Module 1

Introduction to Digital Communications and Information Theory

Lesson 1

Introduction to Digital Communications

After reading this lesson, you will learn about

- ***Lesson-wise organization of this course***
- ***Schematic description of a representative digital communication system***
- ***Milestones in the history of electronic communications***
- ***Names and usage of electromagnetic bands***
- ***Typical transmission loss for several physical media***

Preamble

Usage of the benefits of electrical communications in general and digital communications in particular, is an inseparable part of our daily experience now. Innumerable applications due to developments in digital communications have already started influencing our day-to-day activities directly or indirectly. Popularity of the Internet and television are only two of the most obvious examples to prove the point. In fact, it may not be an overstatement today that 'information highways' are considered as essential ingredients of national infrastructure in the march of a modern society. It is, however, pertinent to mention that isolated developments only in the field of electrical communications have not caused this phenomenon. Remarkable progresses and technical achievements in several related fields in electronics engineering and computer engineering have actually made applications of several principles and theories of communication engineering feasible for implementation and usage. The purpose of this web course, however, is narrow and specific to the principles of digital communications.

This web course on 'Digital Communications' is primarily intended for use by undergraduate students who may be preparing for graduate level studies in the area of electrical communications engineering. A teacher, offering an introductory-level course on digital communications, may also find several topics suitable for classroom coverage. The field of Digital Communications is reach in literature and there is no dearth of excellent text books and research papers on specific topics over and above the bulk of tutorial material, technical standards and product information that are available through the Internet. Hence, the onus is clearly on the present authors to justify the need and relevance of this web course on 'Digital Communications'. To put it humbly, the present authors believe that any 'web course' should primarily cater to the quick requirements of the prime target audience (in our case, an undergraduate student preparing for graduate level studies in the area of electrical communications engineering). The usual requirements are believed to be of the following types: a) exposition to a relevant topic or concept, b) examples and problems to highlight the significance or use of certain principles and c) specific data or information in relation to a topic of study in the area of digital communications. Our teaching experience says that some or all of these requirements are indeed met in several textbooks to a good extent. For ready reference, a consolidated Bibliography is appended at the end of this course material. What stand out, probably, in favour of a 'web course' are the flexibility in using the material may be covered and the scope of continuous upgradation of the material to cater to specific needs of the audience in future.

The general structure of ’40-Lesson course’ is an indication to the implicit limits (of ‘time to read’ and ‘storage’); hence a balance among the reader requirements a) – c), mentioned above, should be worked out. The present version of this web course is designed with more emphasis on exposing relevant topics and concepts [requirement a)] which may supplement classroom teaching.

The course is split in seven Modules as outlined below.

The first module consists of four lessons. The present lesson (Lesson #1) gives an outline of major historical developments in the field of research in telecommunications engineering over a period of hundred years. Materials on radio spectrum should help recapitulate a few basic issues. The lesson ends with a general schematic description on a digital communication system. Lesson #2 gives a brief classification of signals and emphasizes the importance of sampling theory. Lesson #3 presents some basic concepts of information theory, which helps in appreciating other central principles and techniques of digital transmission. The concept of ‘information’ is also outlined here. Needs and benefits of modeling an information source are the topics in Lesson #4.

The second module is devoted to Random Processes. The module starts with a simple to follow introduction to random variables (Lesson #5). It is often necessary to acquire the skill of defining appropriate functions of one or more random variables and their manipulation to have greater insight into parameters of interest. The topic is introduced in Lesson #6 wherein only functions of one random variable have been considered. A powerful and appropriate modeling of a digital communication system is often possible by resorting to the rich theories of stochastic processes and this remains an important tool for deeper analysis of any transmission system in general. The topic has been treated at an elementary level in Lesson #7. A few commonly encountered random distributions, such as binomial, Poisson, Gaussian and Rayleigh are presented in Lesson #6. An emerging and powerful branch in electrical communication engineering is now popularly known as statistical signal processing and it encompasses several interesting issues of communication engineering including those of signal detection and parameter estimation. The basic backgrounds, laid in Lessons #5 to #8 should be useful in appreciating some of the generic issues of signal detection and parameter estimation as outlined in Lesson #9.

The third module on pulse coding focuses on the specific tasks of quantization and coding as are necessary for transmission and reception of an analog electrical signal. It is however, assumed that the reader is familiar with the basic schemes of analog-to-digital conversion. The emphasis in this module is more on the effects of quantization error (Lesson #10) while different pulse coding schemes such as Pulse Code Modulation (Lesson #11), Log-PCM (Lesson #12), Differential Pulse Code Modulation (Lesson #13) and Delta Modulation (Lesson #14) are used for possible reductions in the average number of bits that may have to be transmitted (or stored) for a given analog signal. The example of speech signal has been considered extensively.

Appropriate representation of bits (or information bearing symbol) is a key issue in any digital transmission system if the available bandwidth is not abundant. Most of the physical transmission media (e.g. twisted copper telephone line, good quality coaxial cable, radio frequency bands) are, in general, limited in terms of available frequency band (a simple reason for this general observation: demand for good quality digital communication system, in terms of bits to be transferred per second, has been rising with newer demands and aspirations from users). So, it makes sense to look for time-limited energy pulses to represent logical '1'-s and '0'-s such that the signal, after representation, can be transmitted reliably over the available limited bandwidth. The issue is pertinent for both carrier less (referred as 'baseband' in Module #4) transmission as well as modulated transmission (with carrier, Module #5). Several interesting and relevant issues such as orthogonality amongst time-limited energy pulses (Lesson #15), baseband channel modeling (Lesson #17) and signal reception strategies (Lessons #18 - #21) have, hence, been included in Module #4.

Module #5 is fully devoted to the broad topic of Carrier Modulation. Several simple digital modulation schemes including amplitude shift keying, frequency shift keying (Lesson #23) and phase shift keying (Lessons #24 - #26) have been introduced briefly. Performance of these modulation schemes in the background of additive Gaussian noise process is addressed in Lesson #27 and Lesson #28. If appreciated fully, these basic techniques of performance evaluation will also be useful in assessing performance of the digital modulation schemes in presence of other transmission impairments (e.g. interference). The basic issues of carrier synchronization and timing synchronization have been elaborated with reasonable illustrations in Lesson #31 and Lesson #32.

Module #6 is on error control coding or 'Channel Coding' as it is popularly known today. Basics of block and convolutional codes have been presented in three lessons (Lessons #33 - #35). Two more lessons on turbo coding (Lesson #37) and coded modulation schemes (Lesson #36) have been added in view of the importance of these schemes and procedures in recent years.

Spread spectrum communication techniques have gained popularity in last two decades in view of their widespread commercial use in digital satellite communications and cellular communications. A primary reason for this is the inherent feature of multiple access that helps simultaneous use of radio spectrum by multiple users. Effectively, several users can access the same frequency band to communicate information successfully without appreciable interference. Basic spread spectrum techniques have been discussed in Lesson #38 of Module #7 before highlighting the multiple access feature in Lesson #40. It is interesting to note that a spread spectrum communication system offers several other advantages such as anti-jamming and low probability of interception. In such non-conventional applications, the issue of code acquisition and fine tracking is of utmost importance as no pilot signal is usually expected to aid the process of code synchronization. To appraise the reader about this interesting and practical aspect of code synchronization the topic has been introduced in Lesson #39.

A short Bibliography is appended at the end of Lesson #40.

Block Schematic Description of a Digital Communication System

In the simplest form, a transmission-reception system is a three-block system, consisting of a) a transmitter, b) a transmission medium and c) a receiver. If we think of a combination of the transmission device and reception device in the form of a 'transceiver' and if (as is usually the case) the transmission medium allows signal both ways, we are in a position to think of a both-way (bi-directional) communication system. For ease of description, we will discuss about a one-way transmission-reception system with the implicit assumption that, once understood, the ideas can be utilized for developing / analyzing two-way communication systems. So, our representative communication system, in a simple form, again consists of three different entities, viz. a transmitter, a communication channel and a receiver.

A digital communication system has several distinguishing features when compared with an analog communication system. Both analog (such as voice signal) and digital signals (such as data generated by computers) can be communicated over a digital transmission system. When the signal is analog in nature, an equivalent discrete-time-discrete-amplitude representation is possible after the initial processing of sampling and quantization. So, both a digital signal and a quantized analog signal are of similar type, i.e. discrete-time-discrete-amplitude signals.

A key feature of a digital communication system is that a sense of 'information', with appropriate unit of measure, is associated with such signals. This visualization, credited to Claude E. Shannon, leads to several interesting schematic description of a digital communication system. For example, consider **Fig.1.1.1** which shows the signal source at the transmission end as an equivalent 'Information Source' and the receiving user as an 'Information sink'. The overall purpose of the digital communication system is 'to collect information from the source and carry out necessary electronic signal processing such that the information can be delivered to the end user (information sink) with acceptable quality'. One may take note of the compromising phrase 'acceptable quality' and wonder why a digital transmission system should not deliver exactly the same information to the sink as accepted from the source. A broad and general answer to such query at this point is: well, it depends on the designer's understanding of the 'channel' (**Fig. 1.1.1**) and how the designer can translate his knowledge to design the electronic signal processing algorithms / techniques in the 'Encoder' and 'decoder' blocks in **Fig. 1.1.1**. We hope to pick up a few basic yet good approaches to acquire the above skills. However, pioneering work in the 1940-s and 1950-s have established a bottom-line to the search for 'a flawless (equivalently, 'error-less') digital communication system' bringing out several profound theorems (which now go in the name of Information Theory) to establish that, while error-less transmission of information can never be guaranteed, any other 'acceptable quality', arbitrarily close to error-less transmission may be possible. This 'possibility' of almost error-less information transmission has driven significant research over the last five decades in multiple related areas such as, a) digital modulation schemes, b) error control techniques, c) optimum receiver design, d) modeling and characterization of channel and so forth. As

a result, varieties of digital communication systems have been designed and put to use over the years and the overall performance have improved significantly.

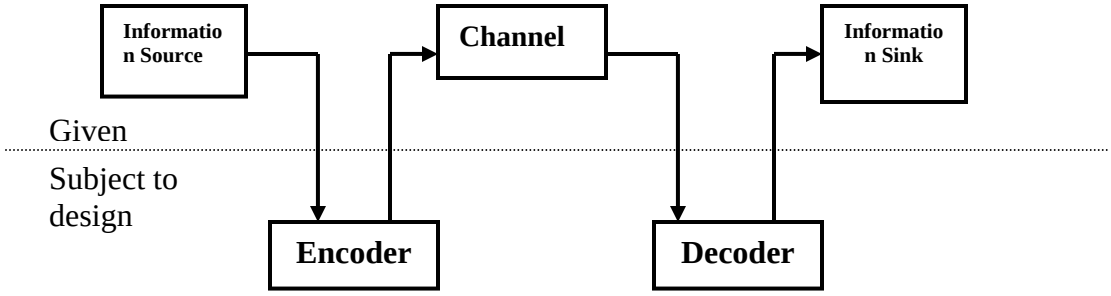


Fig. 1.1.1 *Basic block diagram of a digital communication System*

It is possible to expand our basic 'three-entity' description of a digital communication system in multiple ways. For example, **Fig. 1.1.2** shows a somewhat elaborate block diagram explicitly showing the important processes of 'modulation-demodulation', 'source coding-decoding' and 'channel encoding – decoding'. A reader may have multiple queries relating to this kind of abstraction. For example, when 'information' has to be sent over a large distance, it is a common knowledge that the signal should be amplified in terms of power and then launched into the physical transmission medium. Diagrams of the type in **Figs. 1.1.1** and **1.1.2** have no explicit reference to such issues. However, the issue here is more of suitable representation of a system for clarity rather than a module-by-module replication of an operational digital communication system.

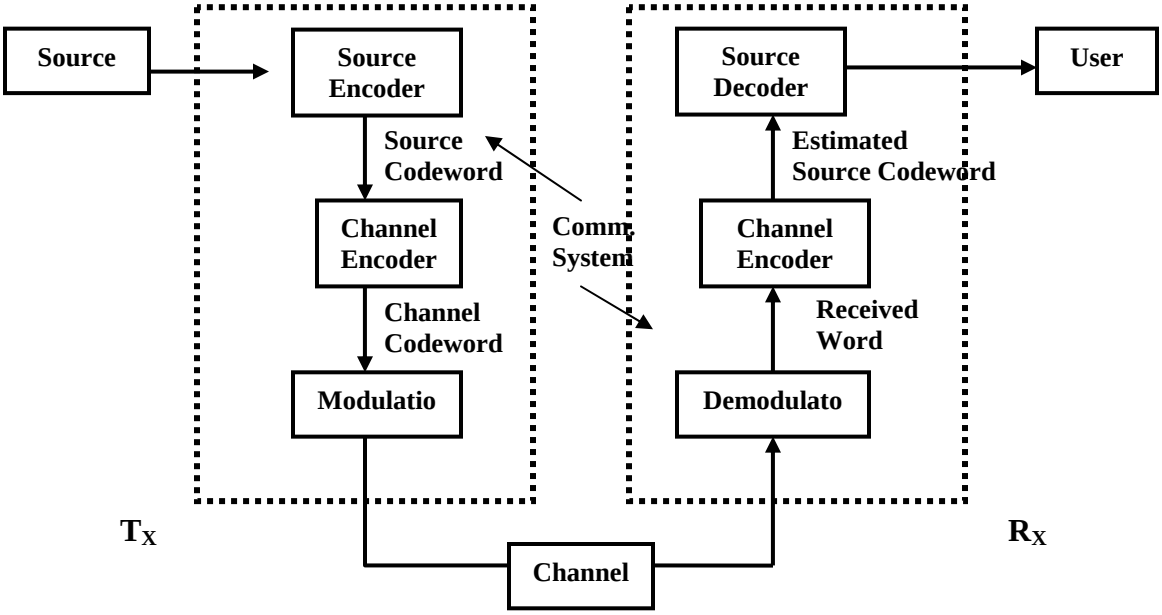


Fig. 1.1.2 *A possible break up of the previous diagram (following Shannon's ideas)*

To elaborate this potentially useful style of representation, let us note that we have hardly discussed about the third entity of our model, viz. the 'channel'. One can define several types of channel. For example, the 'channel' in **Fig. 1.1.2** should more appropriately be called as a 'modulation channel' with an understanding that the actual transmission medium (called 'physical channel'), any electromagnetic (or other wise) transmission- reception operations, amplifiers at the transmission and reception ends and any other necessary signal processing units are combined together to form this 'modulation channel'.

We will see later that a modulation channel usually accepts modulated signals as analog waveforms at its inputs and delivers another version of the modulated signal in the form of analog waveforms. Such channels are also referred as 'waveform channels'. The 'channel' in **Fig. 1.1.1**, on the other hand, appears to accept some 'encoded' information from the source and deliver some 'decoded' information to the sink. Both the figures are potentially useful for describing the same digital communication system. On comparison of the two figures, the reader is encouraged to infer that the 'channel' in **Fig. 1.1.1** includes the 'modulation channel' and the modulation- demodulation operations of **Fig. 1.1.2**. The 'channel' of **Fig. 1.1.1** is widely denoted as a 'discrete channel', implying that it accepts discrete-time-discrete-amplitude signals and also delivers discrete-time-discrete-amplitude signals.

In the following, we introduce a few short tables, which may help a reader to recapitulate some relevant issues of electrical communications. **Table 1.1.1** lists some of the important events which have contributed to the developments in electrical communication. **Table 1.1.2** presents different frequency bands with typical applications that are commonly used for the purpose of electrical communications. This table is very useful for our subsequent lessons. **Table 1.1.3** mentions frequency ranges for a few popular broadcast and communication services. **Table 1.1.4** gives an idea of typical centre frequencies and the nominal bandwidths that are available for five frequency bands. It is important to note that larger bandwidths are available when the operating frequency bands are higher. **Table 1.1.5** provides an idea of typical power losses of several physical transmission media at representative operating frequency. It may be noted that all transmission media are not equally suitable at all frequencies. An important factor other than the power loss in a physical medium is its cost per unit length.

Year / Period	Achievements
1838	Samuel F. B. Morse demonstrated the technique of telegraph
1876	Alexander Graham Bell invents telephone
1897	Guglielmo Marconi patents wireless telegraph system. A few years earlier, Sir J. C. Bose demonstrated the working principle of electromagnetic radiation using a 'solid state coherer'
1918	B. H. Armstrong develops super heterodyne radio receiver
1931	Teletype service introduced
1933	Analog frequency modulation invented by Edwin Armstrong
1937	Alec Reeves suggests pulse code modulation (PCM)
1948-49	Claude E. Shannon publishes seminal papers on 'A Mathematical Theory of Communications'
1956	First transoceanic telephone cable launched successfully
1960	Development of Laser
1962	Telstar I, first satellite for active communication, launched successfully
1970-80	Fast developments in microprocessors and other digital integrated circuits made high bit rate digital processing and transmission possible; commercial geostationary satellites started carrying digital speech, wide area computer communication networks started appearing, optical fibers were deployed for carrying information through light., deep space probing yielded high quality pictures of planets.
1980-90	Local area networks (LAN) making speedy inter-computer data transmission became widely available; Cellular telephone systems came into use. Many new applications of wireless technology opened up remarkable scopes in business automation.
1990-2000	Several new concepts and standards in data network, such as, wireless LAN (WLAN), AdHoc networks, personal area networks (PAN), sensor networks are under consideration for a myriad of potential applications.

Table 1.1.1 *Some milestones in the history of electrical communications*

Frequency Band	Wavelength	Name	Transmission Media	Some Applications
3 – 30 KHz	100–10 Km	Very Low Frequency (VLF)	Air, water, copper cable	Navigation, SONAR
30–300 KHz	10 Km- 1 Km	Low Frequency (LF)	Air, water, copper cable	Radio beacons, Ground wave communication
300KHz – 3 MHz	1 Km – 100 m	Medium Frequency (MF)	Air, copper cable	AM radio, navigation, Ground wave communication
3 MHz – 30 MHz	100 m– 10 m	High Frequency (HF)	Air, copper and coaxial cables	HF communication, Citizen's Band (CB) radio, ionosphere communication
30MHz- 300 MHz	10 m – 1 m	Very High Frequency (VHF)	Air, free space, coaxial cable	Television, Commercial FM broadcasting, point to point terrestrial communication
300 MHz – 3 GHz	1m – 10 cm	Ultra High Frequency (UHF)	Air, free space, waveguide	Television, mobile telephones, satellite communications,
3GHz – 30 GHz	10cm–1cm	Super / Extra High Frequency (SHF / EHF)	Air, free space, waveguide	Satellite communications, wireless LAN, Metropolitan Area network (WMAN), Ultra Wideband communication over a short distance
30 GHz – 300 GHz	1 cm – 1 mm			Mostly at experimental stage
30 Tera Hz – 3000 Tera Hz	10 μm – 0.1μm (approx)	Optical	Optical fiber	Fiber optic communications

Table 1.1.2 *Electromagnetic bands with typical applications*

Any radio operation at 1GHz or beyond (upto several tens of GHz) is also termed as 'microwave' operation.

Name / Description	Frequency Range	Application
AM Broadcast Radio	540 KHz – 1600 KHz	Commercial audio broadcasting using amplitude modulation
FM Broadcast Radio	88 MHz – 108 MHz	Commercial audio broadcasting using frequency modulation
Cellular Telephony	806 MHz – 940 MHz	Mobile telephone communication systems
Cellular Telephony and Personal Communication Systems (PCS)	1.8 GHz – 2.0 GHz	Mobile telephone communication systems
ISM (Industrial Scientific and Medical) Band	2.4 GHz – 2.4835 GHz	Unlicensed band for use at low transmission power
WLAN (Wireless Local Area Network)	2.4 GHz band and 5.5 GHz	Two unlicensed bands are used for establishing high speed data network among willing computers
UWB (Ultra Wide Band)	3.7 GHz – 10.5 GHz	Emerging new standard for short distance wireless communication at a very high bit rate (typically, 100 Mbps)

Table 1.1.3 *A few popular frequency bands*

Frequency band	Carrier frequency	Approx. Bandwidth
Long wave Radio [LF]	100KHz	~ 2 KHz
Short wave [HF]	5MHz	100 KHz
VHF	100MHz	5 MHz
Micro wave	5GHz	100 MHz
Optical	5×10^{14} Hz	10 GHz – 10 THz

Table 1.1.4 *Some Carrier frequency values and nominal bandwidth that may be available at the carrier frequency*

Transmission medium	Frequency	Power loss in [dB/km]
Twisted copper wire [16 AWG]	1 KHz 100KHz	0.05 3.0
Co-Axial Cable [1cm dia.]	100 KHz 3 MHz	1.0 4.0
Wave Guide	10 GHz	1.5
Optical Fiber	$10^{14} – 10^{16}$ Hz	<0.5

Table 1.1.5 Typical power losses during transmission through a few media

Problems

Q1.1.1) Mention two reasons justifying the source encoding operation in a digital communication system.

Q1.1.2) Give examples of three channels, which are used for purpose of communication

Q1.1.3) Give three examples of types of signals that a source (Fig 1.1.2) may generate.

Q1.1.4) Signaling in UHF band allows higher bit rate compared to HF band – criticize this comment.

Module

1

Introduction to Digital Communications and Information Theory

Lesson 2

Signals and Sampling Theory

After reading this lesson, you will learn about

- ***Need for careful representation of signals***
- ***Types of signals***
- ***Nyquist's sampling theorem and its practical implications***
- ***Band pass representation of narrow band signals***

'Signal' in the present context means electrical manifestation of a physical process. Usually an essence of 'information' will be associated with a signal. Mathematical representation or abstraction should also be possible for a signal such that a signal and its features can be classified and analyzed.

Examples of a few signals

(a) Electrical equivalent of speech/voice as obtained at the output of a microphone.
(b) Electrical output of a transducer used to sense the temperature of a furnace.
(c) Stream of electrical pulses (digital) generated by a computer.
(d) Electrical output of a TV camera (video signal).
(e) Electrical waves received by the antenna of a radio/TV/communication receiver.
(f) ECG signal.

When a 'signal' is viewed as electrical manifestation of a process, the signal is a function of one or more independent variables. For all the examples cited above, the respective signals may commonly be considered as function of 'time'. So, a notation like the following may be used to represent a signal:

s(a, b, c, t,..), where 'a', 'b',… are the independent variables.

However, observe that a mere notation of a signal, say s(t), does not reveal all its features and behavior and hence it may not be possible to analyze the signal effectively. Further, processing and analyses of many signals may become easy if we can associate them, in some cases even approximately, with mathematical functions that may be analyzed by well-developed mathematical tools and techniques. The approximate representations, wherever adopted, are usually justified by their ease of analysis or tractability or some other evidently rewarding reason. For example, the familiar mathematical function $s(t) = A\cos(\omega t + \theta)$ is extensively used in the study, analysis and testing of several principles of communication theory such as carrier modulation, signal sampling etc. However, one can very well contest the fact that $s(t) = A\cos(\omega t + \theta)$ hardly implies a physical process because of the following reasons:

(i) no range of 't' is specified and hence, mathematically the range may be from $-\infty$ to $+\infty$. This implies that the innocent looking function s(t) should exist over the infinite range of 't', which is not true for any physical source if 't' represents time. So, some range for 't' should be specified.

(ii) s(t), over a specified range of 't', is a known signal in the sense that, over the range of 't', if we know the value of s(t) at say $t = t_0$, and the values of A, ω and θ we

know the value of s(t) at any other time instant 't'. We say the signal s(t) is deterministic. In a sense, such a mathematical function does not carry information.

While point (i) implies the need for rigorous and precise expression for a signal, point (ii) underlines the usage of theories of mathematics for signals deterministic or non-deterministic (random).

To illustrate this second point further, let us consider the description of $s(t) = A \cos \omega t$, where 't' indicates time and $\omega = 2\pi f$ implies angular frequency:

(a) Note that $s(t) = A \cos \omega t$, $-\infty < t < \infty$ is a periodic function and hence can be expressed by its exponential (complex) Fourier series. However, this signal has infinite energy E, $E = \int_{-\infty}^{+\infty} s^2(t)dt$ and hence, theoretically, can not be expressed by Fourier Transformation.

(b) Let us now consider the following modified expression for s(t) which may be a closer representation of a physical signal:

$s(t) = A \cos \omega t$, $0 \leq t < \infty$

$= A.u(t). \cos \omega t$ where u(t) is the unit step function, $u(t) = 0$, $t < 0$ and $u(t) = 1$, $t \geq 0$

If we further put an upper limit to 't', say, $s(t) = A \cos \omega t$, $t_1 \leq t \leq t_2$, such a signal can be easily generated by a physical source, but the frequency spectrum of s(t) will now be different compared to the earlier forms. For simplicity in notation, depiction and understanding, we will, at times, follow mathematical models for describing and understanding physical signals and processes. We will, though, remember that such mathematical descriptions, while being elegant, may show up some deviation from the actual behavior of a physical process. Henceforth, we will mean the mathematical description itself as the signal, unless explicitly stated otherwise.

