a practical objection to the use of LDPC codes is that, for large block lengths, their encoding complexity is high compared with turbo codes. It can be argued that LDPC codes and turbo codes complement each other, giving the designer more flexibility in selecting the right code for extraordinary decoding performance.

Construction of LDPC Codes

LDPC codes are specified by a parity-check matrix denoted by A, which is purposely chosen to be sparse; that is, the code consists mainly of 0s and a small number of 1s. In particular, we speak of (n, tc, tr) LDPC codes, where n denotes the block length, tc denotes the weight (i.e., number of 1s) in each column of the matrix A, and tr denotes the weight of each row with tr > tc. The rate of such an LDPC code is defined by

$$r = 1 - \frac{t_c}{t_r}$$

Whose validity may be justified as follows. Let ρ denote the density of 1s in the parity check matrix A.

$$t_c = \rho(n - k)$$

$t_r = \rho n$

Where (n – k) is the number of rows in A and n is the number of columns (i.e., the block length). Therefore, dividing tc by tr, we get

$$\frac{t_c}{t_r} = 1 - \frac{k}{n}$$

The structure of LDPC codes is well portrayed by bipartite graphs, which were introduced by Tanner (1981) and, therefore, are known as Tanner graphs. Figure below shows such a graph for the example code of n = 10, tc = 3, and tr = 5. The left-hand nodes in the graph are variable (symbol) nodes, which correspond to elements of the codeword. The right-hand nodes of the graph are check nodes, which correspond to the set of parity-check constraints satisfied by codewords in the code. LDPC codes of the type exemplified by the graph of Figure below are said to be regular, in that all the nodes of a similar kind have exactly the same degree. In Figure below, the degree of the variable nodes is tc = 3 and the degree of the check nodes is tr = 5. As the block length n approaches infinity, each check node is connected to a vanishingly small fraction of variable nodes; hence the term "low density."

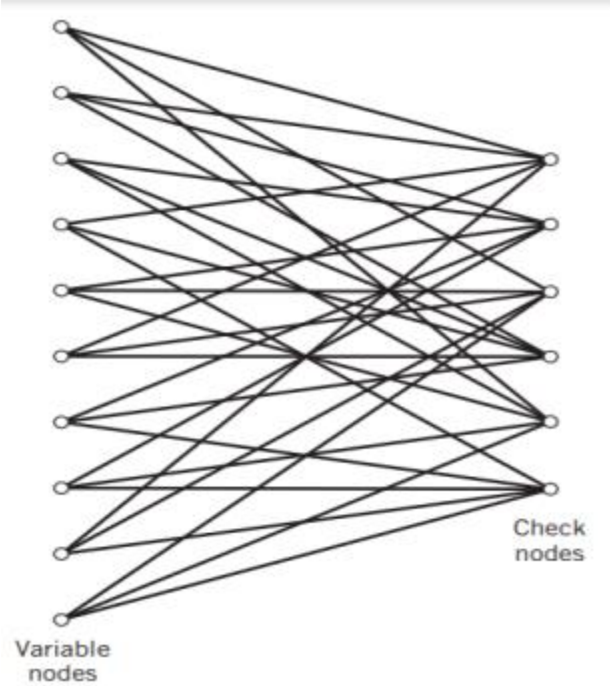


Fig: Bipartite graph of the (10, 3, 5) LDPC code

The matrix A is constructed by putting 1s in A at random, subject to the regularity constraints:

- Each column of matrix A contains a small fixed number tc of 1s;
- Each row of the matrix contains a small fixed number tr of 1s.

In practice, these regularity constraints are often violated slightly in order to avoid having linearly dependent rows in the parity-check matrix A.

The parity-check matrix A of LDPC codes is not systematic (i.e., it does not have the parity-check bits appearing in diagonal form); hence the use of a symbol different from that used in Section 10.4. Nevertheless, for coding purposes, we may derive a generator matrix G for LDPC codes by means of Gaussian elimination performed in modulo-2 arithmetic. The 1-by-n code vector c is first partitioned as shown by

$$c = [b \mid m]$$

Where m is the k-by-1 message vector and b is the (n – k)-by-1 parity-check vector. Correspondingly, the parity-check matrix A is partitioned as

$$A^T = \begin{bmatrix} A_1 \\ --- \\ A_2 \end{bmatrix}$$

Where A1 is a square matrix of dimensions (n – k) × (n – k) and A2 is a rectangular matrix of dimensions k × (n – k); transposition symbolized by the superscript T is used in the partitioning of matrix A for convenience of presentation. Imposing a constraint on the LDPC code we may write

$$[b \mid m]\begin{bmatrix} A_1 \\ --- \\ A_2 \end{bmatrix} = 0$$

$$BA_1 + mA_2 = 0$$

The vectors m and b are related by

$$b=mP$$

The coefficient matrix of LDPC codes satisfies the condition

$$PA_1 + A_2 = 0$$

For matrix P, we get

$$P = A_2A_1^{-1}$$

The generator matrix of LDPC codes is defined by

$$G = [P \mid I_k] = [A_2A_1^{-1} \mid I_k]$$

Consider the Tanner graph pertaining to a (10, 3, 5) LDPC code. The parity-check matrix of the code is defined by

$$A = \begin{bmatrix} \underbrace{\begin{matrix} 1&1&0&1&0&1 \\ 0&1&1&0&1&0 \\ 1&0&0&0&1&1 \\ 0&1&1&1&0&1 \\ 1&0&1&0&1&0 \\ 0&0&0&1&0&0 \end{matrix}}_{A_1^T} & \underbrace{\begin{matrix} 0&0&1&0 \\ 1&1&0&0 \\ 0&0&1&1 \\ 1&0&0&0 \\ 0&1&0&1 \\ 1&1&1&1 \end{matrix}}_{A_2^T} \end{bmatrix}$$