Now, we briefly introduce the major classes of signals that are of frequent interest in the study of digital communications. There are several ways of classifying a signal and a few types are named below.

Energy signal: If, for a signal s(t), $\int_{0}^{+\infty} s^2(t)dt < \infty$ i.e. the energy of the signal is finite, the signal is called an energy signal. However, the same signal may have large power. The voltage generated by lightning (which is of short duration) is a close example of physical equivalent of a signal with finite energy but very large power.

Power signal: A power signal, on the contrary, will have a finite power but may have finite or infinite energy. Mathematically,

$$\lim_{T \to \infty} \frac{1}{T} \int_{-T/2}^{+T/2} s^2(t)\,dt < \infty$$

Note: While electrical signals, derived from physical processes are mostly energy signals, several mathematical functions, usually deterministic, represent power signals.

Deterministic and random signals: If a signal s(t), described at $t = t_1$ is sufficient for determining the signal at $t = t_2$ at which the signal also exists, then s(t) represents a deterministic signal.

Example: $s(t) = A \cos \omega t$, $T_1 \leq t \leq T_2$

There are many signals that can best be described in terms of a probability and one may not determine the signal exactly.

Example: (from real process) Noise voltage/current generated by a resistor.

Such signals are labeled as non-deterministic or random signals.

Continuous time signal: Assuming the independent variable 't' to represent time, if s(t) is defined for all possible values of t between its interval of definition (or existence), $T_1 \leq t \leq T_2$. Then the signal s(t) is a continuous time signal.

If a signal s(t) is defined only for certain values of t over an interval $T_1 \leq t \leq T_2$, it is a discrete-time signal. A set of sample values represent a discrete time signal.

Periodic signal: If $s(t) = s(t + T)$, for entire range of t over which the signal s(t) is defined and T is a constant, s(t) is said to be periodic or repetitive. 'T' indicates the period of the signal and $1/T$ is its frequency of repetition.

Example: $s(t) = A \cos \omega t$, $-\infty \leq t \leq \infty$, where $T = 2\pi/\omega$.

Analog: If the magnitudes of a real signal s(t) over its range of definition, $T_1 \leq t \leq T_2$, are real numbers (there are infinite such values) within a finite range, say, $S_{min} \leq S(t) \leq S_{max}$, the signal is analog.

A digital signal s(t), on the contrary, can assume only any of a finite number of values. Usually, a digital signal implies a discrete-time, discrete-amplitude signal.

The mathematical theories of signals have different flavours depending on the character of a signal. This helps in easier understanding and smarter analyses. There may be considerable similarities among the mathematical techniques and procedures. We assume that the reader has some familiarity with the basic techniques of Fourier series expansion and Fourier Transform. In the following, we present a brief treatise on sampling theory and its implications in digital communications.

Sampling Theorem

The concepts and techniques of sampling a continuous-time signal have important roles in baseband signal processing like digitization, multiplexing, filtering and also in carrier modulations in wireless digital communication systems. A common use of sampling theorem is for converting a continuous-time signal to an equivalent discrete-time signal and vice versa.

Generally, a modulated signal in a wireless system is to be transmitted within an allocated frequency band. To accomplish this, the modulating message signal is filtered before it is modulated and transmitted. When the message signal is already available in digital form, an appropriate pulse shaping operation may be performed on the digital stream before it modulates a carrier. We will see later in this section how the basic concepts of sampling may be adapted to shape the digital pulses. In case the message is available as a continuous-time signal, it is first band-limited and then sampled to generate approximately equivalent set of pulses. A set of pulses may be called equivalent to the original analog signal if a technique exists for reconstructing the filtered signal uniquely from these pulses. Nyquist's famous theorems form the basis of such preprocessing of continuous-time signals.

Nyquist's Uniform Sampling Theorem for Lowpass Signals

Part - I If a signal x(t) does not contain any frequency component beyond W Hz, then the signal is completely described by its instantaneous uniform samples with sampling interval (or period) of $T_s < 1/(2W)$ sec.

Part – II The signal x(t) can be accurately reconstructed (recovered) from the set of uniform instantaneous samples by passing the samples sequentially through an ideal (brick-wall) lowpass filter with bandwidth B, where $W \le B < f_s - W$ and $f_s = 1/(T_s)$.

As the samples are generated at equal (same) interval (T_s) of time, the process of sampling is called uniform sampling. Uniform sampling, as compared to any non-uniform sampling, is more extensively used in time-invariant systems as the theory of uniform sampling (either instantaneous or otherwise) is well developed and the techniques are easier to implement in practical systems.

The concept of 'instantaneous' sampling is more of a mathematical abstraction as no practical sampling device can actually generate truly instantaneous samples (a sampling pulse should have non-zero energy). However, this is not a deterrent in using the theory of instantaneous sampling, as a fairly close approximation of instantaneous sampling is sufficient for most practical systems. To contain our discussion on Nyquist's theorems, we will introduce some mathematical expressions.

If x(t) represents a continuous-time signal, the equivalent set of instantaneous uniform samples $\{x(nT_s)\}$ may be represented as,

$$\{x(nT_s)\} \equiv x_s(t) = \Sigma\, x(t).\delta(t - nT_s) \qquad 1.2.1$$

where $x(nT_s) = x(t)\big|_{t=nTs}$, $\delta(t)$ is a unit pulse singularity function and 'n' is an integer

Conceptually, one may think that the continuous-time signal x(t) is multiplied by an (ideal) impulse train to obtain $\{x(nT_s)\}$ as (1.2.1) can be rewritten as,

$$x_s(t) = x(t).\Sigma\, \delta(t - nT_s) \qquad 1.2.2$$

Now, let X(f) denote the Fourier Transform F(T) of x(t), i.e.

$$X(f) = \int_{-\infty}^{+\infty} x(t).\exp(-j2\pi ft)dt \qquad 1.2.3$$

Now, from the theory of Fourier Transform, we know that the F.T of $\Sigma\ \delta(t- nT_s)$, the impulse train in time domain, is an impulse train in frequency domain:

$$F\{\Sigma\ \delta(t- nT_s)\} = (1/T_s).\Sigma\ \delta(f- n/T_s) = f_s.\Sigma\ \delta(f- nf_s) \qquad 1.2.4$$

If $X_s(f)$ denotes the Fourier transform of the energy signal $x_s(t)$, we can write using Eq. (1.2.4) and the convolution property:

$$\begin{aligned} X_s(f) &= X(f)^* F\{\Sigma\ \delta(t- nT_s)\} \\ &= X(f)^*[f_s.\Sigma\ \delta(f- nf_s)] \\ &= f_s.X(f)^*\Sigma\ \delta(f- nf_s) \\ &= f_s.\int_{-\infty}^{+\infty} X(\lambda).\Sigma\ \delta(f- nf_s-\lambda)d\lambda = f_s.\ \Sigma \int X(\lambda).\delta(f- nf_s-\lambda)d\lambda = f_s.\Sigma\ X(f- nf_s) \end{aligned} \qquad 1.2.5$$

[By sifting property of $\delta(t)$ and considering $\delta(f)$ as an even function, i.e. $\delta(f) = \delta(-f)$]

This equation, when interpreted appropriately, gives an intuitive proof to Nyquist's theorems as stated above and also helps to appreciate their practical implications. Let us note that while writing Eq.(1.2.5), we assumed that x(t) is an energy signal so that its Fourier transform exists. Further, the impulse train in time domain may be viewed as a periodic singularity function with almost zero (but finite) energy such that its Fourier Transform [i.e. a train of impulses in frequency domain] exists. With this setting, if we assume that x(t) has no appreciable frequency component greater than W Hz and if $f_s > 2W$, then Eq.(1.2.5) implies that $X_s(f)$, the Fourier Transform of the sampled signal $x_s(t)$ consists of infinite number of replicas of X(f), centered at discrete frequencies $n.f_s$, $-\infty < n < \infty$ and scaled by a constant $f_s = 1/T_s$ (**Fig. 1.2.1**).

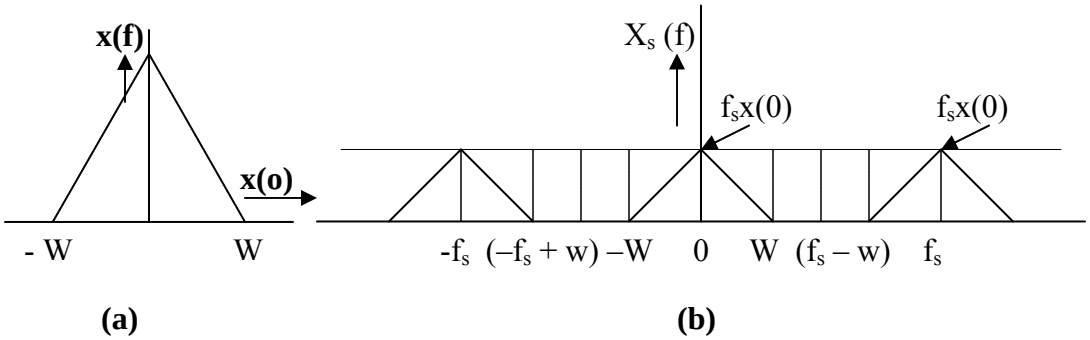


Fig. 1.2.1 *Spectra of (a) an analog signal x(t) and (b) its sampled version*

Fig. 1.2.1 indicates that the bandwidth of this instantaneously sampled wave $x_s(t)$ is infinite while the spectrum of x(t) appears in a periodic manner, centered at discrete frequency values $n.f_s$.

Now, Part – I of the sampling theorem is about the condition $f_s > 2.W$ i.e. $(f_s - W) > W$ and $(-f_s + W) < -W$. As seen from Fig. 1.2.1, when this condition is satisfied, the spectra of $x_s(t)$, centered at $f = 0$ and $f = \pm f_s$ do not overlap and hence, the spectrum of x(t) is present in $x_s(t)$ without any distortion. This implies that $x_s(t)$, the appropriately sampled version of x(t), contains all information about x(t) and thus represents x(t).

The second part of Nyquist's theorem suggests a method of recovering x(t) from its sampled version $x_s(t)$ by using an ideal lowpass filter. As indicated by dotted lines in Fig. 1.2.1, an ideal lowpass filter (with brick-wall type response) with a bandwidth $W \leq B < (f_s - W)$, when fed with $x_s(t)$, will allow the portion of $X_s(f)$, centered at $f = 0$ and will reject all its replicas at $f = n\, f_s$, for $n \neq 0$. This implies that the shape of the continuous-time signal $x_s(t)$, will be retained at the output of the ideal filter. The reader may, supposedly, have several queries at this point such as:

(a) Can a practical LPF with finite slope be used when $W \leq B < (f_s - W)$?
(b) Can we not recover x(t) if there is overlapping ?
(c) What happens to the above description when non-ideal but uniform sampling (like flat-top or natural) is used instead of instantaneous sampling?

One may observe that the above questions are related towards use of Nyquist's sampling theorems in practical systems. Instead of addressing these questions directly at this point, we wish to cast the setting a bit more realistic by incorporating the following issues:

(1) a practical uniform sampling scheme in place of instantaneous sampling (flat top sampling) and
(2) frequency response of conveniently realizable analog lowpass filters.

Flat Top Sampling

A train of pulses with narrow width (τ), rather than an impulse train, is practically realizable (**Fig.1.2.2**). The pulse width τ, though small, is considered to be significant compared to the rise and fall times of the pulses to avoid unnecessary mathematical complexity.

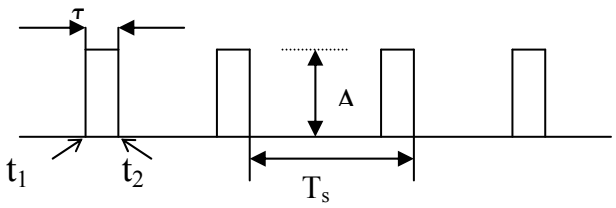


Fig. 1.2.2 *A train of narrow pulses with pulse width 'τ'*

In a flat top sampling scheme, the amplitude of a pulse after sampling is kept constant and is related to the value of signal x(t) at a pre-decided instant within the pulse duration τ. (e.g., we may arbitrarily decide that the pulse amplitude after sampling will be proportional to x(t) at the beginning instant of the pulse). The point to be noted is that, though τ is non zero, the top of a sampled pulse does not follow x(t) during τ and instead is held constant, thus retaining the flavour of instantaneous sampling to an extent. However, the spectrum of such flat-lop sampled signal for the same signal x(t), is different from the earlier spectrum.

The periodic train of sampling pulses with width τ, amplitude 'A' and sampling interval 'T', may be expressed as,

$$P(t) = \sum_{n=-\infty}^{\infty} A\pi\left(t - nT_s\tau/2\right) \qquad 1.2.6$$

Here, $\pi(t)$ indicates a unit pulse of width τ, $-\tau/2 \leq t < \tau/2$.

On Fourier Series expansion of p(t), it can be shown that the envelop of p(t) follows a $|\text{sinc } x|$ function instead of a constant amplitude train of frequency pulses as earlier. Hence, the spectrum of the flat-top sampled signal $x_s(t)$ will not show up the spectrum of x(t) and its exact replicas at $f = \pm nf_s$. Instead, the spectrum of x(t) will be contoured by the $\text{sinc}(n\tau f_s)$ function. **Fig. 1.2.3** sketches this feature only for the main lobe of the sinc function.

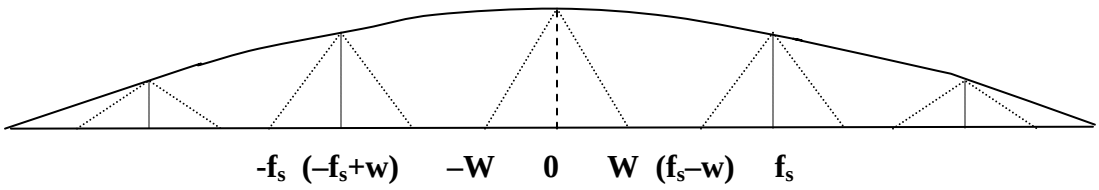


Fig. 1.2.3 *A sketch indicating the change in $X_s(f)$ due to flat-top sampling w.r.t* **Fig. 1.2.1 (b)**

So to be exact, an ideal lowpass filter, with flat amplitude response (versus frequency) will recover the signal x(t) with some distortion. The higher frequency components of x(t) will get more attenuated compared to the lower frequency components of x(t). Sometimes, 'τ' is chosen sufficiently small such that the distortion in the recovered signal x(t) is within a tolerable limit. However, one may note from **Fig. 1.2.3** that perfect recovery of x(t) may still be possible if a different and special kind of lowpass filter is used (instead of an ideal LPF) which can compensate the 'sin x/x' amplitude distortion (and an associated phase distortion which has not been highlighted for the sake of brevity).

Frequency response of conveniently realizable analog lowpass filter

A system designer more often than not searches for a convenient filter realization, which can strike a compromise amongst several conflicting properties. To be specific, let us reiterate that Nyquist's first sampling theorem gives a lower bound of the sampling rate (f_s= 2W) and from **Fig.1.2.2** we can infer that the replicas of the spectrum of x(t) can be separated more from one another by increasing the sampling rate. When the replicas are widely separated, a practical LPF of low/moderate order and with constant passband covering at least up to W H_Z (and also maintaining linear phase relationship up to W H_Z) is good enough to select the spectrum of x(t) located around f = 0. While higher and higher sampling rate eases the restriction on the design of lowpass reconstruction filter, other issues such as bit rate, multiplexing several signals, cost and complexity of the high speed sampling circuit usually come in the way. On the whole, the sampling rate f_s is chosen marginally higher than 2W samples/sec striking a balance among the contending issues.

Sampling of narrow bandpass signals: - an inefficient approach

A form of a narrow bandpass signal that is often encountered in the design and analysis of a wireless communication system is:

$$x(t) = A(t)\cos\{\omega_c t + \theta(t)\} \qquad 1.2.7$$

The spectrum of such a bandpass signal is centered at frequency f_c (= $\omega_c/2\pi$) and the bandwidth is usually small (less than 10%) compared to the centre frequency. Is it possible to represent such a continuous time signal by discrete-time samples? If possible, how is the sampling rate related to f_c and the bandwidth of the signal? How to reconstruct the original signal x(t) from its equivalent set of samples? We will try to find reasonable answers to these questions in the following introductory discussion.

Let the bandpass signal x(t), centered around 'f_c' have a band width 2B i.e. let x(t) be band limited from (f_c-B) to (f_c+B). By taking clue from Nyquist's uniform sampling theorem for lowpass signals, one may visualize the bandpass signal x(t) as a real lowpass signal, whose maximum frequency content is (f_c+B) and the lowest allowable frequency is 0 H_Z though actually there is no frequency component between 0 H_Z and (f_c-B) H_Z. Then one may observe that if x(t) is sampled at a rate greater than 2x(f_c+B), a set of valid samples is obtained which completely represents x(t). While this general approach is flawless, there exists an efficient and smarter way of looking at the problem. We will come back to this issue after having a brief look at complex low pass equivalent description of a real bandpass signal with narrow bandwidth.

Base band representation of narrow bandpass signal

Let, for a narrow band signal,

f_c: Center Frequency $= \frac{f_2 - f_1}{2} + f_1$

f_2: maximum frequency component ; f_1: minimum frequency component

So, the band width of the signal is: $BW = f_2 - f_1$

Now, we describe a real signal as a narrowband signal if $BW << f_c$. A rule of thumb, used to decide quickly whether a signal is a narrowband, is: $0.01 < (BW/f_c) < 0.1$. That is, the bandwidth of a narrow bandpass signal is considerably less than 10% of the centre frequency [refer **Table 1.1.4]**

Representation of narrow band signals

A general form of expressing a bandpass signal is: $x(t) = A(t)\cos[2\pi f_c t + \Phi(t)]$

Now, x(t) may be rewritten as:

$$x(t) = \{A(t)\cos\Phi(t)\}.\cos 2\pi f_c t - \{A(t)\sin\Phi(t)\}.\sin 2\pi f_c t$$
$$= u_I(t).\cos 2\pi f_c t - u_Q(t).\sin 2\pi f_c t$$
$$= \mathrm{Re}\{\tilde{u}(t).\exp(j2\pi f_c t)\}$$

Where, $\tilde{u}(t) = u_I(t) + ju_Q(t)$

$\tilde{u}(t)$: Complex Low pass Equivalent of x(t)

Note: Real band pass x(t) is completely described by complex low pass equivalent $\tilde{u}(t)$ and the centre frequency 'f_c'.

Spectra of x(t) and $\tilde{u}(t)$:

Let X(f) denote the Fourier Transform of x(t). Then,

$$X(f) = \int_{-\infty}^{\infty} x(t)\exp(-j2\pi ft)\,dt$$
$$= \int_{-\infty}^{\infty} R_e\left\{\tilde{u}(t)\exp(j2\pi f_c t)\right\}.\exp(-j2\pi ft)\,dt$$

Now, use the subtle observation that, for any complex number Z, $R_e\{\tilde{Z}\} = \frac{1}{2}\{\tilde{Z} + \tilde{Z}^*\}$

$$\therefore X(f) = \frac{1}{2}\int_{-\infty}^{\infty}\left[\tilde{u}(t)\exp(j\omega_c t) + \tilde{u}^*(t)\exp(-j\omega_c t)\right]$$
$$= \frac{1}{2}\left[U(f - f_c) + U^*(-f - f_c)\right],$$

where

$$U(f) = \int_{-\infty}^{\infty} \tilde{u}(t)\exp(-j\omega t)\,dt$$

Fig. 1.2.4 shows the power spectra of a narrowband signal and its lowpass equivalent

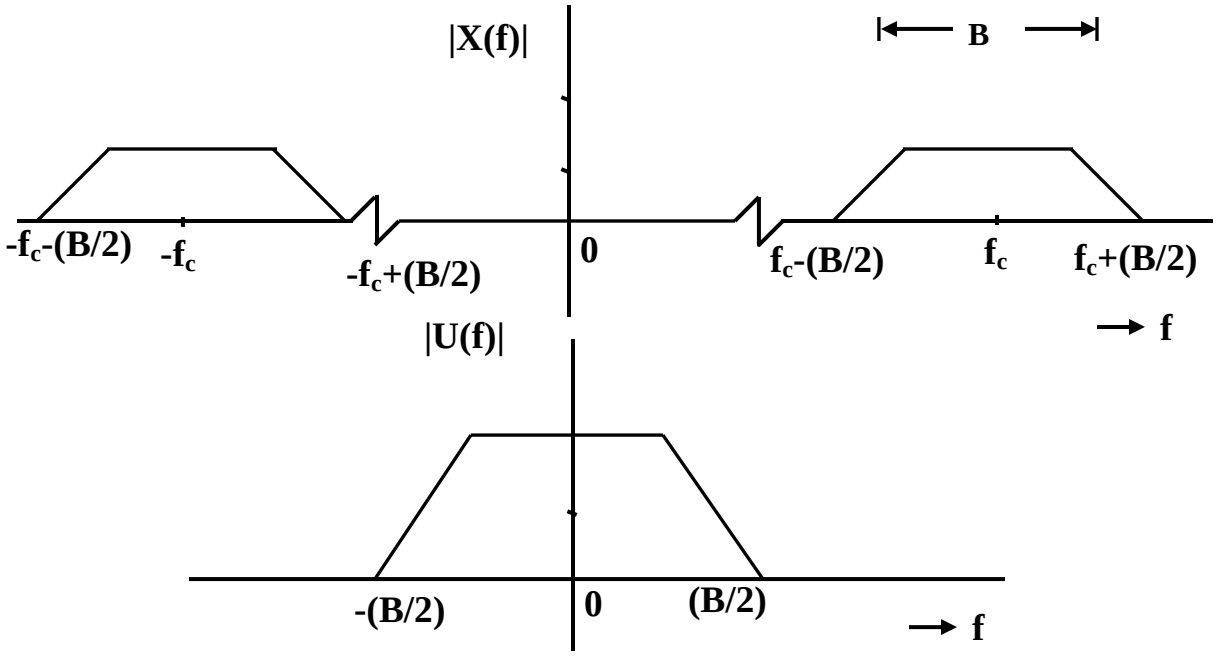


Fig. 1.2.4 *Power spectra of a narrowband signal and its lowpass equivalent*

Now, energy of $x(t) = \int_{-\infty}^{\infty} x^2(t)dt$

$$= \int_{-\infty}^{\infty} \left\{ R_e \left[\tilde{u}(t) \exp\left(j\omega_c t \right) \right] \right\}^2 dt$$

$$= \frac{1}{2} \int_{-\infty}^{\infty} \left| \tilde{u}(t) \right|^2 dt + \frac{1}{2} \int_{-\infty}^{\infty} \left| \tilde{u}(t) \right|^2 .\cos\left[4\pi f_c t + 2\varphi(t) \right] dt$$

Note: $\tilde{u}(t)$ as well as $\left| \tilde{u}(t) \right|^2$ vary slowly compared to the second component whose frequency is around $2f_c$.

This implies, energy of $x(t) \simeq \frac{1}{2} \int_{-\infty}^{\infty} \left| \tilde{u}(t) \right|^2 dt$

The sketch in **Fig. 1.2.5** helps to establish that the energy associated with a narrow band pass signal can be closely assessed by the energy of its equivalent complex lowpass representation.

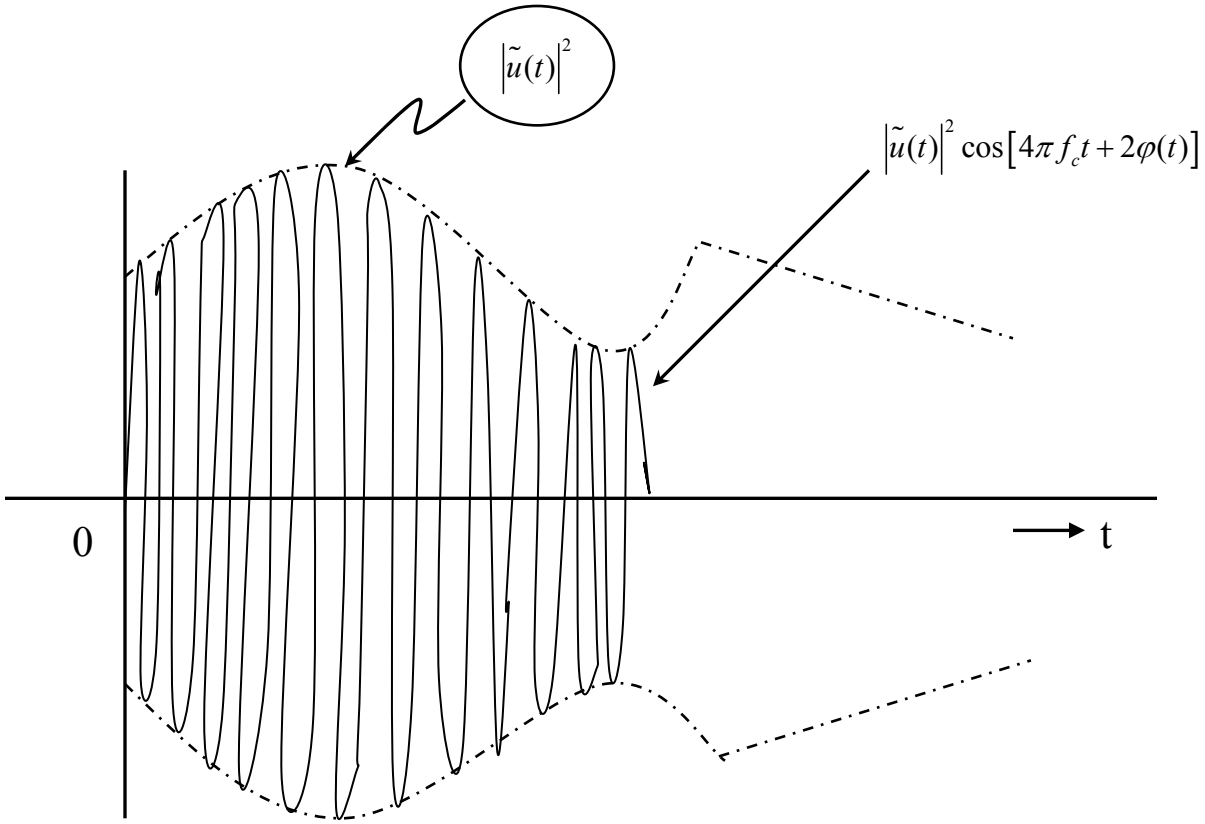


Fig. 1.2.5 *Sketch to establish that the energy associated with a narrow band pass signal can be closely assessed by the energy of its equivalent complex lowpass representation*

Sampling of narrow bandpass signals: - a better approach

Let $u_I(t)$ and $u_Q(t)$ represent x(t) such that each of them is band limited between 0 and B H_Z. Now, if $u_I(t)$ and $u_Q(t)$ are given instead of the actual signal x(t), we could say that the sampling rate for each of $u_I(t)$ and $u_Q(t)$ would be only 2B samples/sec [much less than $2(f_c+B)$ as $f_c>>B$]. Hence, the equivalent sampling rate may be as low as 2 x 2B or 4B samples/sec. **Fig.1.2.6** shows a scheme for obtaining real lowpass equivalent $u_I(t)$ and $u_Q(t)$ from narrow bandpass signal x(t) and their sampling.

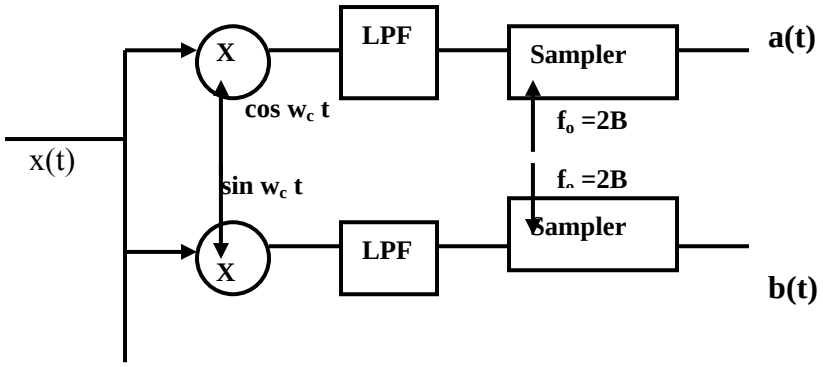


Fig.1.2.6 *A scheme for obtaining real lowpass equivalent a(t) and b(t) from narrow bandpass signal x(t) and their sampling*

Let us now have a look at the next figure (**Fig. 1.2.7**), showing the spectrum of x(t) and also the spectra of a train of sampling impulses at an arbitrary rate f_{1s}. See that, if we make $m.f_{1s} = f_c$ ('m' is an integer) and select f_{1s} such that, after sampling, there is no spectral overlap (or aliasing), then,

$$f_{1smin.} = 2.B$$

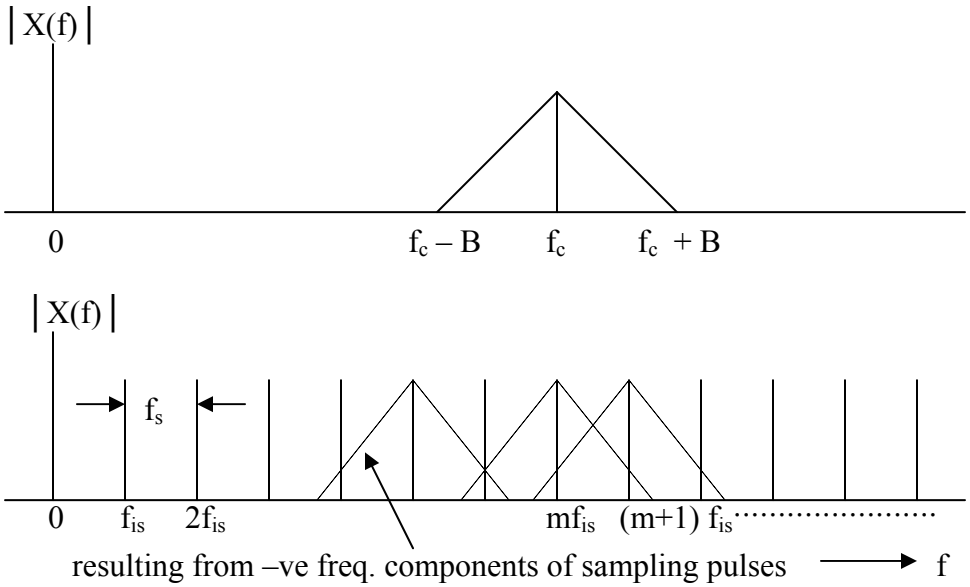


Fig. 1.2.7 *The spectrum of x(t) and the spectra of a train of sampling impulses at an arbitrary rate f_{1s}*

Further, even if we don't satisfy the above condition (i.e. $m.f_{1s} = f_c$) precisely but relax it such that, $(f_c\text{-}B) < m.f_{1s} < (f_c+B)$ for some integral m, it is possible to avoid spectral overlap for the shifted spectrum (after sampling) provided, $2 \times 2B < f_{1s} = 4 \times 2B$ [The factor of 2 comes to avoid overlapping from shifted spectra due to –ve frequency of the sampling impulses]. These basic issues on sampling will be useful in subsequent modules.

Problems

Q1.2.1) Why an electrical signal should be represented carefully and precisely?

Q1.2.2) How can a random electrical signal be represented?

Q1.2.3) Mention two reasons why a signal should be sampled for transmission through digital communication system.

Q1.2.4) Mention three benefits of representing a real narrowband signal by its equivalent complex baseband form

Module 1

Introduction to Digital Communications and Information Theory

Lesson 3

Information Theoretic Approach to Digital Communications

After reading this lesson, you will learn about

- ***Scope of Information Theory***
- ***Measures of self and mutual information***
- ***Entropy and average mutual information***

The classical theory of information plays an important role in defining the principles as well as design of digital communication systems. Broadly, Information Theory provides us answer to questions such as:

a) What is information and how to quantify and measure it?
b) What kind of processing of information may be possible to facilitate transmission of information efficiently through a transmission medium?
c) What if at all, are the fundamental limits for such processing of information and how to design efficient schemes to facilitate transmission (or storage) of information?
d) How should devices be designed to approach these limits?

More specifically, knowledge of information theory helps us in efficient design of digital transmission schemes and also in appreciating scope and limits of the extent of improvement that may be possible.

In this Lesson, we introduce the concepts of 'information' the way they are used in Digital Communications. It is customary and convenient to define information through the basic concepts of a statistical experiment.

Let us consider a statistical experiment, having a number of outcomes, rather than a single outcome. Examples of several such experiments with multiple outcomes can be drawn from the theory of Digital Communications and a few are mentioned below.

Examples of random experiments with multiple outcomes

1. A signal source with finite and discrete number of possible output letters may be viewed as a statistical experiment wherein an outcome of interest may be 'sequences of source letters'.

2. Another example, frequently referred to, is that of a discrete input-discrete output communication channel where the input and output letters (or sequences) may be the outcomes of interest. We discuss more about such a channel later.

Joint Experiment

Let us consider a two-outcome experiment. Let, 'x' and 'y' denote the two outcomes where, 'x' denotes a selection from the set of alternatives

$$X = \{a_1, a_2, ...a_K\} \qquad 1.3.1$$

K is the number of elements in the set X. Similarly, let 'y ' denote a selection from the set of alternatives

$Y = \{b_1, b_2, ...b_J\}$, with 'J' being the number of elements in set Y.

Now, the set of pairs

$\{a_k, b_j\}$ $\quad 1 \le k \le K$ and $1 \le j \le J$, forms a joint sample space.

Let, the corresponding joint probability of $P(x = a_k \ \& \ y = b_j)$ over the joint sample space be denoted as,

$P_{XY}(a_k, b_j)$ where, $1 \le k \le K$ and $\quad 1 \le j \le J$ $\qquad$ 1.3.2

We are now in a position to define a *joint ensemble XY* that consists of the joint sample space and the probability measures of its elements $\{a_k, b_j\}$.

An *event* over the joint ensemble is defined as a subset of elements in the joint sample space.

Example: In XY joint ensemble, the event that x= a_k corresponds to the subset of pairs $\{(a_k, b_1); (a_k, b_2); .....(a_k, b_J)\}$. So, we can write an expression of $P_X(a_k)$ as:

$$P_X(a_k) = \sum_{j=1}^{J} P_{XY}(a_k, b_j), \text{ or in short, } P(x) = \sum_{y} P(x, y) \qquad 1.3.3$$

Conditional Probability

The conditional probability that the outcome 'y' is 'b_j' given that the outcome x is a_k is defined as

$$P_{Y|X}(b_j | a_k) = \frac{P_{XY}(a_k, b_j)}{P_X(a_k)}, \text{ assuming that, } P(a_k) > 0 \qquad 1.3.4$$

or in short, $\quad P(y|x) = \dfrac{p(x, y)}{p(x)}$

Two events x = a_k and y = b_j are said to be statistically independent if,

$$P_{XY}(a_k, b_j) = P_X(a_k).P_Y(b_j)$$

So in this case,

$$P_{Y|X}(b_j | a_k) = \frac{P_X(a_k).P_Y(b_j)}{P_X(a_k)} = P_Y(b_j) \qquad 1.3.5$$

Two ensembles X and Y will be statistically independent if and only if the above condition holds for each element in the joint sample space, i.e., for all j-s and k-s. So, two ensembles X and Y are statistically independent if all the pairs (a_k, b_j) are statistically independent in the joint sample space.

The above concept of joint ensemble can be easily extended to define more than two (many) outcomes.

We are now in a position to introduce the concept of 'information'. For the sake if generality, we first introduce the definition of 'mutual information'.

Mutual Information

Let, $X = \{a_1, a_2, ..., a_K\}$
$Y = \{b_1, b_2, ..., b_J\}$ and

in a joint ensemble with joint probability assignments $P_{XY}(a_k, b_j)$.

Now, the mutual information between the events $x=a_k$ and $y=b_j$ is defined as,

$$I_{X;Y}(a_k;b_j) \triangleq \log_2 \frac{P_{X|Y}(a_k|b_j)}{P_X(a_k)}, \text{ bits}$$

$$= \log_2 \frac{\text{a posteriori probability of 'x'}}{\text{a priori probability of 'x'}}, \text{ bits}$$

i.e, Mutual Information $= \log_2 \dfrac{\text{ratio of conditional probability of } a_k \text{ given by } b_j}{\text{probability of } a_k}$

1.3.6

This is the information provided about the event $x = a_k$ by the occurrence of the event $y = b_j$. Similarly it is easy to note that,

$$I_{Y;X}(b_j;a_k) = \log_2 \frac{P_{Y|X}(b_j|a_k)}{P_Y(b_j)} \qquad 1.3.7$$

Further, it is interesting to note that,

$$I_{Y;X}(b_j;a_k) = \log_2 \frac{P_{Y|X}(b_j|a_k)}{P_Y(b_j)} = \log_2 \frac{P_{XY}(a_k,b_j)}{P_X(a_k)} \cdot \frac{1}{P_Y(b_j)}$$

$$= \log_2 \frac{P_{X|Y}(a_k|b_j)}{P_X(a_k)} = I_{X;Y}(a_k;b_j)$$

$$\therefore I_{X;Y}(a_k;b_j) = I_{Y;X}(b_j;a_k) \qquad 1.3.8$$

i.e, the information obtained about a_k given b_j is same as the information that may be obtained about bj given that $x = a_k$.

Hence this parameter is known as 'mutual information'. Note that mutual information can be negative.

Self-Information

Consider once again the expression of mutual information over a joint ensemble:

$$I_{X;Y}(a_k;b_j) = \log_2 \frac{P_{X|Y}(a_k|b_j)}{P_X(a_k)} \text{ bits,} \qquad 1.3.9$$

Now, if $P_{X|Y}(a_k|b_j) = 1$ i.e., if the occurrence of the event $y=b_j$ certainly specifies the outcome of $x = a_k$ we see that,

$$I_{X;Y}(a_k;b_j) = I_X(a_k) = \log_2 \frac{1}{P_X(a_k)} \text{ bit} \qquad 1.3.10$$

This is defined as the self-information of the event $x = a_k$. It is always non-negative. So, we say that self-information is a special case of mutual information over a joint ensemble.

Entropy (average self-information)

The *entropy* of an ensemble is the average value of all self information over the ensemble and is given by

$$H(X) = \sum_{k=1}^{K} P_X(a_k) \log_2 \frac{1}{P_X(a_k)}$$

$$= -\sum_x p(x) \log p(x), \text{ bit} \qquad 1.3.11$$

To summarize, self-information of an event $x = a_k$ is

$$I_X(a_k) = \log_2 \frac{1}{P_X(a_K)} \text{ bit}$$

while the *entropy* of the ensemble X is

$$H(X) = \sum_{k=1}^{K} P_X(a_k) \log_2 \frac{1}{P_X(a_k)} \text{ bit.}$$

Average Mutual Information

The concept of averaging information over an ensemble is also applicable over a joint ensemble and the average information, thus obtained, is known as *Average Mutual Information:*

$$I(X;Y) \triangleq \sum_{k=1}^{K} \sum_{j=1}^{J} P_{XY}(a_k;b_j) \log_2 \frac{P_{X|Y}(a_k|b_j)}{P_X(a_k)} \text{ bit} \qquad 1.3.12$$

It is straightforward to verify that,

$$I(X;Y) = \sum_{k=1}^{K}\sum_{j=1}^{J} P_{XY}(a_k, b_j)\log_2 \frac{P_{X|Y}(a_k|b_j)}{p_X(a_k)}$$

$$= \sum_{k=1}^{K}\sum_{j=1}^{J} P_{XY}(a_k, b_j)\log_2 \frac{P_{Y|X}(b_j|a_k)}{p_Y(b_j)} = \text{I(Y;X)} \qquad 1.3.13$$

The unit of *Average Mutual Information* is bit. Average mutual information is not dependent on specific joint events and is a property of the joint ensemble. Let us consider an example elaborately to appreciate the significance of self and mutual information.

Example of a Binary Symmetric Channel (BSC)

Let us consider a channel input alphabet X = {a_1, a_2} and a channel output alphabet Y = { b_1, b_2}. Further, let $P(b_1|a_1) = P(b_2|a_2) = 1-\varepsilon$ and $P(b_1|a_2) = P(b_2|a_1) = \varepsilon$. This is an example of a binary channel as both the input and output alphabets have two elements each. Further, the channel is symmetric and unbiased in its behavior to the two possible input letters $a_1$and a_2. Note that the output alphabet Y need not necessarily be the same as the input alphabet in general.

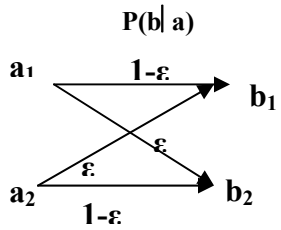


Fig.P.1.3.1 Representation of a binary symmetric channel; ε indicates the probability of an error during transmission.

Usually, if a_1 is transmitted through the channel, b_1 will be received at the output provided the channel has not caused any error. So, ε in our description represents the probability of the channel causing an error on an average to the transmitted letters.

Let us assume that the probabilities of occurrence of a_1 and a_2 are the same, i.e. $P_X(a_1) = P_X(a_2) = 0.5$. A source presenting finite varieties of letters with equal probability is known as a *discrete memory less source (DMS)*. Such a source is unbiased to any letter or symbol.

Now, observe that,

$$P_{XY}(a_1, b_1) = P_{XY}(a_2, b_2) = \frac{1-\in}{2}$$

And $P_{XY}(a_1, b_2) = P_{XY}(a_2, b_1) = \frac{\in}{2}$

Note that the output letters are also equally probable, i.e.

$$\sum_{j=1}^{2} P(y_j|x_i)p(x_i) = \frac{1}{2} \text{ for}$$

So, $P_{X|Y}(a_1|b_1) = P_{X|Y}(a_2|b_2) = 1-\epsilon$ and $P_{X|Y}(a_1|b_2) = P_{X|Y}(a_2|b_1) = \epsilon$

Therefore, possible mutual information are:

$I_{X;Y}(a_1;b_1) = \log_2 2(1-\epsilon) = I_{X;Y}(a_2;b_2)$ and $I_{X;Y}(a_1;b_2) = \log_2 2\epsilon = I_{X;Y}(a_2;b_1)$

Next, we determine an expression for the average mutual information of the joint ensemble XY:

$$I(X;Y) = (1-\epsilon)\log_2 2(1-\epsilon) + \epsilon \log_2 2\epsilon \text{ bit}$$

Some observations

(a) Case#1: Let, $\epsilon \rightarrow 0$. In this case,

$I_{X;Y}(a_1;b_1) = I_{X;Y}(a_2;b_2) = \log_2 2(1-\epsilon)$ bit = 1.0 bit (approx.)

$I_{X;Y}(a_1;b_2) = I_{X;Y}(a_2;b_1) = \log_2 2\epsilon$ bit $\rightarrow$ a large negative quantity

This is an almost noiseless channel. When a_1 is transmitted we receive b_1 at the output almost certainly.

(b) Case#2: Let us put $\epsilon = 0.5$. In this case,

$I_{X;Y}(a_1;b_1) = I_{X;Y}(a_1;b_2) = 0$ bit

It represents a very noisy channel where no information is received. The input and output are statistically independent, i.e. the received symbols do not help in assessing which symbols were transmitted.

(c) Case#3 : Fortunately, for most practical scenario, $0 < \epsilon << 0.5$ and usually,

$$I_{X;Y}(a_1;b_1) = I_{X;Y}(a_2;b_2) = \log_2 2(1-\epsilon) \text{ bit} = 1.0 \text{ bit (approx.)}$$

So, reception of the letter $y=b_1$ implies that it is highly likely that a_1 was transmitted. Typically, in a practical digital transmission system, $\epsilon \cong 10^{-3}$ or less.

The following figure, **Fig 1.3.1** shows the variation of I(X;Y) vs. ϵ.

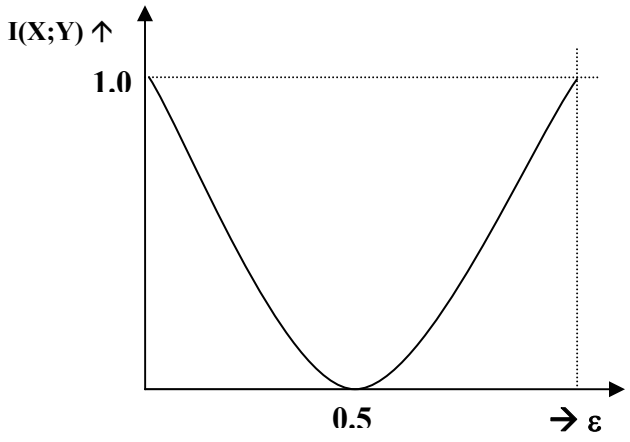


Fig. 1.3.1 *Variation of average mutual information vs. ε for a binary symmetric channel*

Problems

Q1.3.1) Why the theory of information is relevant for understanding the principles of digital communication systems?

Q1.3.2) Consider a random sequence of 16 binary digits where the probability of occurrence is 0.5. How much information is contained in this sequence?

Q1.3.3) A discrete memory less source has an alphabet {1,2,…,9}. If the probability of generating digit is inversely proportional to its value. Determine the entropy of source.

Q1.3.4) Describe a situation drawn from your experience where concept of mutual information may be useful.

Module 1

Introduction to Digital Communications and Information Theory

Lesson 4

Coding for discrete sources

After reading this lesson, you will learn about

- ***Need for coding source letters***
- ***Variable length coding***
- ***Prefix – condition code***
- ***Kraft Inequality Theorem***
- ***Huffman Coding***
- ***Source Coding Theorem***

All source models in information theory may be viewed as random process or random sequence models. Let us consider the example of a discrete memory less source (DMS), which is a simple random sequence model.

A DMS is a source whose output is a sequence of letters such that each letter is independently selected from a fixed alphabet consisting of letters; say a_1, a_2, …….a_k. The letters in the source output sequence are assumed to be random and statistically independent of each other. A fixed probability assignment for the occurrence of each letter is also assumed. Let us, consider a small example to appreciate the importance of probability assignment of the source letters.

Let us consider a source with four letters a_1, a_2, a_3 and a_4 with $P(a_1)=0.5$, $P(a_2)=0.25$, $P(a_3)= 0.13$, $P(a_4)=0.12$. Let us decide to go for binary coding of these four source letters. While this can be done in multiple ways, two encoded representations are shown below:

Code Representation#1: a_1: 00, a_2:01, a_3:10, a_4:11

Code Representation#2: a_1: 0, a_2:10, a_3:001, a_4:110

It is easy to see that in method #1 the probability assignment of a source letter has not been considered and all letters have been represented by two bits each. However in the second method only a_1 has been encoded in one bit, a_2 in two bits and the remaining two in three bits. It is easy to see that the average number of bits to be used per source letter for the two methods are not the same. ($\bar{a}$ for method #1=2 bits per letter and $\bar{a}$ for method #2 < 2 bits per letter). So, if we consider the issue of encoding a long sequence of letters we have to transmit less number of bits following the second method. This is an important aspect of source coding operation in general. At this point, let us note the following:

a) We observe that assignment of small number of bits to more probable letters and assignment of larger number of bits to less probable letters (or symbols) may lead to efficient source encoding scheme.

b) However, one has to take additional care while transmitting the encoded letters. A careful inspection of the binary representation of the symbols in method #2 reveals that it may lead to confusion (at the decoder end) in deciding the end of binary representation of a letter and beginning of the subsequent letter.

So a source-encoding scheme should ensure that

1) The average number of coded bits (or letters in general) required per source letter is as small as possible and

2) The source letters can be fully retrieved from a received encoded sequence.

In the following we discuss a popular variable-length source-coding scheme satisfying the above two requirements.

Variable length Coding

Let us assume that a DMS U has a K- letter alphabet $\{a_1, a_2, \ldots\ldots\ldots a_K\}$ with probabilities $P(a_1)$, $P(a_2)$,……. $P(a_K)$. Each source letter is to be encoded into a codeword made of elements (or letters) drawn from a code alphabet containing D symbols. Often for ease of implementation a binary code alphabet (D = 2) is chosen. As we observed earlier in an example, different codeword may not have same number of code symbols. If n_k denotes the number of code symbols corresponding to the source letter a_k , the average number of code letters per source letter ($\bar{n}$) is:

$$\bar{n} = \sum_{k=1}^{K} P(a_k) n_k \qquad 1.4.1$$

Intuitively, if we encode a very long sequence of letters from a DMS, the number of code letters per source letter will be close to $\bar{n}$.

Now, a code is said to be *uniquely decodable* if for each source sequence of finite length, the sequence of code letters corresponding to that source sequence is different from the sequence of code letters corresponding to any other possible source sequence.