Which appears to be random, while maintaining the regularity constraints: tc = 3 and tr = 5. Partitioning the matrix A in the manner just described, we write

$$A_1 = \begin{bmatrix} 1&0&1&0&1&0 \\ 1&1&0&1&0&0 \\ 0&1&0&1&1&0 \\ 1&0&0&1&1&0 \\ 0&1&1&0&1&0 \\ 1&0&1&1&0&0 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 0&1&0&1&0&1 \\ 0&1&0&0&1&1 \\ 1&0&1&0&0&1 \\ 0&0&1&0&1&1 \end{bmatrix}$$

To derive the inverse of matrix A_1

$$\Big[\underbrace{b_0, b_1, b_2, b_3, b_4, b_5}_{b}\Big] \begin{bmatrix} \underbrace{\begin{matrix} 1&0&1&0&1&0 \\ 1&1&0&1&0&0 \\ 0&1&0&1&1&0 \\ 1&0&0&1&1&0 \\ 0&1&1&0&1&0 \\ 1&0&1&1&0&0 \end{matrix}}_{A_1} \end{bmatrix} = \Big[\underbrace{u_0, u_1, u_2, u_3, u_4, u_5}_{u = mA_2}\Big]$$

Where we have introduced the vector u to denote the matrix product mA_2. By using Gaussian elimination, modulo-2, the matrix A_1 is transformed into lower diagonal form (i.e., all the elements above the main diagonal are zero), as shown by

$$A_1 \rightarrow \begin{bmatrix} 1&0&0&0&0&0 \\ 1&1&0&0&0&0 \\ 0&1&1&0&0&0 \\ 1&0&1&1&0&0 \\ 0&1&0&1&1&0 \\ 1&0&0&1&0&1 \end{bmatrix}$$

This transformation is achieved by the following modulo-2 additions performed on the columns of square matrix A1:

- Columns 1 and 2 are added to column 3;
- Column 2 is added to column 4;
- Columns 1 and 4 are added to column 5;
- Columns 1, 2, and 5 are added to column 6.

Correspondingly, the vector u is transformed as shown by

$$u \rightarrow [u_0, u_1, u_0 + u_1 + u_2, u_1 + u_3, u_0 + u_3 + u_4, u_0 + u_1 + u_4 + u_5]$$

Accordingly, pre multiplying the transformed matrix A1 by the parity vector b, using successive eliminations in modulo-2 arithmetic working backwards and putting the solutions for the elements of the parity vector b in terms of the elements of the vector u in matrix form, we get

$$\underbrace{[u_0, u_1, u_2, u_3, u_4, u_5]}_{u} \begin{bmatrix} \underbrace{\begin{matrix} 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 \end{matrix}}_{A_1^{-1}} \end{bmatrix} = \underbrace{[b_0, b_1, b_2, b_3, b_4, b_5]}_{b}$$

The inverse of matrix A1 is therefore

$$A_1^{-1} = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$

Using the given value of A_2 and A_1^{-1} the value of just found, the matrix product $A_2A_1^{-1}$ is given by

$$A_2A_1^{-1} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}$$

LDPC code is defined by

$$G = \begin{bmatrix} \underbrace{\begin{matrix} 1&0&0&1&1&0 \\ 0&0&0&1&1&1 \\ 0&0&1&1&1&0 \\ 0&1&0&1&1&0 \end{matrix}}_{A_2A_1^{-1}} & \Big| & \underbrace{\begin{matrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \end{matrix}}_{I_k} \end{bmatrix}$$

Key takeaway

In practice, the block length n is orders of magnitude larger than that considered in this example. Moreover, in constructing the matrix A, we may constrain all pairs of columns to have a matrix overlap (i.e., inner product of any two columns in matrix A) not to exceed one; such a constraint, over and above the regularity constraints, is expected to improve the performance of LDPC codes.

References:

1. Bernard Sklar, Prabitra Kumar Ray, "Digital Communications Fundamentals and Applications", Pearson Education, 2nd Edition

2. Wayne Tomasi, "Electronic Communications System", Pearson Education, 5th Edition

3. A.B Carlson, P B Crully, J C Rutledge, "Communication Systems", Tata McGraw Hill Publication, 5th Edition

4. Simon Haykin, "Communication Systems", John Wiley & Sons, 4th Edition

5. Simon Haykin, "Digital Communication Systems", John Wiley & Sons, 4th Edition.