We will briefly discuss about a subclass of uniquely decodable codes, known as *prefix condition code*. Let the code word in a code be represented as

$\overline{x_k} = (x_{k,1}, x_{k,2}, ..., x_{k,n_k})$, where $x_{k,1}, x_{k,2}, ..., x_{k,n_k}$ denote the individual code letters (when D=2, these are 1 or 0). Any sequence made up of an initial part of $\overline{x_k}$ that is $x_{k,1}, x_{k,2}, ..., x_{k,i}$ for $i \le n_k$ is called a prefix of $\overline{x_k}$.

A prefix condition code is a code in which no code word is the prefix of any other codeword.

Example: consider the following table and find out which code out of the four shown is / are prefix condition code. Also determine $\bar{n}$ for each code.

Source letters:- a_1, a_2, a_3 and a_4
P(a_k) :- $P(a_1)=0.5$, $P(a_2)=0.25$, $P(a_3)= 0.125$, $P(a_4)=0.125$
Code Representation#1: a_1: 00, a_2:01, a_3:10, a_4:11
Code Representation#2: a_1: 0, a_2:1, a_3:00, a_4:11
Code Representation#3: a_1: 0, a_2:10, a_3:110, a_4:111
Code Representation#4: a_1: 0, a_2:01, a_3:011, a_4:0111

A prefix condition code can be decoded easily and uniquely. Start at the beginning of a sequence and decode one word at a time. Finding the end of a code word is not a problem as the present code word is not a prefix to any other codeword.

Example: Consider a coded sequence 0111100 as per Code Representation #3 of the previous example. See that the corresponding source letter sequence is a_1, a_4, a_2, a_1.

Now, we state one important theorem known as Kraft Inequality theorem without proof.

Kraft Inequality Theorem

If the integers n_1, n_2, …….n_K satisfy the inequality

$$\sum_{k=1}^{K} D^{-n_k} \leq 1 \qquad 1.4.2$$

then a prefix condition code of alphabet size D exists with these integers as codeword lengths.

This also implies that the lengths of code words of any prefix condition code satisfy the above inequality.

It is interesting to note that the above theorem does not ensure that any code whose codeword lengths satisfy the inequality is a prefix condition code. However it ensures that a prefix condition code exists with code word length n_1, n_2, …….n_K.

Binary Huffman Coding (an optimum variable-length source coding scheme)

In Binary Huffman Coding each source letter is converted into a binary code word. It is a prefix condition code ensuring minimum average length per source letter in bits.

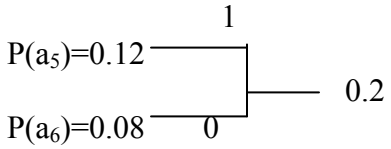
Let the source letters a_1, a_2, …….a_K have probabilities $P(a_1)$, $P(a_2)$,……. $P(a_K)$ and let us assume that $P(a_1) \geq P(a_2) \geq P(a_3) \geq \ldots \geq P(a_K)$.

We now consider a simple example to illustrate the steps for Huffman coding.

Steps to calculate Huffman Coding

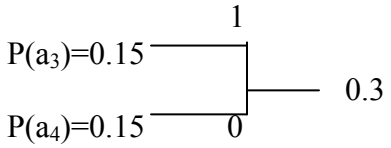
Example Let us consider a discrete memoryless source with six letters having $P(a_1)=0.3$, $P(a_2)=0.2$, $P(a_3)=0.15$, $P(a_4)=0.15$, $P(a_5)=0.12$ and $P(a_6)=0.08$.

1. Arrange the letters in descending order of their probability (here they are arranged).
2. Consider the last two probabilities. Tie up the last two probabilities. Assign, say, 0 to the last digit of representation for the least probable letter (a_6) and 1 to the last digit of representation for the second least probable letter (a_5). That is, assign '1' to the upper arm of the tree and '0' to the lower arm.



3. Now, add the two probabilities and imagine a new letter, say b_1, substituting for a_6 and a_5. So $P(b_1) = 0.2$. Check whether a_4 and b_1 are the least likely letters. If not, reorder the letters as per Step#1 and add the probabilities of two least likely letters. For our example, it leads to:
   $P(a_1)=0.3$, $P(a_2)=0.2$, $P(b_1)=0.2$, $P(a_3)=0.15$ and $P(a_4)=0.15$

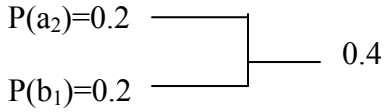
4. Now go to Step#2 and start with the reduced ensemble consisting of a_1 , a_2 , a_3 , a_4 and b_1. Our example results in:



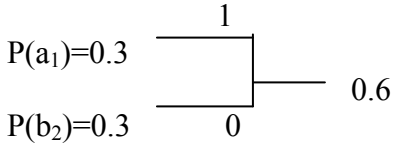
   Here we imagine another letter b_1, with $P(b_2)=0.3$.

5. Continue till the first digits of the most reduced ensemble of two letters are assigned a '1' and a '0'.

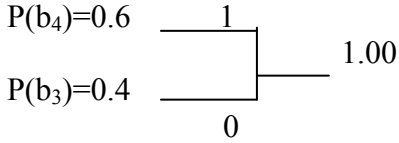
   Again go back to the step (2): $P(a_1)=0.3$, $P(b_2)=0.3$, $P(a_2)=0.2$ and $P(b_1)=0.2$. Now we consider the last two probabilities:



   So, $P(b_3)=0.4$. Following Step#2 again, we get, $P(b_3)=0.4$, $P(a_1)=0.3$ and $P(b_2)=0.3$.
   Next two probabilities lead to:



   with $P(b_4) = 0.6$. Finally we get only two probabilities



6. Now, read the code tree inward, starting from the root, and construct the codewords. The first digit of a codeword appears first while reading the code tree inward.

Hence, the final representation is: a_1=11, a_2=01, a_3=101, a_4=100, a_5=001, a_6=000.

A few observations on the preceding example

1. The event with maximum probability has least number of bits.
2. Prefix condition is satisfied. No representation of one letter is prefix for other. Prefix condition says that representation of any letter should not be a part of any other letter.
3. Average length/letter (in bits) after coding is
   $= \sum_i P(a_i)n_i = 2.5$ bits/letter.
4. Note that the entropy of the source is: H(X)=2.465 bits/symbol. Average length per source letter after Huffman coding is a little bit more but close to the source entropy. In fact, the following celebrated theorem due to C. E. Shannon sets the limiting value of average length of codewords from a DMS.

Source Coding Theorem (for coding source letters with variable length codes)

Given a finite source ensemble U with entropy H (U) and given a code alphabet of D symbols, it is possible to assign code words to the source letters in such a away that the prefix condition is satisfied and the average length of the code words $\bar{n}$, satisfies the inequality.

$$\bar{n} < \frac{H(U)}{\log_2 D} + 1,$$

Further, for any uniquely decodable set of code words,

$$\bar{n} \geq \frac{H(U)}{\log_2 D}$$

As all prefix condition codes are uniquely decodable, for such a code,

$$\frac{H(U)}{\log D} \leq \bar{n} \leq \frac{H(U)}{\log D} + 1$$

For binary representation (i.e. D = 2, as in our example),

$$H(U) \leq \bar{n} < H(U) + 1$$

The equality holds if,

$$P(a_k) = D^{-n_k} \quad \text{for } 1 \leq k \leq K, \qquad 1.4.3$$

where n_k is the length of the k-th source letter.

Problems

Q1.4.1) What is a prefix-condition code?

Q1.4.2) Is optimum source coding possible when the length of coded letters is the same (i.e. fix length coding)?

Q1.4.3) What is the significance of source coding theorem?

Module 2

Random Processes

Lesson 5

Introduction to Random Variables

After reading this lesson, you will learn about

- ***Definition of a random variable***
- ***Properties of cumulative distribution function (cdf)***
- ***Properties of probability density function (pdf)***
- ***Joint Distribution***

- A random variable (RV) is a real number x(s) assigned to every outcome 's' of an experiment. An experiment may have a finite or an infinite number of outcomes. Ex: i) Voltage of a random noise source, ii) gain points in a game of chance.
- An RV is also a function whose domain is the set 'S' of outcomes of the experiment.
- Domain of a function: A 'function' y(t) is a rule of correspondence between values of 't' and 'y'. All possible values of the independent variable 't' form a set 'T', known as the 'domain of y(t)'.
- The values of the dependent variable y(t) form a set Y on the y-axis and the set Y is called the range of the function y(t).
- The rule of correspondence between 't' and 'y' may be a table, a curve, a formula or any other precise form of description.
- Events: Any subset of S, the set of valid outcomes, may define an event. The subset of outcomes usually depends on the nature of interest / investigation in the experiment. Each event, however, must have a notion of probability associated with it, represented by a non-negative real number less than or equal to unity.

A more formal definition of a Random Variable (RV)

A random variable 'X' is a process of assigning a (real) number x(s) to every outcome s of a statistical experiment. The resulting function must satisfy the following two conditions:

1. The set $\{X \leq a\}$ is an event for every 'a'.
2. The probabilities of the events $\{X = +\infty\}$ and $\{X = -\infty\}$ equal zero, i.e., $P\{X = +\infty\} = \{X = -\infty\} = 0$.

A complex random variable is defined as, $\mathbf{z} = x + jy$ where x and y are real random variables.

Cumulative Distribution Function [cdf]

The cdf of a random variable X is the function $F_x(a) = P\{X \leq a\}$, for $-\infty < a < \infty$. Here, note that $\{X \leq a\}$ denotes an event and 'a' is a real number. The subscript 'x' is added to F() to remind that 'X' is the random variable. The cdf $F_x(a)$ is a positive number which depends on 'a'. We may also use the simpler notation F (x) to indicate the cdf $F_x(a)$ when there is no confusion.

A complex RV **z** = x + jy has no cdf in the ordinary sense because, an inequality of the type (x + jy) ≤ (a + jb) has no meaning. Hence, the statistics of a complex RV (like **z**) are specified in terms of the joint statistics of the random variables x & y.

Some fundamental properties of cdf $F_x(a)$

For a small and positive ε, let, $F_x(a^+) = \lim_{\varepsilon->0} F_x(a+\varepsilon)$

i.e., $F_x(a^+)$ is the cdf of X when the number 'a' is approached from the right hand side (RHS) of the real number axis.

Similarly, let, for a small and positive ε, $F_x(a^-) = \lim_{\varepsilon->0} F_x(a-\varepsilon)$

i.e., $F_x(a^-)$ is the cdf of X when the number 'a' is approached from the left hand side (LHS) of the real number axis.

Property #1 $F_x(+\infty) = 1$ and $F_x(-\infty) = 0$

Hint: $F_x(+\infty) = P\{X \le \infty\} = P\{S\} = 1$ and $F_x(-\infty) = P\{X \le -\infty\} = 0$

Property #2 The cdf $F_x(a)$ is a non-decreasing function of 'a', i.e.,

if $a_1 < a_2$, $F_x(a_1) \le F_x(a_2)$

Hint: An event $\{x \le a_1\}$ is a subset of the event $\{x \le a_2\}$ because, if $x(s) \le a_1$ for some's', then $x(s) \le a_2$ too.

Hence, $P\{x \le a_1\} \le P\{x \le a_2\}$

From properties #1 and #2, it is easy to see that $F_x(a)$ increases from 0 to 1 as 'a' increases from -∞ to +∞. The particular value of $a = a_m$ such that, $F_x(a_m) = 0.5$ is called the *median* of the RV 'X'.

Property #3 If $F_x(a_0) = 0$ then, $F_x(a) = 0$ for every $a \le a_0$

Hint: As $F_x(-\infty) = 0$ and $F_x(a_1) \le F_x(a_2)$ when $a_1 < a_2$.

Property #4 $P\{x > a\} = 1 - F_x(a)$

Hint: The events $\{x \le a\}$ *and* $\{x \ge a\}$ are mutually exclusive and $\{x \le a\} \cup \{x > a\} = S$

Property #5 The function $F_x(a)$ is continuous from the right, i.e.,

$F_x(a^+) = F_x(a)$

Hint: Observe that, $P\{x \le a + \in\} \to F_x(a)$ as $\in \to 0$

Because, $P\{x \le a + \in\} \to F_x(a + \in)$ and $F_x'(a + \in) \to F_x(a^+)$

Also observe that the set $\{x \le a + \in\}$ tends to the set $\{x \le a\}$ *as* $\in \to 0$

Property #6 $P\{a_1 < x \le a_2\} = F_x(a_2) - F_x(a_1)$

Hint: The events $\{x \le a_1\}$ *and* $\{a_1 < x \le a_2\}$ are mutually exclusive. Further, $\{ x \le a_2 \} = \{ x \le a_1 \} + \{ a_1 < x \le a_2 \}$ and hence, $P\{ x \le a_2 \} = P\{ x \le a_1 \} + P\{ a_1 < x \le a_2 \}$.

Property #7 $P\{x = a\} = F_x(a) - F_x(a^-)$

Hint: Put $a_1 = a - \in$ *and* $a_2 = a$ in Property #6.

Property #8 $P\{a_1 \le x \le a_2\} = F_x(a_2) - F_x(a_1^-)$

Hint: Note that, $\{a_1 \le x \le a_2\} = \{a_1 < x \le a_2\} + \{x = a_1\}$

Now use Property #6 and Property #7.

- **Continuous RV** A random variable X is of continuous type if its cdf $F_x(a)$ is continuous. In this case, $F_x(a^-) = F_x(a)$ and hence, $P\{x = a\} = 0$ for every 'a'. [Property #7]

- **Discrete RV** A random variable X is of discrete type if $F_x(a)$ is a staircase function. If 'a_i' indicates the points of discontinuity in $F_x(a)$, we have,
  $F_x(a_i) - F_x(a_i^-)$
  $= P\{x = a_i\} = p_i$
  In this case, the statistics of X are determined in terms of a_i-s and p_i-s.
- **Mixed RV** A random variable X is of mixed type if $F_x(a)$ is discontinuous but not a staircase.

Probability Density Function (pdf) / Frequency Function / Density Function

- The derivative $f_x(a) = \dfrac{dF_x(a)}{da}$ is called the probability density function (pdf) / density function / frequency function of the random variable X.
  For a discrete random variable X taking values a_i-s with probabilities p_i,
  $f_x(a) = \sum_i p_i \delta(a - a_i)$, where $p_i = P\{x = a_i\}$ and $\delta(x)$ is the impulse function.

Properties of Probability Density Function (pdf)

- From the monotonicity of $F_x(a)$, $f_x(a) \ge 0$
- Integrating the basic expression of $f_x(a) = \dfrac{dF_x(a)}{da}$ from - ∞ to 'a' and recalling that $F_x(-\infty) = 0$, we get,

$$F_x(a) = \int_{-a}^{a} f_x(\tau) d\tau \qquad 2.5.1$$

- $\int_{-a}^{a} f_x(a)da = 1$ 2.5.2

- $F_x(a_2) - F_x(a_1) = \int_{a_1}^{a_2} f_x(\tau)d\tau$ or $P\{a_1 < x \le a_2\} = \int_{a_1}^{a_2} f_x(\tau)d\tau$ 2.5.3

  Note, if X is a continuous RV, $P\{a_1 < x \le a_2\} = P\{a_1 \le x \le a_2\}$. However, if $F_x(a)$ is discontinuous at, say a_2, the integration in 2.5.3 must include the impulse at x = a_2.

- For a continuous RV x, with $a_1 = a$ & $a_2 = a + \Delta a$, we may write from 2.5.3: $P\{a_1 \le x \le a_2\} \cong f_x(x).\Delta a$ for $\Delta a \to 0$ which means,

  $$f_x(a) = \lim_{\Delta a \to 0} \frac{P\{a \le x \le a + \Delta a\}}{\Delta a} \quad 2.5.4$$

Eq. 2.5.4 shows that for $\Delta a \to 0$, the probability that the random variable X is in a small interval Δa is proportional to $f_x(a)$, i.e. $f_x(a) \propto P\{a \le x \le a + \Delta a\}$. When the interval Δa includes the point a = a_{mode} where $f_x(a)$ is maximum, a_{mode} is called the *mode* or *most likely value* of the random variable X.

Now, if f(x) denotes the pdf of a random variable X, then, $P\{ x_0 < X \le x_0 + dx \} = f(x_0)dx$, for vanishingly small dx.

If the random variable X takes values between - ∞ and + ∞, then its average (expectation or mean) value is: $E[X] = \lim_{N\to\infty} \sum_{i=1}^{N} x_i P\{X = x_i\} = \int_{-\infty}^{+\infty} xf(x)dx$

The mean-squared value of a random variable X is defined as:

$$E[X^2] = \lim_{N\to\infty} \sum_{i=1}^{N} x_i^2 P\{X = x_i\} = \int_{-\infty}^{+\infty} x^2 f(x)dx \quad 2.5.5$$

The variance of a random variable X is its mean squared value about its mean, i.e.

$$\text{Variance of X} = E\left[\{X - E(X)\}^2\right] = \lim_{N\to\infty} \sum_{i=1}^{N} \{x_i - E(X)\}^2 P\{X = x_i\}$$

$$= \int_{-\infty}^{+\infty} \{x - E(X)\}^2 f(x)dx \quad 2.5.6$$

The variance of a random variable is popularly denoted by σ^2. The positive square root of the variance is called the 'standard deviation' of the random variable (from its mean value).

In addition to the above basic statistical parameters, several 'moments' are also defined to provide detail description of a random experiment. These are the means or averages of

positive integer powers of the random variable X. In general, the n-th moment of X is expressed as:

$$\alpha_n = E\left[X^n\right] = \lim_{N\to\infty}\sum_{i=1}^{N} x_i^n P\{X = x_i\} = \int_{-\infty}^{+\infty} x^n f(x)\,dx \qquad 2.5.7$$

It is easy to verify and good to remember that, $\alpha_0 = 1$, $\alpha_1 = E[X]$, $\alpha_2 = E[X^2]$, the mean squared value of X and, the variance $\sigma^2 = \alpha_2 - \alpha_1^2$

Another class of moments, known as 'central moments' of the distribution is obtained by subtracting the mean of X from its values and taking moments as defined above. That is, the n-th central moment of X is defined as:

$$\mu_n = E\left\{\left[X - E(X)\right]^n\right\} = \lim_{N\to\infty}\sum_{i=1}^{N}\left[x_i - E(X)\right]^n P(X = x_i)$$

$$= \int_{-\infty}^{+\infty}\left[x - E(X)\right]^n f(x)\,dx \qquad 2.5.8$$

This time, note that, $\mu_0 = 1$, $\mu_1 = 0$ and $\mu_2 = \sigma^2$.

Two or more random variables

Let X and Y are two random variables. We define a joint distribution function F(x, y) as the (joint) probability of the combined events { $X \le x$ } and { $Y \le y$ }, i.e.,

$$F(x, y) = \text{Prob.}[\, X \le x \text{ and } Y \le y\,] \qquad 2.5.9$$

Extending our earlier discussion over a joint event, it is easy to note that,

$$F(-\infty, y) = 0,\ F(x, -\infty) = 0 \text{ and } F(+\infty, +\infty) = 1 \qquad 2.5.10$$

The joint probability density function [joint pdf] can be defined over the joint probability space when the partial derivatives exist and are continuous:

$$f(x,y) = \frac{\partial^2 F(x,y)}{\partial x \partial y} \qquad 2.5.11$$

This implies,

$$\int_{a_1}^{a_2}\int_{b_1}^{b_2} f(x,y)\,dx\,dy = F(a_2,b_2) - F(a_1,b_2) - \left[F(a_2,b_1) - F(a_1,b_1)\right]$$

$$= \text{Prob.}\left[a_1 < X \le a_2 \text{ and } b_1 < Y \le b_2\right] \qquad 2.5.12$$

For two or more random variables with defined joint distribution functions F (x, y) and joint pdf f (x , y), several second order moments can be defined from the following general expression:

$$\alpha_{ij} = E\left(x^i y^j\right) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} x^i y^j f\left(x,y\right) dydx\,, \qquad 2.5.13$$

The following second order moment α_{11} is called the correlation function of X and Y:

$$\alpha_{11} = E\left(X\ Y\right) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} xy\, f\left(x,y\right) dydx \qquad 2.5.14$$

The set of second order central moments can also be expressed in general as:

$$\mu_{ij} = E\left\{\left[X - E\left(X\right)\right]^i\right\} E\left\{\left[Y - E\left(Y\right)\right]^j\right\}$$

$$= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \left[x - E\left(X\right)\right]^i \left[Y - E\left(Y\right)\right]^j\ f\left(x,y\right) dydx\,, \qquad 2.5.15$$

The reader may try out some exercise at this point to verify the following:

$\mu_{20} = \alpha_{20} - \alpha_{10}^2$; $\mu_{02} = \alpha_{02} - \alpha_{01}^2$; and $\mu_{11} = \alpha_{11} - \alpha_{01}.\alpha_{10}$;

Now, let $F_X(x)$ denote the cdf of X and $F_Y(y)$ denote the cdf of Y. That is,

$F_X\left(x\right) = \text{Prob}\left[X \le x \text{ and } Y \le \infty\right]$ and $F_Y\left(y\right) = \text{Prob}\left[Y \le y \text{ and } X \le \infty\right]$.

It is straightforward to see that,

$$F_X\left(x\right) = \text{Prob}\left[X \le x \text{ and } Y \le \infty\right] = \int_{-\infty}^{x}\int_{-\infty}^{+\infty} f\left(x,y\right) dydx \qquad 2.5.16$$

$$F_Y\left(y\right) = \text{Prob}\left[\ X \le \infty \text{ and } Y \le y\right] = \int_{-\infty}^{+\infty}\int_{-\infty}^{y} f\left(x,y\right) dydx \qquad 2.5.17$$

If the first order probability density functions of X and Y are now denoted by $f_X(x)$ and $f_Y(y)$ respectively, by definition, $f_X\left(x\right) = \dfrac{dF_X\left(x\right)}{dx}$ and $f_Y\left(y\right) = \dfrac{dF_Y\left(y\right)}{dy}$.

Further,

$$f_X\left(x\right) = \int_{-\infty}^{+\infty} f\left(x,y\right) dy \quad \text{and} \quad f_Y\left(y\right) = \int_{-\infty}^{+\infty} f\left(x,y\right) dx \qquad 2.5.18$$

The above expressions highlight that individual probability density functions are contained within the joint probability density function.

If the two random variables X and Y are known to be independent of each other, we may further see that,

$$\text{Prob}\left[X \le x \text{ and } Y \le y\right] = \text{Prob}\left[\ X \le x\right] \cdot \text{Prob}\left[\ Y \le y\right]$$

and hence,

$$F\left(x,y\right) = F_X\left(x\right) \cdot F_Y\left(y\right),$$

$$f\left(x,y\right) = f_X\left(x\right) \cdot f_Y\left(y\right)\ ,$$

and also,

$$\alpha_{11} = E(X,Y) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} xy \cdot f(x,y)\,dydx = E(X)E(Y) \qquad 2.5.19$$

Occasionally it is necessary to consider and analyze a joint distribution involving three or more random variables and one has to extend the above ideas carefully to obtain the desired statistics. For example, the joint distribution function F(x, y, z) for three random variables X, Y and Z, by definition is:

$$F(x,y,z) = \text{Prob}\left[X \le x \text{ and } Y \le y \text{ and } Z \le z\right] \qquad 2.5.20$$

And their joint probability density function [pdf] f(x, y, z) can be expressed in a general form as:

$$f(x,y,z) = \frac{\partial^3 F(x,y,z)}{\partial x \partial y \partial z} \qquad 2.5.21$$

Problems

Q2.5.1) Mention the two basic conditions that a random variable must satisfy?

Q2.5.2) Prove property #5 under section "Some fundamental properties of cdf $F_x(a)$.

Q2.5.3) What is a mixed RV? Explain with two examples.

Q2.5.4) Give an example where the concept of joint pdf may be useful.

Module 2

Random Processes

Lesson 6

Functions of Random Variables

After reading this lesson, you will learn about

- ***cdf of function of a random variable.***
- ***Formula for determining the pdf of a random variable.***

Let, X be a random variable and $g(a)$ is a function of a real variable a. Then, the expression $y = g(x)$ leads to a new random variable Y with the following connotation:

Let 's' indicate an outcome of a random experiment, as introduced earlier in Lesson #5. For a given 's', x(s) is a real number and $g[x(s)]$ is another real number specified in terms of x(s) and g(a). This new number is the value $y(s) = g[x(s)]$, which is assigned to the random variable Y. In brief, Y = g (X) indicates this functional relationship between the random variables X and Y.

The cdf $F_y(b)$ of the new random variable Y, so formed, is the probability of the event $\{y \le b\}$, consisting of all outcomes 's' such that $y(s) = g[x(s)] \le b$.
This means,

$$F_y(b) = P\{y \le b\} = P\{g(s) \le b \qquad 2.6.1$$

For a specific b, there may be multiple values of 'a' for which $g(a) \le b$. Let us assume that all these values of 'a' for which $g(a) \le b$, form a set on the a-axis and let us denote this set as I_y. This set is known as the point set.

So, $g[x(s)] \le b$ if x(s) is a number in the set I_y, i.e. $F_y(b) = P\{x \in Iy\}$ 2.6.2

Now, g(a) must have the following properties so that g(x) is a random variable :

a) The domain of g(a) must include the range of the random variable X.
b) For every b such that $g(a) \le b$, the set I_y must consist of the union and intersection of a countable number of intervals since then only $\{y \le b\}$ is an event.
c) The events $\{g(x) = \pm\infty\}$ must have zero probability.

Cumulative Distribution Function [cdf] of g(x)

We wish to express the cdf $F_y(b)$ of the new random variable Y where y = g(x) in term of the cdf $F_x(a)$ of the random variable X and the function g(a). To do this, we determine the set I_y on the a-axis so that $g(a) \le b$ and also the probability that the random variable X is in this set.

Let us assume that $F_x(a)$ is continuous and consider a few examples to illustrate the point.

Example #2.6.1

Let, y = g(x) = c.x + d, where c and d are constants [This is an equation of a straight line].

To find $F_y(b)$, we have to find the values of 'a' such that, c.a + d ≤ b.

$$For\ c > 0:\ ca + d \le b\ means\ \ a \le \frac{b-d}{c}$$

$$So,\ F_y(b) = P\left\{x \le \frac{b-d}{c}\right\} = F_x\left(\frac{b-d}{c}\right)$$

While, for c < 0, $ca + d \le b\ means\ a \ge \frac{b-d}{c}$ and so

$$F_y(b) = P\left\{x \ge \frac{b-d}{c}\right\} = 1 - F_x\left(\frac{b-d}{c}\right)$$

■

Example #2.6.2

Let, y = g(x) = x^2

It is easy to see that, for b < 0, $F_y(b) = 0$

However, for b ≥ 0 $a^2 \le b$ for $-\sqrt{b} \le a \le \sqrt{b}$ and hence,

$$F_y(b) = P\left\{-\sqrt{b} \le x \le \sqrt{b}\right\} = F_x\left(\sqrt{b}\right) - F_x\left(\sqrt{b}\right)$$

■

Example #2.6.3

Let us consider the following function g(a):

$$g(a) = \begin{cases} a + c, & a < -c \\ 0, & -c \le a \le c \\ a - c, & a > c \end{cases}$$

It is a good idea to sketch g(a) versus 'a' to gain a closer look at the function.

Note that, $F_y(b)$ is discontinuous at $b = g(a) = 0$ by the amount $F_x(c) - F_x(-c)$

Further,

$$for\ b \ge 0, \quad P\{y \le b\} = P\{x \le b + c\} = F_x(b + c)$$

$$\&\ for\ b < 0, \quad P\{y \le b\} = P\{x \le b - c\} = F_x(b - c)$$

■

Example #2.6.4

While we will discuss more about linear and non-linear quantizers in the next Module, let us consider the simple transfer characteristics of a linear quantizer here:

Let, $g(a) = n.s, (n-a)s < a \le ns$ where 's' is a constant, indicating a fixed step size and 'n' is an integer, representing the n-th quantization level.

Then for $y = g(x)$, the random variable Y takes values

$$b_n = ns\ with$$

$$P\{y = ns\} = P\{(n-1)s < x \le ns\} = F_x(ns) - F_x(ns - s)$$

■

Example #2.6.5

Let, $g(a) = \begin{Bmatrix} a+c, a \geq 0 \\ a-c, a<0 \end{Bmatrix}$, where 'c' is a constant. Plot g(a) versus 'a' and see that g(a) is discontinuous at a = 0, with $g(0^-) = -c$ and $g(0^+) = +c$. This implies that, $F_Y(b) = F_X(0)$, for $|b| \leq c$.

Further, for $b \geq c$, $g(a) \leq b$ *for* $a \leq b-c$; *hence*, $F_y(b) = F_x(b-c)$

$-c \leq b \leq c$, $g(a) \leq b$ *for* $a \leq c$; *hence*, $F_y(b) = F_x(0)$

$b \leq -c$, $g(a) \leq b$ *for* $a \leq b+c$; *hence*, $F_y(b) = F_x(b+c)$

■

An important step while dealing with functions of random variables is to find the point set I_y and thereby the cdf $F_Y(Y)$ when the functions g(x) and $F_X(X)$ are known. In terms of probability, it is equivalent to finding the values of the random variable X such that, $F_Y(y) = P\{Y \leq y\} = P\{X \in I_y\}$. We now briefly discuss about a concise and convenient relationship for determination of the pdf of Y, i.e $f_Y(Y)$.

Formula for determining the pdf of Y, i.e., $f_Y(Y)$:

Let, X be a continuous random variable with pdf $f_x(X)$ and g(x) be a differentiable function of x. [*i.e.* $g'(x) \neq 0$]. We wish to establish a general expression for the pdf of Y = g(X).

Note that, an event $\{y < Y \leq y+dy\}$ can be written as a union of several disjoint elementary events $\{E_i\}$.

Let, the equation $y = g(x)$ have n real roots $x_1, x_2, \ldots\ldots, x_n$,
i.e. $y - g(x_i) = 0$, for i = 1, 2, … n.

Then, the disjoint events are of the forms:

$$E_i = \{x_i - |dx_i| < X < x_i\}, \quad if\ g'(x_i)\ is - ve$$
$$or\ E_i = \{x_i < X < x_i + |dx_i|\}, \quad if\ g'(x_i)\ is + ve$$

In either case, we can write (following the basic definition of pdf), that,

Pr. of an event = (pdf at x = xi).$|dx_i|$

So, for the above disjoint events $\{E_i\}$, we may, approximately write,

$P\{E_i\}$ = Probability of event $E_i = f_X(x_i)|dx_i|$

As we have considered the events E_i – s disjoint, we may now write that,

Prob.$\{ y < Y \leq (y + dy) \} = f_Y(y). |dy|$
$= f_x(x_1). |dx_1| + f_x(x_2). |dx_2| + \ldots\ldots\ldots + f_x(x_n). |dx_n|$

$$= \sum_{i=1}^{n} f_x(x_i).\ |dx_i|$$

The above expression can equivalently be written as,

$$f_Y(y) = \sum_{i=1}^{n} f_X(x_i).\left|\frac{dx_i}{dy}\right|$$

$$= \sum_{i=1}^{n} f_X(x_i).\left|\frac{dy}{dx_i}\right|^{-1}$$

Let us note that, at the i-th root of y = g(x), $\frac{dy}{dx_i} = g'(x_i).$ = value of the derivative of g(x) with respect to 'x' , evaluated at $x = x_i$.

Using the above convenient notation, we finally get,

$$f_Y(y) = \sum_{i=1}^{n} f_X(x_i) / \left|g'(x_i)\right|, \qquad 2.6.3$$

Here, x_i is the i-th real root of y = g(x) and $g'(x_i) \neq 0$. If, for a given y, y = g(x) has no real root, then $f_Y(y) = 0$ as X being a random variable and 'x' being real, it can not take imaginary values with non-zero probability.

Let us take up a small example before concluding this lesson.

■

Example #2.6.6

Let X be a random variable known to follow uniform distribution between -π and +π. So, the mean of X is 0 and its probability density function [pdf] is:

$$f_X(x) = \begin{cases} \frac{1}{2\pi}, -\pi < x \leq \pi \\ 0, otherwise \end{cases}$$

Now consider a new random variable Y which is a function of X and the functional relationship is, Y = g(X) = sinX.

So, we can write, y = g(x) = sin x . Further, one can easily observe that, the pdf of Y exists for $-1.0 \leq y < 1.0$.

Let us first consider the interval $0 \leq y < 1.0$:

The roots of y – sin x = 0 for y >0 are, $x_1 = \sin^{-1}(y)$ and $x_2 = \pi - \sin^{-1}(y)$.

$$Further,\ \frac{dg(x)}{dx} = \cos x \quad while$$

$$\left.\frac{dg(x)}{dx}\right|_{x=x_1} = \cos\left(\sin^{-1} y\right) \quad and$$

$$\left.\frac{dg(x)}{dx}\right|_{x=x_2} = \cos\left(\pi - \sin^{-1} y\right)$$

$$= \cos\pi.\cos\left(\sin^{-1} y\right) + \sin\pi.\sin\left(\sin^{-1} y\right) = -\cos\left(\sin^{-1} y\right)$$

We see that,

$$\left|\frac{dg(x)}{dx}\right|_{x_1} \mp \left|\frac{dg(x)}{dx}\right|_{x_2} = \sqrt{1-y^2}$$

$$\therefore f_Y(y) = \frac{f_X(x_1)}{\left|g'(x_1)\right|} + \frac{f_X(x_2)}{\left|g'(x_2)\right|}$$

$$= \frac{f_X(\sin^{-1} y)}{\sqrt{1-y^2}} + \frac{f_X(\pi - \sin^{-1} y)}{\sqrt{1-y^2}}$$

$$= \frac{1}{2\pi} \times \frac{2}{\sqrt{1-y^2}} = \frac{1}{\pi}.\frac{1}{\sqrt{1-y^2}}, \qquad 0 \le y < 1$$

Following similar procedure for the range $-1 \le y < 0$, it can ultimately be shown that,

$$f_Y(y) = \begin{cases} \frac{1}{\pi}.\frac{1}{\sqrt{1-y^2}}, & |y| < 1 \\ 0, & \textit{otherwise} \end{cases}$$

Problems

Q2.6.1) Let, $y=2x^2 + 3x+1$. If pdf is x is $f_X(x)$, determine an expression for pdf of y.

Q2.6.2) Sketch the pdf of y of problem 2.6.1, if X has u form distribution between -1 and +1.

Module 2

Random Processes

Lesson

6

Some useful Distributions

After reading this lesson, you will learn about

- ***Uniform Distribution***
- ***Binomial Distribution***
- ***Poison Distribution***
- ***Gaussian Distribution***
- ***Central Limit Theorem***
- ***Generation of Gaussian distributed random numbers using computer***
- ***Error function***

Uniform (or Rectangular) Distribution

The pdf f(x) and the cdf F(x) are defined below:

f(x) = 0 for x < a and
for x > (a + b).

However, let, f(x) = 1/b for a < x < (a + b). It is easy to see that,

$$\int_{-\infty}^{+\infty} f(x)\,dx = \int_{a}^{a+b} \left(\frac{1}{b}\right) dx = 1 \qquad 2.7.1$$

The cdf $F(x) = \int_{-\infty}^{x} f(x)\,dx = \begin{cases} 0 & \text{for } x < a; \\ \dfrac{x-a}{b} & \text{for } a < x < (a+b); \\ 1 & \text{for } x > (a+b); \end{cases}$ 2.7.2

Binomial Distribution

This distribution is associated with discrete random variables. Let 'p' is the probability of an event (say, 'S', denoting success) in a statistical experiment. Then, the probability that this event does not occur (i.e. failure or 'F' occurs) is (1 – p) and, for convenience, let, q = 1 – p, i.e. p + q = 1. Now, let this experiment be conducted repeatedly (without any fatigue or biasness or partiality) 'n' times. Binomial distribution tells us the probability of exactly 'x' successes in 'n' trials of the experiment (x ≤ n). The corresponding binomial probability distribution is:

$$f(x) = \binom{n}{x} p^x q^{n-x} = \left[\frac{n!}{x!(n-x)!}\right] p^x (1-p)^{n-x} \qquad 2.7.3$$

Note that, $\sum_{x=0}^{n} f(x) = \sum_{x=0}^{n} \binom{n}{x} p^x q^{n-x} = 1$

The following general binomial expression and its derivative with respect to a dummy variable 'τ'are useful in verifying the above comment and also for finding expectations of 'S':

$$(p\tau + q)^n = \sum_{x=0}^{n} \binom{n}{x} (p\tau)^x q^{n-x}$$

Differentiating this expression once w.r.t. 'τ', we get,

$$n(p\tau + q)^{n-1} p = \sum_{x=0}^{n} \binom{n}{x} p^x q^{n-x} x\tau^{x-1}$$

The mean number of successes, $E(s) = \sum_{x=0}^{n} x f(x) = \sum_{x=0}^{n} \binom{n}{x} p^x q^{n-x} x$

A simple form of the last term can be obtained by putting τ = 1 in the previous expression resulting in a fairly easy-to-remember mean of the random variable:
The mean number of successes, E (S) = np

Following a similar approach it can be shown that, the variance of the number of successes,

σ^2 = npq.

An approximation

The following approximation of the binomial distribution of 2.7.3, known as Laplace approximation, is good to use when 'n' is large, i.e. n → ∞:

$$f(x) = \{(2\pi npq)^{-1/2}\}.\exp[-\{(x-np)^2\}/(2npq)] \qquad 2.7.4$$

Poisson Distribution

A formal discussion on Poisson distribution usually follows a description of binomial distribution because the Poisson distribution can be approximately viewed as a limiting case of binomial distribution.

The approximate Poisson distribution, $f_p(x)$ is:

$$f_p(x) = \left[\{.\lambda^x\}/x!\right].e^{-\lambda} \qquad 2.7.5$$

'λ' in the above expression is the mean of the distribution. A special feature of this distribution is that the variance of this distribution is approximately the same as its mean (λ).

Poisson distribution plays an important role in traffic modeling and analysis in data networks.

Gaussian (Normal) Distribution:

This is a commonly applied distribution often exhibited by continuous random variables in nature. A relevant example is the weak electrical noise (voltage or equivalently current) generated because of random (Brownian) movement of carriers in electronic devices and components (such as resistor, diode, transistor etc.).

A Gaussian or, normally distributed random variable X has the following probability density function:

$$f_N(x) = [\ 1/\{\sigma.\ (2\pi)^{1/2}\}].\ \exp[\ -\ \{(x - m)^2\}/\ (2\sigma^2)\ ] \qquad 2.7.6$$

$m = E[X]$: Mean or expected value of the distribution and

$\sigma^2 = E[\{X - E[X]\}^2]$: the variance of the distribution.

Fig. 2.7.1 shows two Gaussian pdf curves with means 0.0 and 1.5 but same variance 1.0. The particular curve with $m = 0.0$ and $\sigma^2 = 1.0$ is known as the normalized Gaussian pdf. It may be noted that a pdf curve is symmetric about its mean value. The mean may be positive or negative. Further, a change in the mean of a Gaussian pdf curve only shifts the curve horizontally without any change in shape of the curve. A smaller variance, however, increases the sharpness of the peak value of the pdf which always occurs at the average value of the random variable. This explains the significance of 'variance' of a distribution. A smaller variance means that a random variable mostly assumes values close to its expected or mean value.

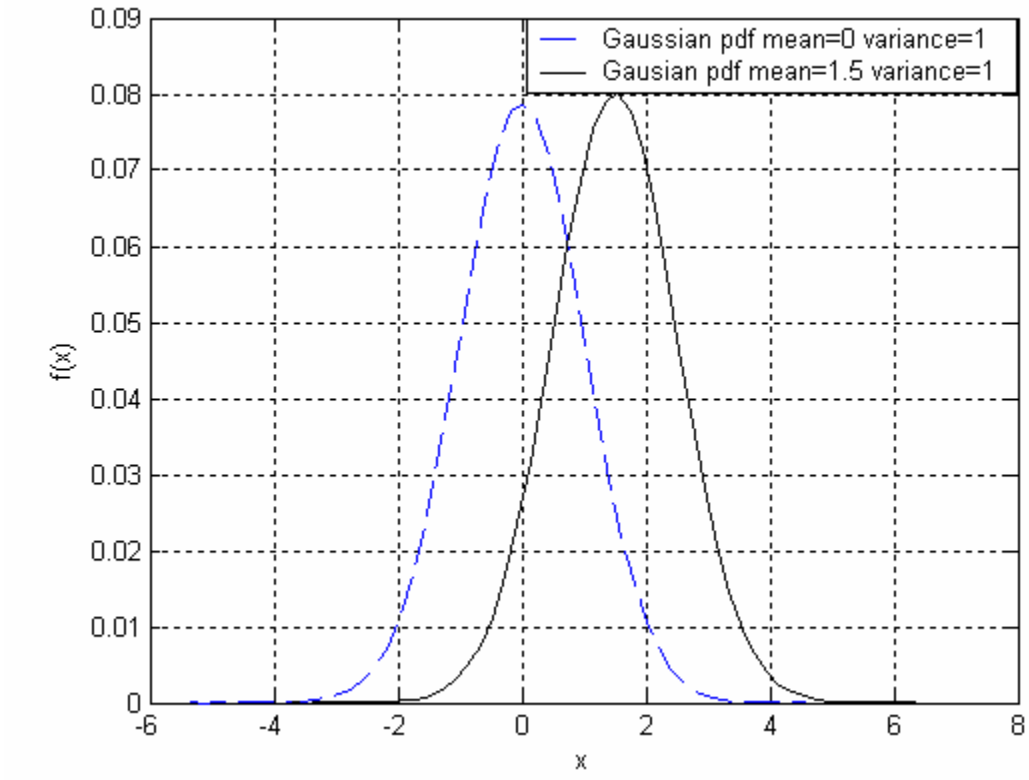


Fig.2.7.1. *Gaussian pdf with mean = 0.0 and variance = 1.0 and Gaussian pdf with mean = 1.5 and variance = 1.0*

It can be shown with some approximation that about 68% of the values (assumed by a Gaussian distributed random variable over a large number of unbiased trials) lie within $\pm \sigma$ around the mean value and about 95% of the values lie within $\pm 2\sigma$ around the mean value.

The sum of mutually independent random variables, each of which is Gaussian distributed, is a Gaussian distributed random variable.

Central Limit Theorem

Let X_1, X2, X_3, ……, X_N denote N mutually independent random variables whose individual distributions are not known and they are not necessarily Gaussian distributed. If m_i , i = 1, 2, ….,N, indicates the mean of the i-th random variable and σ_i , i = 1, 2, ….,N, indicates the variance of the i-th random variable, the central limit theorem, under a few subtle conditions, results in a very powerful and significant inference. The theorem establishes that the sum of the N random variables (say, X) is a random variable which tends to follow a Gaussian (or, normal) distribution as $N \rightarrow \infty$. Further, a) the mean of the sum random variable X is the sum of the mean values of the constituent random variables and b) the variance of the sum random variable X is the sum of the variance values of the constituent random variables X_1, X2, X_3, ……, X_N.

The Central Limit Theorem is very useful in modeling and analyzing several situations in the study of electrical communications. However, one necessary condition to look for before invoking Central Limit theorem is that no single random variable should have significant contribution to the sum random variable.

Generation of Gaussian distributed random numbers using computers

Gaussian distributed random numbers can be generated by generating random numbers having uniform distribution. A simple (but not very good always) method of doing this is by applying central limit theorem. A set of 'n' (usually $n \geq 12$) independent uniformly distributed random numbers are generated and they are summed up and scaled appropriately to result in a Gaussian distributed random variable.

A faster and more precise way for generating Gaussian random variables is known as the Box-Müller method. Only two uniform distributed random variables say, X_1 and X_2, ensure generation of two (almost) independent and identically distributed Gaussian random variables (N_1 and N_2). If x_1 and x_2 are two uncorrelated values assigned to X_1 and X_2 respectively, the two independent Gaussian distributed values n_1 and n_2 are obtained from the following relations:

$n_1 = \sqrt{[\ -2.(\ln x_1).\cos 2\pi x_2\ ]}$

$n_2 = \sqrt{[\ -2.(\ln x_2).\sin 2\pi x_1\ ]}$

2.7.7

Error Functions

A few frequently used functions that are closely related to Gaussian distribution are summarized below. The reader may refer back to this sub-section as necessary.

The error function, erf(x) is defined as:

$$erf\,(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-z^2}\, dz \qquad 2.7.8$$

It may be noted that the above definite integral is not easy to calculate and tables containing approximate values for the argument 'x' are readily available. A few properties of the error function erf(x) are noted below:

a) erf(-x) = - erf(x) [Symmetry Property]

b) as x → + ∞, erf(x) → 1, since $\frac{2}{\sqrt{\pi}} \int_0^\infty e^{-z^2}\, dz = 1$

c) if 'X' is a Gaussian random variable with mean m_x and variance σ_x^2, the probability that X lies in the interval ($m_x - a$, $m_x + a$) is:

$$P\left(m_x - a < X \le m_x + a\right) = erf\left(\frac{a}{\sqrt{2}\sigma_x}\right) = \frac{2}{\sqrt{\pi}} \int_0^{a/\sqrt{2}\sigma_x} e^{-z^2}\, dz \qquad 2.7.9$$

The complementary error function is directly related to the error function erf(x):

$$erfc(u) = \frac{2}{\sqrt{\pi}} \int_u^\infty \exp\left(-z^2\right) dz = \frac{2}{\sqrt{\pi}} \int_u^\infty e^{-z^2}\, dz \;=\; erfc(u) = 1 - erf\,(u) \qquad 2.7.10$$

A useful bound on the complementary error function erfc(v):

For large positive v,

$$\frac{\exp\left(-v^2\right)}{\sqrt{\pi}\cdot v}\left(1 - \frac{1}{2v^2}\right) < erfc(v) < \frac{\exp\left(-v^2\right)}{\sqrt{\pi}\cdot v} \qquad 2.7.11$$

The Q- function or the Marcum function (**Fig.2.7.2**) is another frequently occurring function. Considering a standardized Gaussian random variable X with $m_x = 0$ & $\sigma_x^2 = 1$, the probability that an observed value of x will be greater than v is given by the following Q – function:

$$Q(v) = \frac{1}{\sqrt{2\pi}} \int_v^\infty \exp\left(-\frac{x^2}{2}\right) dx \qquad 2.7.12$$

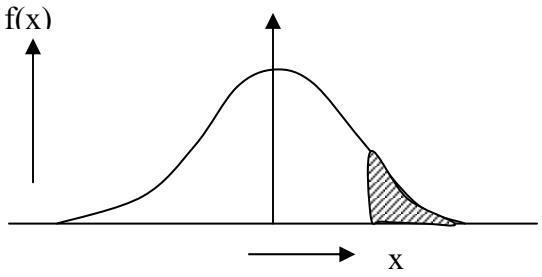


Fig.2.7.2 *The shaded area is measured by the Q-function*

We may note that,

$$Q(v) = \frac{1}{2} erfc\left(\frac{v}{\sqrt{2}}\right) \qquad 2.7.13$$

And conversely, putting $u = v/\sqrt{2}$, $\quad erfc(u) = 2Q\left(\sqrt{2}u\right)$ 2.7.14

Problems

Q2.7.1) If a fair coin is tossed ten times, determine the probability that exactly two heads occur.

Q2.7.2) Mention an example/situation where Poisson distribution may be applied.

Q2.7.3) Explain why Gaussian distribution is also called as Normal distribution.

Module 2

Random Processes

Lesson

8

Stochastic Processes

After reading this lesson, you will learn about

- ***Stochastic Processes***
- ***Statistical Average***
- ***Wide sense stationary process***
- ***Complex valued stochastic process***
- ***Power Density Spectrum of a stochastic process***

- Many natural and man-made phenomena are random and are functions of time.
  e.g., i) Fluctuation of air temperature or pressure, ii) Thermal noise that are generated due to Brownian motion of carriers in conductors and semiconductors and iii) Speech signal.

- All these are examples of Stochastic Processes.

- An instantaneous sample value of a stochastic process is a random variable.

- A stochastic process, some time denoted by X(t), may be defined as an ensemble of (time) sample functions.

  Ex: Consider several identical sources each of which is generating thermal noise.

- Let us define $X_{ti}=X(t=t_i)$, i=1,2,….., n (n is arbitrarily chosen) as the random variables obtained by sampling the stochastic process X(t) for any set of time instants $t_1<t_2<t_3…..<t_n$. So, these 'n' random variables can be characterized by their joint pdf, i.e. $p(x_{t1}, x_{t2}, x_{t3},….., x_{tn})$.

- Now, let us consider another set of random variables $X(t_i+\tau)$, generated from same stochastic process X(t) with an arbitrary time shift $''\tau'$

  $$X_{t_1+\tau} = X(t=t_1+\tau),.....X_{t_i+\tau} = X(t=t_i+\tau)\ldots\ldots$$

  and with a joint pdf $p(x_{t_1+\tau}, x_{t_2+\tau},....., x_{t_n+\tau})$.

  In general, these two joint pdf-s may not be same. However, if they are same for all τ and any 'n', then the stochastic process X(t) is said to be stationary in the strict sense. i.e.,

  $$p(x_{t1}, x_{t2},....., x_{tn}) = p(x_{t_1+\tau}, x_{t_2+\tau},....., x_{t_n+\tau}) \text{ for all } '\tau' \text{ and all 'n'.}$$

  Strict sense stationarity means that the statistics of a stochastic process is invariant to any translation of the time axis. Finding a quick example of a physical random process which is stationary in the strict sense may not be an easy task.

- If the joint pdf's are different for any $'\tau'$or 'n' the stochastic process is non-stationary.

Statistical Averages

- Like random variables, we can define statistical averages or ensemble averages for a stochastic process.

  Let X(t) be a random process and $X_{ti} = X(t = t_i)$
  Then, the n-th moment of X_{ti} is:

$$E[X_{ti}^{\,n}] = \int_{-\infty}^{+\infty} x_{ti}^{\,n}.p(x_{ti})dx_{ti} \qquad 2.8.1$$

- For a stationary process, $p(x_{ti+\tau}) = p(x_{ti})$ for all τ and hence the n-th moment is independent of time.

- Let us consider two random variables $X_{ti} = X(t_i), i = 1, 2$
  The correlation between X_{t1} & X_{t2} is measured by their joint moment:

$$E[X_{t1}X_{t2}] = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} x_{t1}.x_{t2}\,p(x_{t1},x_{t2})dx_{t1}dx_{t2} \qquad 2.8.2$$

  This joint moment is also known as the autocorrelation function, $\Phi(t_1,t_2)$ of the stochastic process X(t). In general, $\Phi(t_1,t_2)$ is dependent on t_1 and t_2.

- However, for a strict-sense stationary process,

$$p(x_{t1},x_{t2}) = p(x_{t1}+T,\ x_{t2}+T),$$

  For any T and this means that autocorrelation function $\Phi(t_1,t_2)$ for a stationary process is dependent on $(t_2 - t_1)$, rather than on t_2 and t_1,
  i.e.,

$$E[X_{t1}X_{t2}] = \Phi(t_1,t_2) = \Phi(t_2 - t_1) = \Phi(\tau), \quad \text{say where } \tau = t_2 - t_1 \qquad 2.8.3$$

  It may be seen that, $\Phi(-\tau) = \Phi(\tau)$ and hence, $\Phi(\tau)$ is an even function.

- Note that hardly any physical stochastic process can be described as stationary in the strict sense, while many of such processes obey the following conditions:
  a) Mean value, i.e., E[X] of the process is independent of time and,
  b) $\Phi(t_1,t_2) = \Phi(t_1 - t_2)$ over the time domain of interest.

  Such stochastic processes are known as *wide sense stationary*. This is a less stringent condition on stationarity.

- The auto-covariance function of a stochastic process is closely related to the autocorrelation function:

$$\mu(t_1,t_2) = E\{[X_{t1} - m(t_1)][X_{t2} - m(t_2)]\}$$
$$= \Phi(t_1 - t_2) - m(t_1).m(t_2) \qquad 2.8.4$$

  Here, $m(t_1) = E[X_{t1}]$. For a wide sense stationary process,

$$\mu(t_1 - t_2) = \mu(t_2 - t_1) = \mu(\tau) = \Phi(\tau) - m^2 \qquad 2.8.5$$

- It is interesting to note that a Gaussian process is completely specified by its mean and auto covariance functions and hence, if a Gaussian process is wide-sense stationary, it is also strict-sense stationary.

- We will mean wide-sense stationary process when we discuss about a stationary stochastic process.

Averages for joint stochastic processes

Let X(t) & Y(t) denote any two stochastic processes and $X_{ti} = X(t_i)$, i=1,2,….n and $Y_{t'_j} = Y(t'_j)$, j=1,2,……,m are the random variables at $t_1 > t_2 > t_3 \ldots > t_n$ and $t'_1 > t'_2 > \ldots\ldots > t'_m$ [n and m are arbitrary integers].

The two processes are statistically described by their joint pdf:
$p(x_{t1}, x_{t2}, \ldots\ldots x_{tn}, yt'_1, yt'_2, \ldots\ldots yt'_m)$

Now, the cross-correlation function of X(t) and Y(t) is defined as their joint moment:

$$\Phi_{xy}(t_1, t_2) = E(X_{t1}Y_{t2}) = \int_{-\alpha}^{\alpha}\int_{-\alpha}^{\alpha} x_{t1}y_{t2}\Phi(x_{t1}, y_{t2})dx_{t1}dy_{t2} \qquad 2.8.6$$

The cross-covariance is defined as,

$$\mu_{xy}(t_1, t_2) = \Phi_{xy}(t_1, t_2) - m_x(t_1).m_y(t_2) \qquad 2.8.7$$

Here, $m_x(t_1) = E[X_{t1}]$ and $m_y(t_1) = E[Y_{t2}]$. If X and Y are individually and jointly stationary, we have,

$$\Phi_{xy}(t_1, t_2) = \Phi_{xy}(t_1 - t_2)$$
$$\& \qquad 2.8.8$$
$$\mu_{xy}(t_1, t_2) = \mu_{xy}(t_1 - t_2)$$

In particular, $\Phi_{xy}(-\tau) = E(X_{t1}Y_{t1+\tau}) = E(X_{t'_1-\tau}.Y_{t'_1}) = \Phi_{yx}(\tau)$

Now, two stochastic processes are said to be statistically independent if and only if,

$$p(x_{t1}, x_{t2}, \ldots\ldots x_{tn}, y_{t'1}, y_{t'2}\ldots\ldots, y_{t'm})$$
$$= p(x_{t1}, x_{t2}, \ldots\ldots t_n).p(y_{t'1}, y_{t'2}\ldots\ldots, y_{t'm}) \qquad 2.8.9$$
for all t_i, t'_j, n and m

Two stochastic processes are said to be uncorrelated if,
$\Phi_{xy}(t_1, t_2) = E[X_{t1}].E[Y_{t2}]$. This implies, $\mu_{xy}(t_1, t_2) = 0$ 2.8.10
i.e., *the processes have zero cross-covariance.*

Complex valued stochastic process

Let, Z(t) = X(t) + jY(t), where X(t) and Y(t) are individual stochastic processes.
One can now define joint pdf of Z_{ti} = Z(t =i), i=1,2,……,n, as
$p(x_{t1}, x_{t2}, \ldots\ldots x_{tn}, y_{t1}, y_{t2,} \ldots\ldots y_{tn})$

Now the auto correlation function $\tilde{\Phi}_{zz}(t_1, t_2)$ of the complex process z(t) is defined as,

$$\tilde{\Phi}_{zz}(t_1, t_2) = \frac{1}{2}.E[Z_{t1}.Z^{*}_{t2}] = \frac{1}{2}.E[(x_{t1} + jy_{t1})(x_{t2} + jy_{t2})]$$

$$= \frac{1}{2}.[\Phi_{xx}(t_1,t_2) + \Phi_{yy}(t_1,t_2) + j\{\Phi_{yx}(t_1,t_2) - \Phi_{xy}(t_1,t_2)\}] \qquad 2.8.11$$

Here, $\Phi_{xx}(t_1,t_2)$: Auto-correlation of X;
$\Phi_{yy}(t_1,t_2)$: Auto-correlation of Y;
$\Phi_{yx}(t_1,t_2)$ and $\Phi_{xy}(t_1,t_2)$ are cross correlations of X&Y.

For a stationary $\tilde{z}(t)$, $\Phi_{zz}(t_1,t_2) = \Phi_{zz}(\tau)$ and $\Phi^{*}_{zz}(t_1,t_2) = \Phi^{*}_{zz}(\tau) = \Phi_{zz}(-\tau)$.

2.8.12

Power Density Spectrum of a Stochastic Process

A stationary stochastic process is an infinite energy signal and hence its Fourier Transform does not exist. The spectral characteristic of a stochastic process is obtained by computing the Fourier Transform of the auto correlation function.

That is, the distribution of power with frequency is described as:

$$\Phi(f) = \int_{-\alpha}^{\alpha} \Phi(\tau) e^{-j2\pi f\tau} d\tau \qquad 2.8.13$$

The Inverse Fourier Transform relationship is:

$$\Phi(\tau) = \int_{-\infty}^{\infty} \Phi(f) e^{j2\pi f\tau} df \qquad 2.8.14$$

Note that, $\Phi(0) = \int_{-\alpha}^{\alpha} \Phi(f) df = E(X_1^{\,2}) \geq 0$

$\therefore \Phi(0)$ represents the average power of the stochastic signal, which is the area under the $\Phi(f)$ curve.
Hence, $\Phi(f)$ is called the power density spectrum of the stochastic process.

If X(t) is real, then $\Phi(\tau)$ is real and even.
Hence $\Phi(f)$ is real and even.

Problems

Q2.8.1) Define and explain with an example a “wide sense stationary stochastic process”.

Q2.8.2) What is the condition for two stochastic processes to be uncorrelated?

Q2.8.3) How to verify whether two stochastic processes are statistically independent of each other?

Module 2

Random Processes

Lesson 9

Introduction to Statistical Signal Processing

After reading this lesson, you will learn about

- ***Hypotheses testing***
- ***Unbiased estimation based on minimum variance***
- ***Mean Square Error (MSE)***
- ***Crammer - Rao Lower Bound (CRLB)***

As mentioned in Module #1, demodulation of received signal and taking best possible decisions about the transmitted symbols are key operations in a digital communication receiver. We will discuss various modulation and demodulation schemes in subsequent modules (Modules # 4 and #5). However, a considerable amount of generalization is possible at times and hence, it is very useful to have an overview of the underlying principles of statistical signal processing. For example, the process of signal observation and decision making is better appreciated if the reader has an overview of what is known as 'hypothesis testing' in detection theory. In the following, we present a brief treatise on some fundamental concepts and principles of statistical signal processing.

Hypothesis Testing

A hypothesis is a statement of a possible source of an observation. The observation in a statistical hypothesis is a random variable. The process of making a decision from an observation on 'which hypothesis is true (or correct)', is known as hypothesis testing.

Ex.#2.9.1. Suppose it is known that a modulator generates either a pulse 'P_1' or a pulse 'P2' over a time interval of 'T' second and 'r' is received in the corresponding observation interval of 'T' sec. Then, the two hypotheses of interest may be,

H_1: 'P_1' is transmitted.
H_2: 'P_2' is transmitted.

Note that 'P_1 is not transmitted' is equivalent to 'P_2 is transmitted' and vice versa, as it is definitely known that either 'P_1'or'P_2' has been transmitted.

Let us define a parameter (random variable) 'u' which is generated in the demodulator as a response to the received signal 'r', over the observation interval. The parameter 'u' being a function of an underlying random process, is defined as a random variable (single measurement over T) and is called an 'observation' in the context of hypothesis testing. An 'observation' is also known as a 'decision variable' in the jargon of digital communications. The domain (range of values) of 'u' is known as 'observation space'. The relevant hypothesis is, 'making a decision on whether H_1 or H_2 is correct' (upon observing 'u'). The observation space in the above example is the one dimensional real number axis. Next, to make decisions, the whole

observation space is divided in appropriate 'decision regions' so as to associate the possible decisions from an observation 'r' with these regions.

Note that the observation space need not be the same as the range of pulses P_1 or P_2. However, decision regions can always be identified for the purpose of decision-making. If a decision doesn't match with the corresponding true hypothesis, an error is said to have occurred. An important aim of a communication receiver is to clearly identify the decision regions such that the (decision) errors are minimized.

Estimation

There are occasions in the design of digital communications systems when one or more parameters of interest are to be estimated from a set of signal samples. For example, it is very necessary to 'estimate' the frequency (or, instantaneous phase) of a received carrier-modulated signal as closely as possible so that the principle of 'coherent demodulation' may be used and the 'decision' errors can be minimized.

Let us consider a set of N random, discrete-time samples or data points, {x[0], x[1], x[2], …,x[N-1]} which depends (in some way) on a parameter θ which is unknown. The unknown parameter θ is of interest to us. Towards this, we express an 'estimator' $\hat{\theta}$ as a function of the data points:

$$\hat{\theta} = f(x[0], x[1], x[2], \ldots, x[N-1]) \qquad 2.9.1$$

Obviously, the issue is to find a suitable function f(.) such that we can obtain θ from our knowledge of $\hat{\theta}$ as precisely as we should expect. Note that, we stopped short of saying that at the end of our estimation procedure, '$\hat{\theta}$ will be equal to our parameter of interest θ'.

Next, it is important to analyze and mathematically model the available data points which will usually be of finite size. As the sample points are inherently random, it makes sense to model the data set in terms of some family of probability density function [pdf] , parameterized by the parameter θ: p(x[0], x[1], x[2], …,x[N-1] ; θ). It means that the family of pdf is dependent on 'θ' and that different values of θ may give rise to different pdf-s. The distribution should be so chosen that the mathematical analysis for estimation is easier. A Gaussian pdf is a good choice on several occasions. In general, if the pdf of the data depends strongly on θ, the estimation process is likely to be more fruitful.

A classical estimation procedure assumes that the unknown parameter θ is deterministic while the Bayesian approach allows the unknown parameter to be a random variable. In this case, we say that the parameter, being estimated, is a 'realization' of the random variable. If the set of sample values {x[0], x[1], x[2], …,x[N-1]} is represented by a data vector **x,** a joint pdf of the following form is considered before formulating the procedure of estimation:

$p(\mathbf{x}, \theta) = p(\mathbf{x} \mid \theta).\ p(\theta)$ 2.9.2

Here $p(\theta)$ is the prior pdf of θ, the knowledge of which we should have before observing the data and $p(\mathbf{x} \mid \theta)$ is the conditional pdf of **x**, conditioned on our knowledge of θ. With these interpretations, an estimator may now be considered as a rule that assigns a value to θ for each realization of data vector **x** and the 'estimate' of θ is the value of θ for a specific realization of the data vector **x**. An important issue for any estimation is to assess the performance of an estimator. All such assessment is done statistically. As the process of estimation involves multiple computations, sometimes the efficacy of an estimator is decided in a practical application by its associated computational complexity. An 'optimal' estimator may need excessive computation while the performance of a suboptimal estimator may be reasonably acceptable.

An Unbiased Estimation based on Minimum Variance:

Let us try to follow the principle of a classical estimation approach, known as unbiased estimation. We expect the estimator to result in the true value of the unknown parameter on an average. The best estimator in this class will have minimum variance in terms of estimation error ($\hat{\theta}$-θ) . Suppose we know that the unknown parameter θ is to lie within the interval $\theta_{min} < \theta < \theta_{max}$. The estimate is said to be unbiased if $E[\hat{\theta}] = \theta$ in the interval $\theta_{min} < \theta < \theta_{max}$. The criterion for optimal estimation that is intuitive and easy to conceive is the 'mean square error (MSE)', defined as below:

$$MSE(\hat{\theta}) = E[(\hat{\theta} - \theta)^2] \quad 2.9.3$$

However, from practical considerations, this criterion is not all good as it may not be easy to formulate a good estimator (such as 'minimum MSE') that can be directly expressed in terms of the data points. Usually, the MSE results in a desired variance term and an undesired 'bias' term, which makes the estimator dependent on the unknown parameter (θ). So, summarily, the intuitive 'mean square error (MSE)' approach does not naturally lead to an optimum estimator unless the bias is removed to formulate an 'unbiased' estimator.

A good approach, wherever applicable, is to constrain the bias term to zero and determine an estimator that will minimize the variance. Such an estimator is known as a 'minimum variance unbiased estimator'.

The next relevant issue in classical parameter estimation is to find the estimator. Unfortunately, there is no single prescription for achieving such an optimal estimator for all occasions. However, several powerful approaches are available and one has to select an appropriate method based on the situation at hand. One such approach is to determine the Crammer-Rao lower bound (CRLB) and to check whether an estimator at hand satisfied this bound. The CRLB results in a bound over the permissible range

of the unknown parameter θ such that the variance of any unbiased estimator will be equal or greater than this bound.

Further, if an estimator exists whose variance equals the CRLB over the complete range of the unknown parameter θ, then that estimator is definitely a minimum variance unbiased estimator. The theory of CRLB can be used with reasonable ease to see for an application whether an estimator exists which satisfies the bound. This is important in modeling and performance evaluation of several functional modules in a digital receiver.

The CRLB theorem can be stated in multiple ways and we choose the simple form which is applicable when the unknown parameter is a scalar.

Theorem on Crammer- Rao Lower Bound

The variance of any unbiased estimator $\hat{\theta}$ will satisfy the following inequality provided the assumptions stated below are valid:

$$\text{var}(\hat{\theta}) \geq 1 / \{ -E[(\partial^2 \ln p(\mathbf{x};\theta))/\partial\theta^2] \} = 1/ \{E[(\partial \ln p(\mathbf{x};\theta)/\partial\theta)^2]\} \quad 2.9.4$$

Associated assumptions:

a) all statistical expectation operations are with respect to the pdf $p(\mathbf{x};\theta)$,
b) the pdf $p(\mathbf{x};\theta)$ also satisfies the following 'regularity' condition for all θ:

$$E[(\partial \ln p(\mathbf{x};\theta))/\partial\theta] = \int \{ (\partial \ln p(\mathbf{x};\theta))/\partial\theta \}. p(\mathbf{x};\theta) d\mathbf{x} = 0 \quad 2.9.5$$

Further, an unbiased estimator, achieving the above bound may be found only if the following equality holds for some functions g(.) and I(.):

$$\partial \ln p(\mathbf{x};\theta))/\partial\theta = I(\theta).\{g(\mathbf{x}) - \theta \} \quad 2.9.6$$

The unbiased estimator is given by $\hat{\theta} = g(\mathbf{x})$ and the resulting minimum variance of $\hat{\theta}$ is the reciprocal of I(θ), i.e. 1/I(θ).

It is interesting to note that, $I(\theta) = -E[(\partial^2 \ln p(\mathbf{x};\theta))/\partial\theta^2] = E[(\partial \ln p(\mathbf{x};\theta)/\partial\theta)^2]$ and is known as Fisher information, associated with the data vector '**x**'.

CRLB for signals in White Gaussian Noise

Let us denote time samples of a deterministic signal, which is some function of the unknown parameterθ, as y[n;θ]. Let us assume that we have received the samples after corruption by white Gaussian noise, denoted by ω[n], as:

$$x[n] = y[n;\theta] + \omega[n], \qquad n = 1, 2, \ldots, N$$

We wish to determine an expression for the Crammer-Rao Lower Bound with the above knowledge. Note that we can express ω[n] as $\omega[n] = x[n] - y[n;\theta]$.

As ω[n] is known to follow Gaussian probability distribution, we may write,

$$p(\mathbf{x};\theta) = \frac{1}{(2\pi\sigma^2)^{\frac{N}{2}}}\exp\left[\frac{-1}{2\sigma^2}\sum_{n=1}^{N}(x[n]-y[n;\theta])^2\right] \quad 2.9.7$$

Partial derivative of $p(\mathbf{x};\theta)$ with respect to θ gives:

$$\frac{\partial \ln p(x;\theta)}{\partial\theta} = \frac{1}{\sigma^2}\cdot\sum_{n=1}^{N}(x[n]-y[n;\theta])\frac{\partial y[n;\theta]}{\partial\theta} \quad 2.9.8$$

Similarly, the second derivative leads to the following expression:

$$\frac{\partial^2 \ln p(x;\theta)}{\partial\theta^2} = \frac{1}{\sigma^2}\cdot\sum_{n=1}^{N}\{(x[n]-y[n;\theta])\frac{\partial^2 y[n;\theta]}{\partial\theta^2} - (\frac{\partial y[n;\theta]}{\partial\theta})^2\} \quad 2.9.9$$

Now, carrying out the expectation operation gives,

$$E[\frac{\partial^2 \ln p(x;\theta)}{\partial\theta^2}] = -\frac{1}{\sigma^2}\cdot\sum_{n=1}^{N}(\frac{\partial y[n;\theta]}{\partial\theta})^2 \quad 2.9.10$$

Now, following the CRLB theorem, we see that the bound is given by the following inequality:

$$\text{var}(\hat{\theta}) \geq \frac{\sigma^2}{\sum_{n=1}^{N}(\frac{\partial y[n;\theta]}{\partial\theta})^2} \quad 2.9.11$$

Problems

Q2.9.1) Justify why “Hypotheses testing” may be useful in the study of digital communications.

Q2.9.2) What is MSE? Explain its significance.

Module 3

Quantization and Coding

Lesson 10

Quantization and Preprocessing

After reading this lesson, you will learn about

- ***Need for preprocessing before quantization;***

Introduction

In this module we shall discuss about a few aspects of analog to digital (A/D) conversion as relevant for the purpose of coding, multiplexing and transmission. The basic principles of analog to digital conversion will not be discussed. Subsequently we will discuss about several lossy coding and compression techniques such as the pulse code modulation (PCM), differential pulse code modulation (DPCM), and delta modulation (DM). The example of telephone grade speech signal having a narrow bandwidth of 3.1 kHz (from 300 Hz to 3.4 kHz) will be used extensively in this module.

Need for Preprocessing Before Digital Conversion

It is easy to appreciate that the electrical equivalent of human voice is summarily a random signal, **Fig.3.10.1**. It is also well known that the bandwidth of an audible signal (voice, music etc.) is less than 20 KHz (typical frequency range is between 20 Hz and 20KHz). Interestingly, the typical bandwidth of about 20 KHz is not considered for designing a telephone communication system. Most of the voice signal energy is limited within 3.4 KHz. While a small amount of energy beyond 3.4 KHz adds to the quality of voice, the two important features of a) message intelligibility and b) speaker recognition are retained when a voice signal is band limited to 3.4 KHz. This band limited voice signal is commonly referred as 'telephone grade speech signal'.

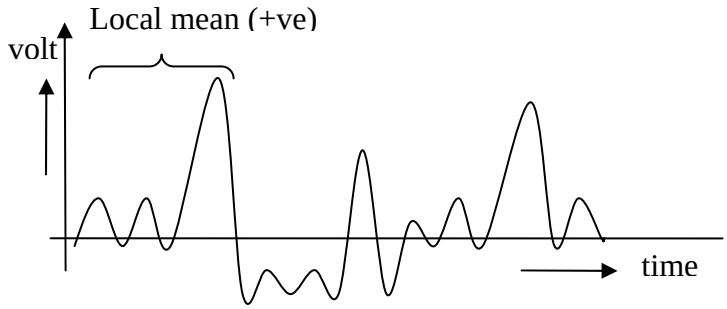


Fig. 3.10.1 *Sketch of random speech signal vs. time*

A very popular ITU-T (International Telecommunication Union) standard specifies the bandwidth of telephone grade speech signal between 300 Hz and 3.4 kHz. The lower cut off frequency of 300 Hz has been chosen instead of 20 Hz for multiple practical reasons. The power line frequency (50Hz) is avoided. Further the physical size and cost of signal processing elements such as transformer and capacitors are also suitable for the chosen lower cut-off frequency of 300 Hz. A very important purpose of squeezing the bandwidth is to allow a large number of speakers to communicate simultaneously through a telephone network while sharing costly trunk lines using the

principle of multiplexing. A standard rate of sampling for telephone grade speech signal of one speaker is 8-Kilo samples/ sec (Ksps).

Usually, an A/D converter quantizes an input signal properly if the signal is within a specified range. As speech is also a random signal, there is always a possibility that the amplitude of speech signal at the input of a quantizer goes beyond this range. If no protection is taken for this problem and if the probability of such event is not negligible, even a good quantizer will lead to unacceptably distorted version of the signal. A possible remedy of this problem is to (a) study and assess the variance of random speech signal amplitude and (b) to adjust the signal variance within a reasonable limit. This is often ensured by using a variance estimation unit and a variable gain amplifier, **Fig. 3.10.2**. Another preprocessing which may be necessary is of DC adjustment of the input speech signal. To explain this point, let us assume that the input range of a quantizer is '±V' volts. This quantizer expects an input signal whose DC (i.e average) value is zero. However if the input signal has an unwanted dc bias ($\overline{x}$ in **Fig 3.10.2**) this also should be removed. For precise quantization, a mean estimation unit can be used to estimate a local mean and subtract it from the input signal. If both the processes of mean removal and variance normalization are to be adopted, the mean of the signal should be adjusted first.

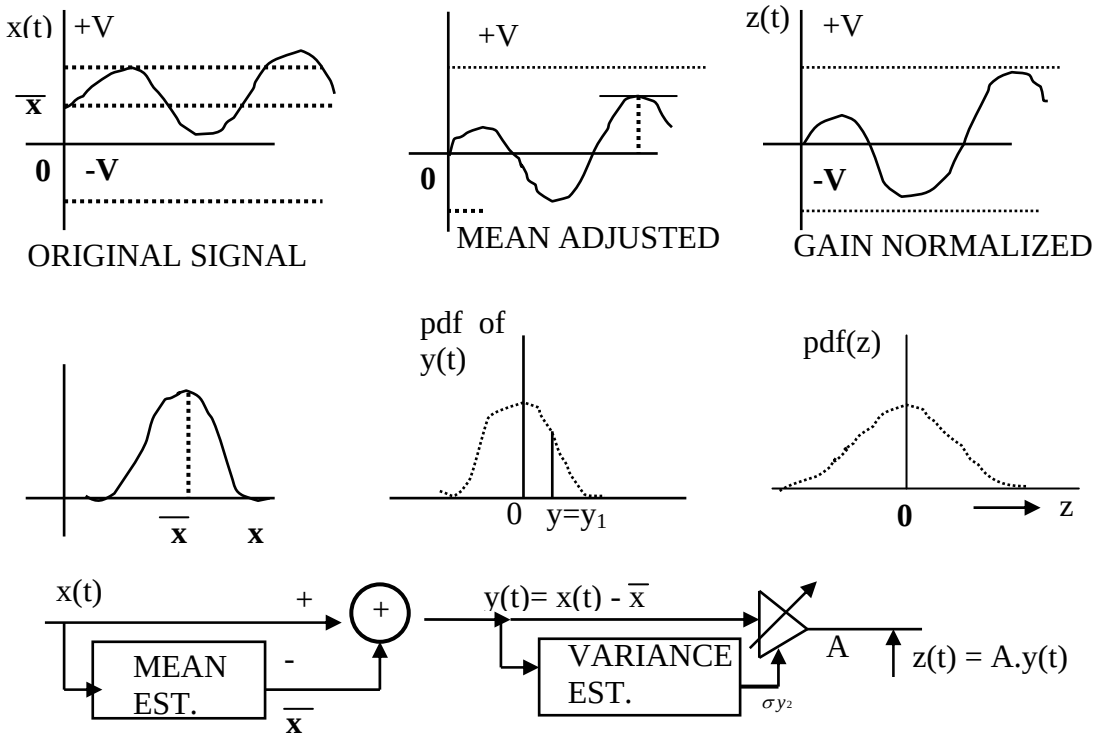


Fig. 3.10.2 *Scheme for mean removal and variance normalization*

Module 3

Quantization and Coding

Version 2, ECE IIT, Kharagpur

Lesson 11

Pulse Code Modulation

Version 2, ECE IIT, Kharagpur

After reading this lesson, you will learn about

- ***Principle of Pulse Code Modulation;***
- ***Signal to Quantization Noise Ratio for uniform quantizaton;***

A schematic diagram for Pulse Code Modulation is shown in **Fig 3.11.1**. The analog voice input is assumed to have zero mean and suitable variance such that the signal samples at the input of A/D converter lie satisfactorily within the permitted single range. As discussed earlier, the signal is band limited to 3.4 KHz by the low pass filter.

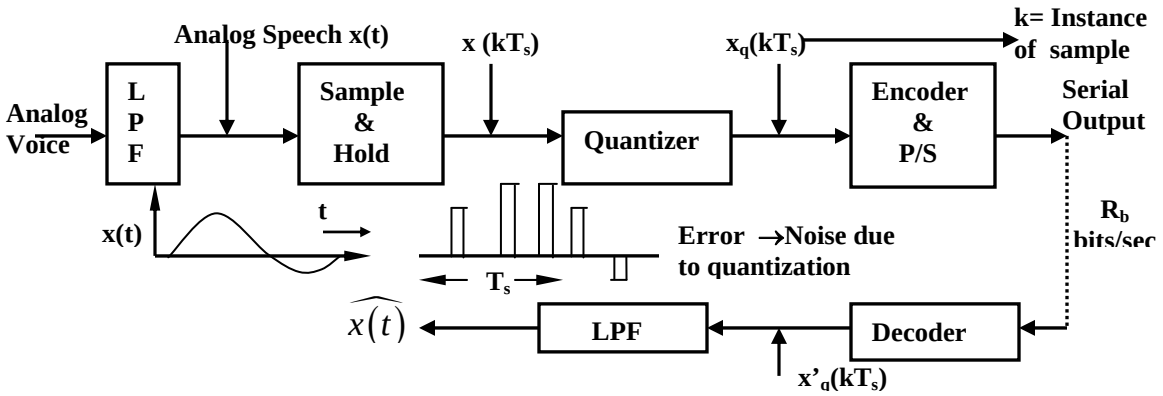


Fig. 3.11.1 *Schematic diagram of a PCM coder – decoder*

Let x (t) denote the filtered telephone-grade speech signal to be coded. The process of analog to digital conversion primarily involves three operations: (a) Sampling of x (t), (b) Quantization (i.e. approximation) of the discrete time samples, x (kT_s) and (c) Suitable encoding of the quantized time samples x_q (kT_s). T_s indicates the sampling interval where $R_s = 1/T_s$ is the sampling rate (samples /sec). A standard sampling rate for speech signal, band limited to 3.4 kHz, is 8 Kilo-samples per second (T_s = 125μ sec), thus, obeying Nyquist's sampling theorem. We assume instantaneous sampling for our discussion. The encoder in **Fig 3.11.1** generates a group of bits representing one quantized sample. A parallel–to–serial (P/S) converter is optionally used if a serial bit stream is desired at the output of the PCM coder. The PCM coded bit stream may be taken for further digital signal processing and modulation for the purpose of transmission.

The PCM decoder at the receiver expects a serial or parallel bit-stream at its input so that it can decode the respective groups of bits (as per the encoding operation) to generate quantized sample sequence [x'_q (kT_s)]. Following Nyquist's sampling theorem for band limited signals, the low pass filter produces a close replica $\hat{x}(t)$ of the original speech signal x (t).

If we consider the process of sampling to be ideal (i.e. instantaneous sampling) and if we assume that the same bit-stream as generated by PCM encoder is available at PCM decoder, we should still expect $\hat{x}(t)$ to be somewhat different from x (t). This is

solely because of the process of quantization. As indicated, quantization is an approximation process and thus, causes some distortion in the reconstructed analog signal. We say that quantization contributes to "noise". The issue of quantization noise, its characterization and techniques for restricting it within an acceptable level are of importance in the design of high quality signal coding and transmission system. We focus a bit more on a performance metric called SQNR (Signal to Quantization Noise power Ratio) for a PCM codec. For simplicity, we consider uniform quantization process. The input-output characteristic for a uniform quantizer is shown in **Fig 3.11.2(a).** The input signal range (± V) of the quantizer has been divided in eight equal intervals. The width of each interval, δ, is known as the step size. While the amplitude of a time sample x (kT_s) may be any real number between +V and –V, the quantizer presents only one of the allowed eight values (±δ, ±3δ/2, …) depending on the proximity of x (kT_s) to these levels.

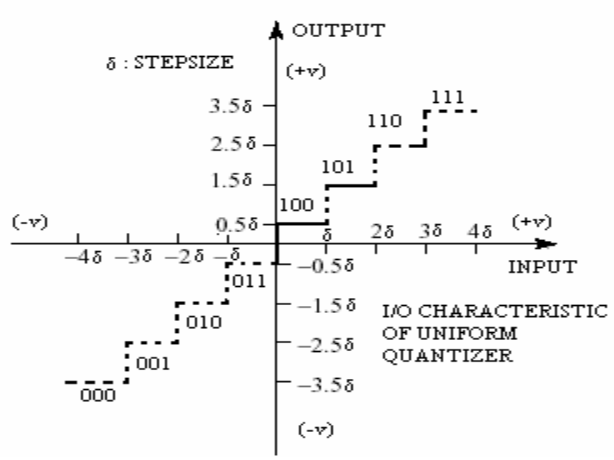


Fig 3.11.2(a) *Linear or uniform quantizer*

The quantizer of **Fig 3.11.2(a)** is known as "mid-riser" type. For such a mid-riser quantizer, a slightly positive and a slightly negative values of the input signal will have different levels at output. This may be a problem when the speech signal is not present but small noise is present at the input of the quantizer. To avoid such a random fluctuation at the output of the quantizer, the "mid-tread" type uniform quantizer [**Fig 3.11.2(b)**] may be used.

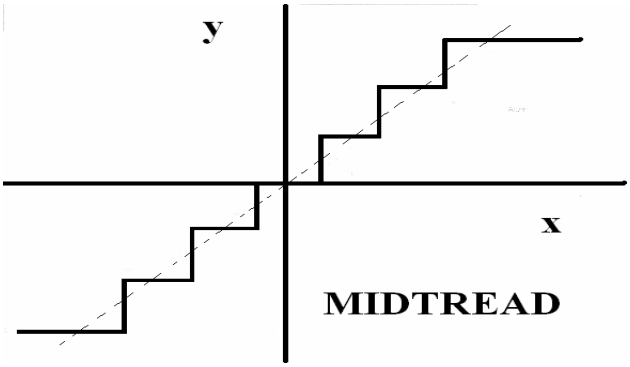


Fig 3.11.2(b) *Mid-tread type uniform quantizer characteristics*

SQNR for uniform quantizer

In **Fig.3.11.1** $x(kT_s)$ represents a discrete time ($t = kT_s$) continuous amplitude sample of x(t) and $x_q(kT_s)$ represents the corresponding quantized discrete amplitude value. Let e_k represents the error in quantization of the k^{th} sample i.e.

$$e_k = x_q(kT_s) - x(kT_s) \qquad 3.11.1$$

Let,

M = Number of permissible levels at the quantizer output.

N = Number of bits used to represent each sample.

±V = Permissible range of the input signal x (t).

Hence,

$M=2^N$ and,

$M.\delta \cong 2.V$ [Considering large M and a mid-riser type quantizer]

Let us consider a small amplitude interval dx such that the probability density function (pdf) of x(t) within this interval is p(x). So, p(x)dx is the probability that x(t) lies in the range $(x-\frac{dx}{2})$ and $(x+\frac{dx}{2})$. Now, an expression for the mean square quantization error $\overline{e^2}$ can be written as:

$$\overline{e^2} = \int_{x_1-\delta/2}^{x_1+\delta/2} p(x)(x-x_1)^2 dx + \int_{x_2-\delta/2}^{x_2+\delta/2} p(x)(x-x_2)^2 dx + ... \qquad 3.11.2$$

For large M and small δ we may fairly assume that p(x) is constant within an interval, i.e. $p(x) = p_1$ in the 1st interval, $p(x) = p_2$ in the 2nd interval, …., $p(x) = p_k$ in the k^{th} interval.

Therefore, the previous equation can be written as

$$\overline{e^2} = (p_1 + p_2 + ...) \int_{-\delta/2}^{\delta/2} y^2 dy$$

Where, $y = x-x_k$ for all 'k'.

So,

$$\overline{e^2} = (p_1 + p_2 + ...)\frac{\delta^3}{12}$$

$$= [(p_1 + p_2 + ...)\delta]\frac{\delta^2}{12}$$

Now, note that $(p_1 + p_2 + \ldots + p_k + \ldots)\delta = 1.0$

$$\therefore \overline{e^2} = \frac{\delta^2}{12}$$

The above mean square error represents power associated with the random error signal. For convenience, we will also indicate it as N_Q.

Calculation of Signal Power (S_i)

After getting an estimate of quantization noise power as above, we now have to find the signal power. In general, the signal power can be assessed if the signal statistics (such as the amplitude distribution probability) is known. The power associated with x (t) can be expressed as

$$S_i = \overline{x^2(t)} = \int_{-V}^{+V} x^2(t)p(x)\,dx$$

where p (x) is the pdf of x (t). In absence of any specific amplitude distribution it is common to assume that the amplitude of signal x (t) is uniformly distributed between ±V.

In this case, it is easy to see that

$$S_i = \overline{x^2(t)} = \int_{-V}^{+V} x^2(t)\frac{1}{2V}dx = \left[\frac{x^3}{3.2V}\right]_{-V}^{+V} = \frac{V^2}{3} = \frac{(M\delta)^2}{12}$$

Now the SNR can be expressed as,

$$\frac{S_i}{N_Q} = \frac{\frac{V^2}{3}}{\frac{\delta^2}{12}} = \frac{\frac{(M\delta)^2}{12}}{\frac{\delta^2}{12}} = M^2$$

It may be noted from the above expression that this ratio can be increased by increasing the number of quantizer levels N.

Also note that S_i is the power of x (t) at input of the sampler and hence, may not represent the SQNR at the output of the low pass filter in PCM decoder. However, for large N, small δ and ideal and smooth filtering (e.g. Nyquist filtering) at the PCM

decoder, the power S_o of desired signal at the output of the PCM decoder can be assumed to be almost the same as S_i i.e.,

$$S_o \simeq S_i$$

With this justification the SQNR at the output of a PCM codec, can be expressed as,

$$SQNR = \frac{S_o}{N_Q} \simeq M^2 = \left(2^N\right)^2 = 4^N$$

and in dB,

$$\left.\frac{S_o}{N_Q}\right|_{dB} = 10\log_{10}\left(\frac{S_o}{N_Q}\right) \simeq 6.02NdB$$

A few observations

(a) Note that if actual signal excursion range is less than $\pm$ V, $S_o / N_o < 6.02$NdB.

(b) If one quantized sample is represented by 8 bits after encoding i.e., N = 8, $SQNR \simeq 48dB$.

(c) If the amplitude distribution of x (t) is not uniform, then the above expression may not be applicable.

Problems

Q3.11.1) If a sinusoid of peak amplitude 1.0V and of frequency 500Hz is sampled at 2 k-sample /sec and quantized by a linear quantizer, determine SQNR in dB when each sample is represented by 6 bit.

Q3.11.2) How much is the improvement in SQNR of problem 3.11.1 if each sample is represented by 10 bits?

Q3.11.3) What happens to SQNR of problem 3.11.2 if each sampling rate is changed to 1.5 k-samples/ sec?

Module 3

Quantization and Coding

Lesson 12

Logarithmic Pulse Code Modulation (Log PCM) and Companding

After reading this lesson, you will learn about:

- ***Reason for logarithmic PCM;***
- ***A-law and μ–law Companding;***

In a linear or uniform quantizer, as discussed earlier, the quantization error in the k-th sample is

$$e_k = x(t) - x_q(kT_s) \qquad 3.12.1$$

and the maximum error magnitude in a quantized sample is,

$$Max|e_k| = \frac{\delta}{2} \qquad 3.12.2$$

So, if x (t) itself is small in amplitude and such small amplitudes are more probable in the input signal than amplitudes closer to '± V', it may be guessed that the quantization noise of such an input signal will be significant compared to the power of x (t). This implies that SQNR of usually low signal will be poor and unacceptable. In a practical PCM codec, it is often desired to design the quantizer such that the SQNR is almost independent of the amplitude distribution of the analog input signal x (t).

This is achieved by using a non-uniform quantizer. A non-uniform quantizer ensures smaller quantization error for small amplitude of the input signal and relatively larger step size when the input signal amplitude is large. The transfer characteristic of a non – uniform quantizer has been shown in **Fig 3.12.1**. A non-uniform quantizer can be considered to be equivalent to an amplitude pre-distortion process [denoted by y = c (x) in **Fig 3.12.2**] followed by a uniform quantizer with a fixed step size 'δ'. We now briefly discuss about the characteristics of this pre-distortion or 'compression' function y = c (x).

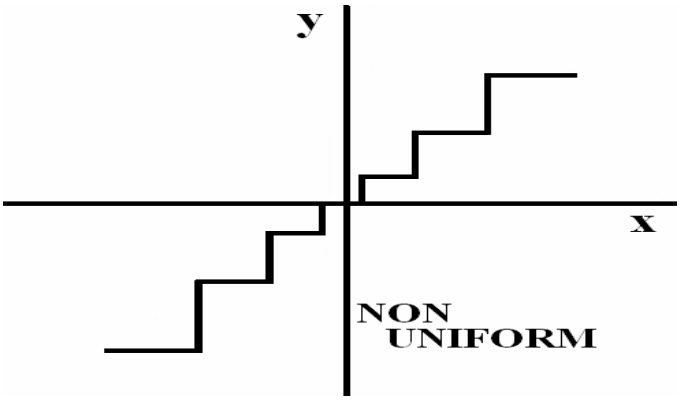


Fig 3.12.1 *Transfer characteristic of a non-uniform quantizer*

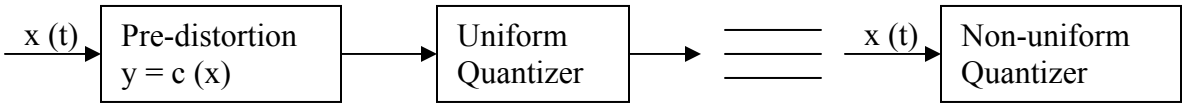


Fig. 3.12.2 *An equivalent form of a non-uniform quantizer*

Mathematically, c (x) should be a monotonically increasing function of 'x' with odd symmetry **Fig 3.12.3**. The monotonic property ensures that c^{-1} (x) exists over the range of 'x(t)' and is unique with respect to c (x) i.e., $c(x)\times c^{-1}(x)=1$.

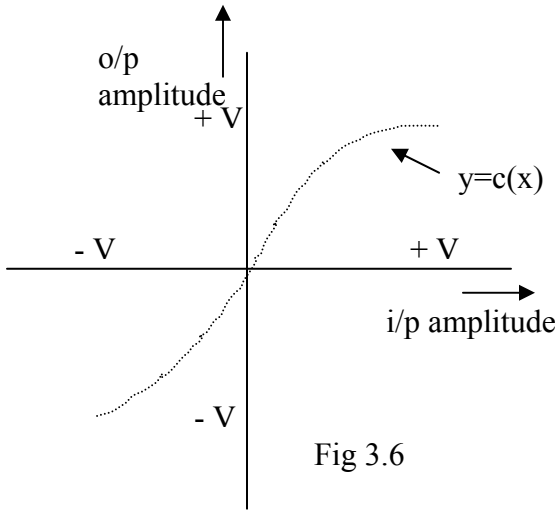


Fig. 3.12.3 *A desired transfer characteristic for non-linear quantization process*

Remember that the operation of c^{-1} (x) is necessary in the PCM decoder to get back the original signal undistorted. The property of odd symmetry i.e., c (-x) = - c (x) simply takes care of the full range '± V' of x (t). The range '± V' of x (t) further implies the following:

$$\begin{aligned} c(x) &= +V, && \text{for } x = +V; \\ &= 0, && \text{for } x = 0; \\ &= -V, && \text{for } x = -V; \end{aligned} \qquad 3.12.3$$

Let the k-th step size of the equivalent non-linear quantizer be 'δ_k' and the number of signal intervals be 'M'. Further let the k-th representation level after quantization when the input signal lies between 'x_k' and 'x_{k+1}' be 'y_k' where

$$y_k = \frac{1}{2}\left(x_k + x_{k+1}\right), \; k = 0,1,\ldots..,(M-1) \qquad 3.12.4$$

The corresponding quantization error 'e_k' is

$$e_k = x - y_k ; \quad x_k < x \le x_{k+1}$$

Now observe from **Fig 3.12.3** that 'δ_k' should be small if '$\frac{dc(x)}{dx}$', i.e., the slope of y = c (x) is large.

In view of this, let us make the following simple approximation on c (x):

$$\frac{dc(x)}{dx} \simeq \frac{2V}{M}\frac{1}{\delta_k}, \qquad k = 0,1,\ldots\ldots,(M-1) \qquad 3.12.5$$

and $$\delta_k = x_{k+1} - x_k\ , \qquad k = 0,1,\ldots\ldots,(M-1)$$

Note that, '$\frac{2V}{M}$' is the fixed step size of the uniform quantizer **Fig. 3.12.2**.

Let us now assume that the input signal is zero mean and its pdf p(x) is symmetric about zero. Further for large number of intervals we may assume that in each interval I_k, k = 0,1,…..,(M-1), the p(x) is constant. So if the input signal x (t) is between x_k and x_{k+1}, i.e.,

$x_k < x \le x_{k+1}$,

$$p(x) \simeq p(y_k)$$

So, the probability that x lies in the k-th interval I_k,

$$I_k = p_k \triangleq P_r(x_k < x \le x_{k+1}) = p(y_k)\delta_k \qquad 3.12.6$$

where, $\sum_{0}^{M-1} P_r(x_k < x \le x_{k+1}) = 1$

Now, the mean square quantization error $\overline{e^2}$ can be determined as follows:

$$\overline{e^2} = \int_{-V}^{+V} (x - y_k)^2 p(x) dx$$

$$= \sum_{k=0}^{M-1} \int_{x_k}^{x_{k+1}} (x - y_k)^2 p(y_k) dx$$

$$= \sum_{k=0}^{M-1} \frac{p_k}{\delta_k} \int_{x_k}^{x_{k+1}} (x - y_k)^2 dx$$

$$= \sum_{k=0}^{M-1} \frac{p_k}{\delta_k} \frac{1}{3}\left[(x_{k+1} - y_k)^3 - (x_k - y_k)^3\right]$$

$$= \sum_{k=0}^{M-1} \frac{1}{3}\left(\frac{p_k}{\delta_k}\right)\left\{\left[x_{k+1} - \frac{1}{2}(x_k + x_{k+1})\right]^3 - \left[x_k - \frac{1}{2}(x_k + x_{k+1})\right]^3\right\}$$

$$= \frac{1}{3}\sum_{k=0}^{M-1}\frac{p_k}{\delta_k}\frac{1}{4}\delta_k^3 = \frac{1}{12}\sum_{k=0}^{M-1}p_k\delta_k^2 \qquad 3.12.7$$

Now substituting

$$\delta_k \simeq \frac{2V}{M}\left[\frac{dc(x)}{dx}\right]^{-1}$$

in the above expression, we get an approximate expression for mean square error as

$$\overline{e^2} = \frac{V^2}{3M^2}\sum_{k=0}^{M-1}p_k\left[\frac{dc(x)}{dx}\right]^{-2} \qquad 3.12.8$$

The above expression implies that the mean square error due to non-uniform quantization can be expressed in terms of the continuous variable x, -V< x < +V, and having a pdf p (x) as below:

$$\overline{e^2} \simeq \frac{V^2}{3M^2}\int_{-V}^{+V}p(x)\left[\frac{dc(x)}{dx}\right]^{-2}dx \qquad 3.12.9$$

Now, we can have an expression of SQNR for a non-uniform quantizer as:

$$SQNR \simeq \left(\frac{3M^2}{V^2}\right)\frac{\int_{-V}^{+V}x^2p(x)dx}{\int_{-V}^{+V}p(x)\left[\frac{dc(x)}{dx}\right]^{-2}dx} \qquad 3.12.10$$

The above expression is important as it gives a clue to the desired form of the compression function y = c(x) such that the SQNR can be made largely independent of the pdf of x (t).

It is easy to see that a desired condition is:

$$\frac{dc(x)}{dx} = \frac{K}{x} \quad \text{where } -V < x < +V \text{ and K is a positive constant.}$$

i.e., $$c(x) = V + K\ln\left(\frac{x}{V}\right) \qquad \text{for } x > 0 \qquad 3.12.11$$

and $$c(x) = -c(x) \qquad 3.12.12$$

Note:

Let us observe that c (x) $\rightarrow \pm \infty$ as x $\rightarrow$ 0 from other side. Hence the above c(x) is not realizable in practice. Further, as stated earlier, the compression function c (x) must pass through the origin, i.e., c (x) = 0, for x = 0. This requirement is forced in a compression function in practical systems.

There are two popular standards for non-linear quantization known as

(a) The μ - law companding
(b) The A – law companding.

The μ - law has been popular in the US, Japan, Canada and a few other countries while the A - law is largely followed in Europe and most other countries, including India, adopting ITU-T standards.

The compression function c (x) for μ - law companding is (**Fig. 3.12.4** and **Fig. 3.12.5**):

$$\frac{c(|x|)}{V} = \frac{\ln\left(1+\frac{\mu|x|}{V}\right)}{\ln(1+\mu)}, \qquad 0 \le \frac{|x|}{V} \le 1.0 \qquad 3.12.13$$

'μ' is a constant here. The typical value of μ lies between 0 and 255. μ = 0 corresponds to linear quantization.

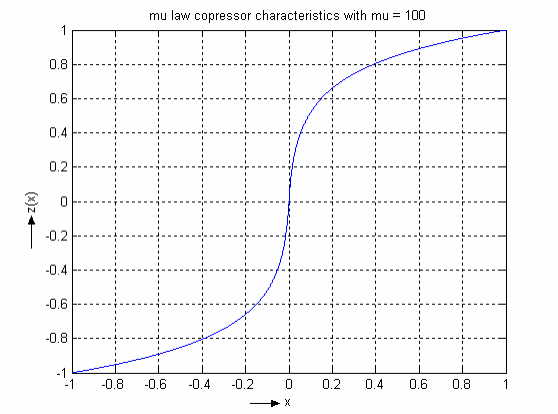


Fig. 3.12.4 *μ-law companding characteristics(mu = 100)*

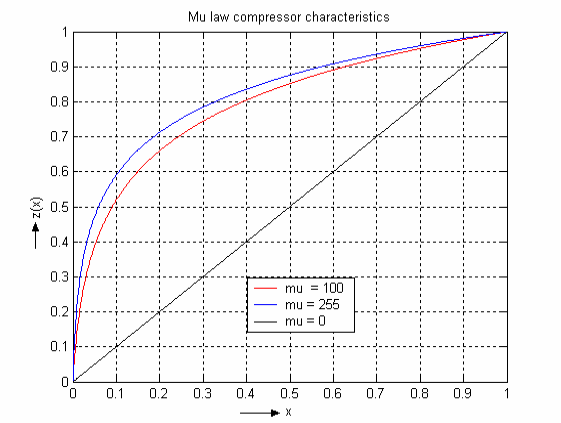


Fig. 3.12.5 *μ-law companding characteristics (mu = 0, 100, 255)*

The compression function c (x) for A - law companding is (**Fig. 3.12.6**):

$$\frac{c(|x|)}{V} = \frac{A\frac{|x|}{V}}{1+\ln A}, \qquad 0 \le \frac{|x|}{V} \le \frac{1}{A}$$

$$= \frac{1+\ln\left(A\frac{|x|}{V}\right)}{1+\ln A}, \qquad \frac{1}{A} \le \frac{|x|}{V} \le 1.0 \qquad 3.12.14$$

'A' is a constant here and the typical value used in practical systems is 87.5.

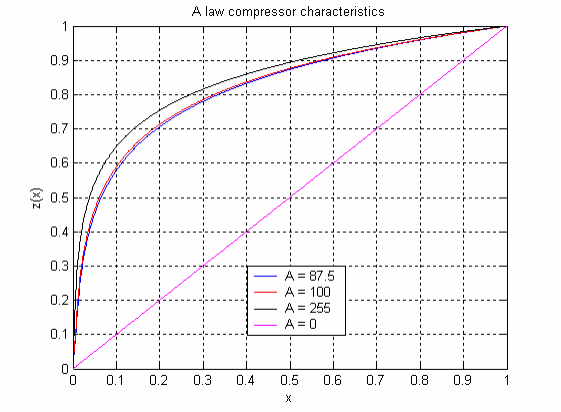


Fig. 3.12.6 *A-law companding characteristics (A = 0, 87.5, 100, 255)*

For telephone grade speech signal with 8-bits per sample and 8-Kilo samples per second, a typical SQNR of 38.4 dB is achieved in practice.

As approximately logarithmic compression function is used for linear quantization, a PCM scheme with non-uniform quantization scheme is also referred as "Log PCM" or "Logarithmic PCM" scheme.

Problems

Q3.12.1) Consider Eq. 3.12.13 and sketch the compression of c (x) for $\mu = 50$ and V = 2.0V

Q3.12.2) Sketch the compression function c (x) for A - law companding (Eq.3.12.14) when V = 1V and A = 50.

Q3.12.3) Comment on the effectiveness of a non-linear quantizer when the peak amplitude of a signal is known to be considerably smaller than the maximum permissible voltage V.

Module 3

Quantization and Coding

Lesson 13

Differential Pulse Code Modulation (DPCM)

After reading this lesson, you will learn about

- ***Principles of DPCM;***
- ***DPCM modulation and de-modulation;***
- ***Calculation of SQNR;***
- ***One tap predictor;***

The standard sampling rate for pulse code modulation (PCM) of telephone grade speech signal is f_s = 8 Kilo samples per sec with a sampling interval of 125 μ sec. Samples of this band limited speech signal are usually correlated as amplitude of speech signal does not change much within 125 μ sec. A typical auto correlation function R (τ) for speech samples at the rate 8 Kilo samples per sec is shown in **Fig 3.13.1**. R (τ = 125 μ sec) is usually between 0.79 and 0.87. This aspect of speech signal is exploited in differential pulse code modulation (DPCM) technique. A schematic diagram for the basic DPCM modulator is shown in **Fig 3.13.2.** Note that a predictor block, a summing unit and a subtraction unit have been strategically added to the chain of blocks of PCM coder instead of feeding the sampler output x (kT_s) directly to a linear quantizer. An error sample e_p (kT_s) is fed.

The error sample is given by the following expression:

$$e_p(kT_s) = x(kT_s) - \hat{x}(kT_s) \qquad 3.13.1$$

$\hat{x}(kT_s)$ is a predicted value for $x(kT_s)$ and is supposed to be close to $x(kT_s)$ such that $e_p(kT_s)$ is very small in magnitude. $e_p(kT_s)$ is called as the 'prediction error for the n-th sample'.

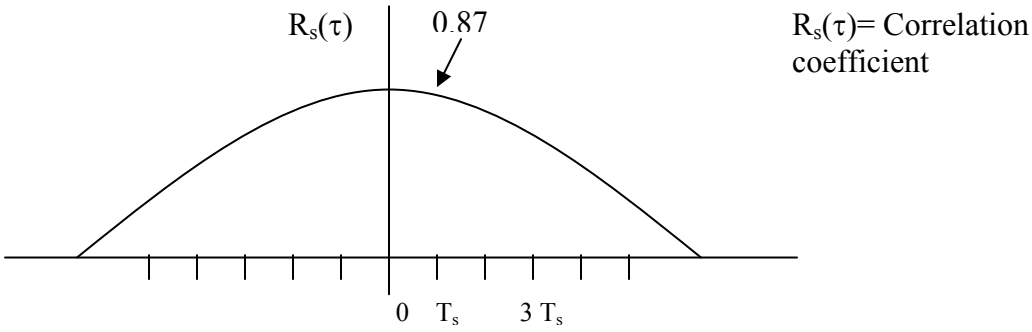


Fig. 3.13.1 *Typical normalized auto-correlation coefficient for speech signal*

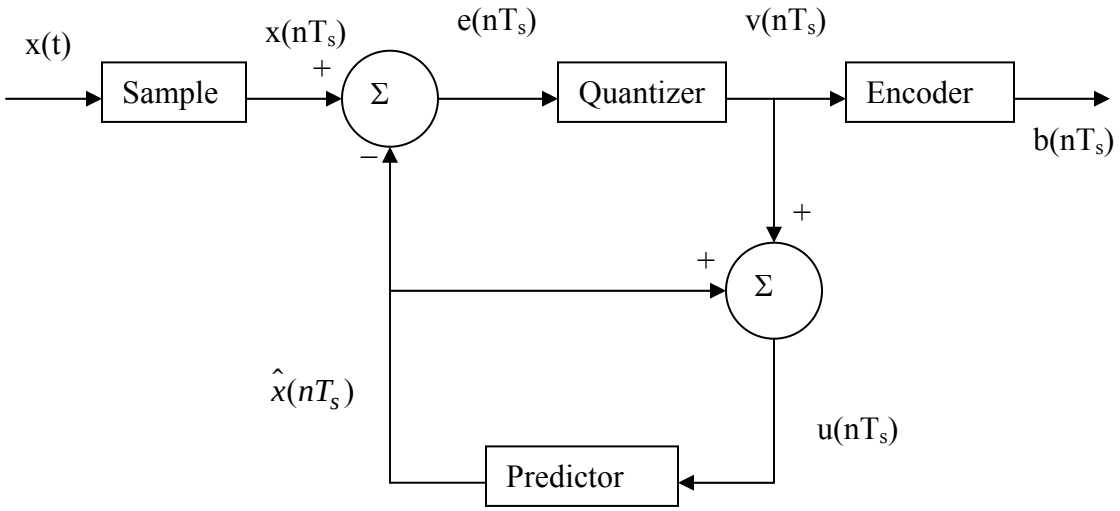


Fig. 3.13.2 *Schematic diagram of a DPCM modulator*

If we assume a final enclosed bit rate of 64kbps as of a PCM coder, we envisage smaller step size for the linear quantizer compared to the step size of an equivalent PCM quantizer. As a result, it should be possible to achieve higher SQNR for DPCM codec delivering bits at the same rate as that of a PCM codec.

There is another possibility of decreasing the coded bit rate compared to a PCM system if an SQNR as achievable by a PCM codec with linear equalizer is sufficient. If the predictor output $\hat{x}(kT_s)$ can be ensured sufficiently close to $x(kT_s)$ then we can simply encode the quantizer output sample $v(kT_s)$ in less than 8 bits. For example, if we choose to encode each of $v(kT_s)$ by 6 bits, we achieve a serial bit rate of 48 kbps, which is considerably less than 64 Kbps. This is an important feature of DPCM, especially when the coded speech signal will be transmitted through wireless propagation channels.

We will now develop a simple analytical structure for a DPCM encoding scheme to bring out the role that nay be played by the prediction unit.
As noted earlier,

$e_p(kT_s)$ = k-th input to quantizer = $x(kT_s) - \hat{x}(kT_s)$

$\hat{x}(kT_s)$ = prediction of the k-th input sample $x(kT_s)$.

$e_q(kT_s)$ = quantizer output for k-th prediction error.

$= c\left[e_p(kT_s)\right]$, where c[] indicates the transfer characteristic of the quantizer

If $q(kT_s)$ indicates the quantization error for the k-th sample, it is easy to see that

$$e_q(kT_s) = e_p(kT_s) + q(kT_s) \qquad 3.13.2$$

Further the input $u(kT_s)$ to the predictor is,

$$\begin{aligned} u(kT_s) &= \hat{x}(kT_s) + e_q(kT_s) = \hat{x}(kT_s) + e_p(kT_s) + q(kT_s) \\ &= x(kT_s) + q(kT_s) \end{aligned} \qquad 3.13.3$$

This equation shows that $u(kT_s)$ is indeed a quantized version of $x(kT_s)$. For a good prediction $e_p(kT_s)$ will usually be small compared to $x(kT_s)$ and $q(kT_s)$ in turn will be very small compared to. Hence, the predictor unit should be so designed that variance of $q(kT_s)$ < variance of $e_p(kT_s)$ << variance of $x(kT_s)$.

A block schematic diagram of a DPCM demodulator is shown in **Fig 3.13.3**. The scheme is straightforward and it tries to estimate $u(kT_s)$ using a predictor unit identical to the one used in the modulator. We have already observed that $u(kT_s)$ is very close to $x(kT_s)$ within a small quantization error of $q(kT_s)$. The analog speech signal is obtained by passing the samples $\hat{u}(kT_s)$ through an appropriate low pass filter. This low pass filter should have a 3 dB cut off frequency at 3.4kHz.

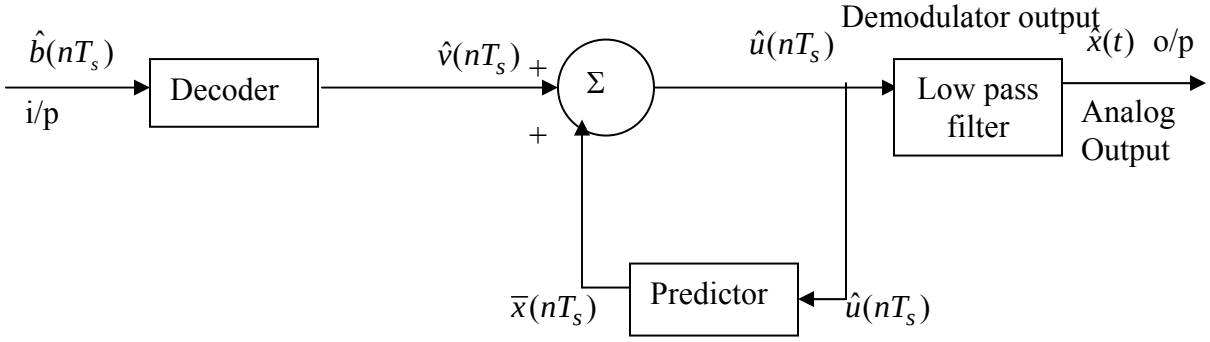


Fig. 3.13.3 *Schematic diagram of a DPCM demodulator; note that the demodulator is very similar to a portion of the modulator*

Calculation of SQNR for DPCM

The expression for signal to quantization noise power ratio for DPCM coding is:

$$SQNR = \frac{Variance\ of\ x(kT_s)}{Variance\ of\ q(kT_s)}$$

$$= \left[\frac{Variance\ of\ x(kT_s)}{Variance\ of\ e_p(kT_s)}\right].\left[\frac{Variance\ of\ e_p(kT_s)}{Variance\ of\ q(kT_s)}\right]$$

As in PCM coding, we are assuming instantaneous sampling and ideal low pass filtering. The first term in the above expression is the 'predictor gain (G_p)'. This gain visibly increases for better prediction, i.e., smaller variance of $e_p(kT_s)$. The second term, SNR_p is a property of the quantizer. Usually, a linear or uniform quantizer is used for simplicity in a DPCM codec. Good and efficient design of the predictor plays a crucial role in enhancing quality of signal or effectively reducing the necessary bit rate for transmission. In the following we shall briefly take up the example of a single-tap predictor.

Single-Tap Prediction

A single-tap predictor predicts the next input example $x(kT_s)$ from the immediate previous input sample $x([k-1]T_s)$.

Let, $\hat{x}(kT_s) = \hat{x}(k|k-1)$ = the k-th predicted sample, given the (k-1)th input sample

$$= a.u(k-1|k-1)$$

Here, 'a' is known as the prediction co-efficient and $u(k-1|k-1)$ is the (k-1)-th input to the predictor given the (k-1)-th input speech sample, i.e., x(k-1).

Now the k-th prediction error sample at the input of the quantizer may be expressed as

$$e_p(kT_s) \equiv e_p(k)$$

$$= x(k) - \hat{x}(k|k-1)$$

$$= x(k) - a.u(k-1|k-1) \qquad 3.13.4$$

The mean square error or variance of this prediction error samples is the statistical expectation of $e^2_p(k)$.

Now,

$$E[e^2{}_p(k)] = E[\{x(k) - a.u(k-1|k-1)\}^2]$$

$$= E[x(k).x(k) - 2.a.x(k).u(k-1|k-1) + a^2.u(k-1|k-1).u(k-1|k-1)]$$

$$= E[x(k).x(k) - 2aE[x(k).u(k-1|k-1)] + a^2.E[u(k-1|k-1).u(k-1|k-1)]$$

3.13.5

Let us note that E[x(k).x(k)]=R(0). Where ($R(\tau)$ indicates the autocorrelation coefficient. For the second term, let us assume that $u(k-1|k-1)$ is an unbiased estimate of x(k-1) and that $u(k-1|k-1)$ is a satisfactorily close estimate of x(k-1), so that we can use the following approximation:

$$E[x(k).u(k-1|k-1)] \simeq E[x(k).x(k-1)]$$
$$= R(\tau = 1.T_s) \equiv R(1), \; say$$

The third term in the expanded form of $E[e^2_p(k)]$ can easily be identified as:

$$a^2.E[u(k-1|k-1).u(k-1|k-1)] = a^2.R(\tau = 0) = a^2.R(0)$$

$$\therefore E[e^2{}_p(k)] = R(0) - 2.a.R(1) + a^2 R(0)$$

$$= R(0)[1 - 2a.\frac{R(1)}{R(0)} + a^2] \qquad 3.13.6$$

The above expression shows that the mean square error or variance of the prediction error can be minimized if a = R(1)/R(0).

Problems

Q3.13.1) Is there any justification for DPCM, if the samples of a signal are known to be uncorrelated with each other?

Q3.13.2) Determine the value of prediction co-efficient for one tap prediction unit if R(0) = 1.0 and R(1) = 0.9.

Module 3

Quantization and Coding

Lesson 14

Delta Modulation (DM)

After reading this lesson, you will learn about

- ***Principles and features of Delta Modulation;***
- ***Advantages and limitations of Delta Modulation;***
- ***Slope overload distortion;***
- ***Granular Noise;***
- ***Condition for avoiding slope overloading;***

If the sampling interval 'Ts' in DPCM is reduced considerably, i.e. if we sample a band limited signal at a rate much faster than the Nyquist sampling rate, the adjacent samples should have higher correlation (**Fig. 3.14.1**). The sample-to-sample amplitude difference will usually be very small. So, one may even think of only 1-bit quantization of the difference signal. The principle of Delta Modulation (DM) is based on this premise. Delta modulation is also viewed as a 1-bit DPCM scheme. The 1-bit quantizer is equivalent to a two-level comparator (also called as a hard limiter). **Fig. 3.14.2** shows the schematic arrangement for generating a delta-modulated signal. Note that,

$$e(kT_s) = x(kT_s) - \hat{x}(kT_s) \qquad 3.14.1$$

$$= x(kT_s) - u([k-1]T_s) \qquad 3.14.2$$

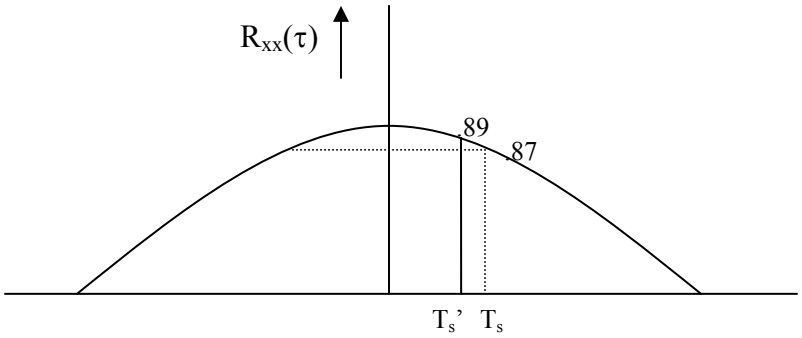


Fig. 3.14.1 *The correlation increases when the sampling interval is reduced*

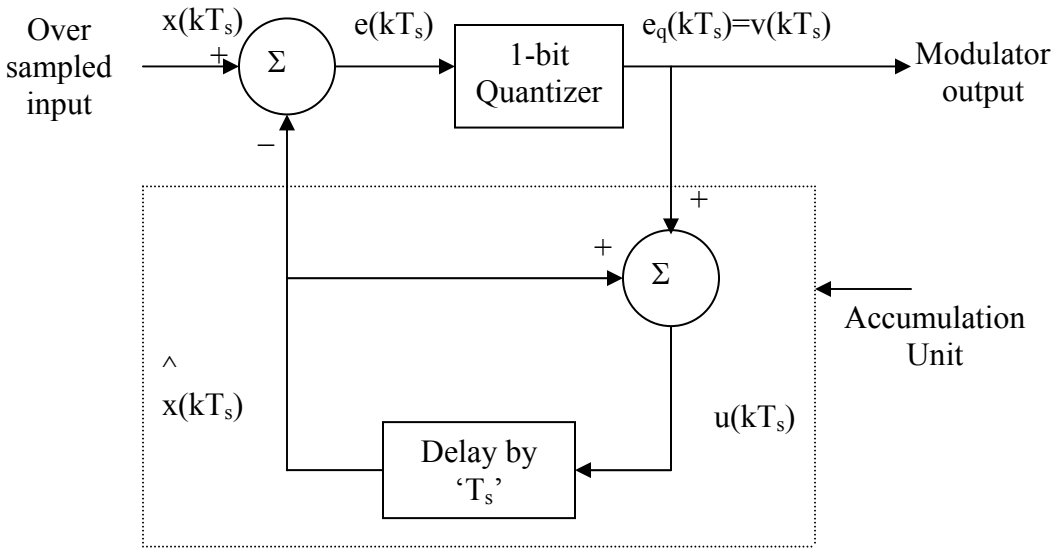


Fig. 3.14.2 *Block diagram of a delta modulator*

Some interesting features of Delta Modulation

- No effective prediction unit – the prediction unit of a DPCM coder (**Fig. 3.13.2**) is eliminated and replaced by a single-unit delay element.
- A 1-bit quantizer with two levels is used. The quantizer output simply indicates whether the present input sample $x(kT_s)$ is more or less compared to its accumulated approximation $\hat{x}(kT_s)$.
- Output $\hat{x}(kT_s)$ of the delay unit changes in small steps.
- The accumulator unit goes on adding the quantizer output with the previous accumulated version $\hat{x}(kT_s)$.
- $u(kT_s)$, is an approximate version of $x(kT_s)$.
- Performance of the Delta Modulation scheme is dependent on the sampling rate. Most of the above comments are acceptable only when two consecutive input samples are very close to each other.

■

Now, referring back to **Fig. 3.14.2**, we see that,

$$e(kT_s) = x(kT_s) - \{\hat{x}([k-1]T_s) + v([k-1]T_s)\} \qquad 3.14.3$$

Further,

$$v(kT_s) = e_q(kT_s) = s.sign[e(kT_s)] \qquad 3.14.4$$

Here, 's' is half of the step-size δ as indicated in **Fig. 3.14.3.**

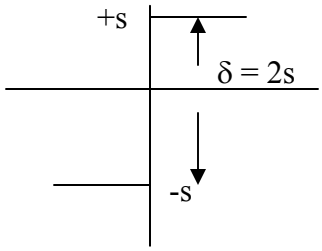


Fig. 3.14.3 *This diagram indicates the output levels of 1-bit quantizer. Note that if δ is the step size, the two output levels are ± s*

Now, assuming zero initial condition of the accumulator, it is easy to see that

$$u(kT_s) = s.\sum_{j=1}^{k} sign[e(jT_s)]$$

$$u(kT_s) = \sum_{j=1}^{k} v(jT_s) \qquad 3.14.5$$

Further,

$$\hat{x}(kT_s) = u([k-1]T_s) = \sum_{j=1}^{k-1} v(jT_s) \qquad 3.14.6$$

Eq. 3.14.6 shows that $\hat{x}(kT_s)$ is essentially an accumulated version of the quantizer output for the error signal $e(kT_s)$. $\hat{x}(kT_s)$ also gives a clue to the demodulator structure for DM. **Fig. 3.14.4** shows a scheme for demodulation. The input to the demodulator is a binary sequence and the demodulator normally starts with no prior information about the incoming sequence.

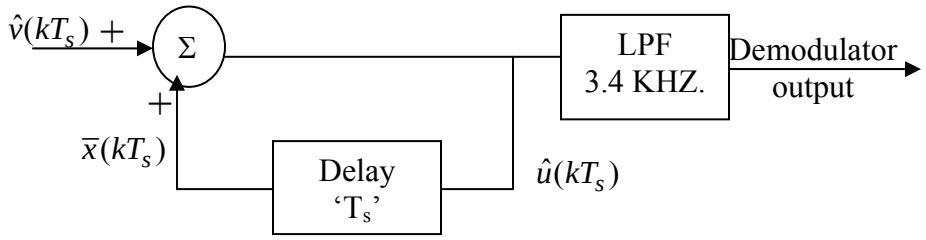


Fig. 3.14.4 *Demodulator structure for DM*

Now, let us recollect from our discussion on DPCM in the previous lesson (Eq. 3.13.3) that, $u(kT_s)$ closely represents the input signal with small quantization error $q(kT_s)$, i.e.

$$u(kT_s) = x(kT_s) + q(kT_s) \qquad 3.14.7$$

Next, from the close loop including the delay-element in the accumulation unit in the Delta modulator structure, we can write

$$u([k-1]T_s) = \hat{x}(kTs) = x(kT_s) - e(kT_s) = x([k-1]T_s) + q([k-1]T_s) \qquad 3.14.8$$

Hence, we may express the error signal as,

$$e(kT_s) = \{x(kT_s) - x([k-1]T_s)\} - q([k-1]T_s) \qquad 3.14.9$$

That is, the error signal is the difference of two consecutive samples at the input except the quantization error (when quantization error is small).

Advantages of a Delta Modulator over DPCM

a) As one sample of $x(kT_s)$ is represented by only one bit after delta modulation, no elaborate word-level synchronization is necessary at the input of the demodulator. This reduces hardware complexity compared to a PCM or DPCM demodulator. Bit-timing synchronization is, however, necessary if the demodulator in implemented digitally.

b) Overall complexity of a delta modulator-demodulator is less compared to DPCM as the predictor unit is absent in DM.

However DM also suffers from a few **limitations** such as the following:

a) **Slope over load distortion***:* If the input signal amplitude changes fast, the step-by-step accumulation process may not catch up with the rate of change (see the sketch in **Fig. 3.14.5**). This happens initially when the demodulator starts operation from cold-start but is usually of negligible effect for speech. However, if this phenomenon occurs frequently (which indirectly implies smaller value of auto-correlation co-efficient $R_{xx}(\tau)$ over a short time interval) the quality of the received signal suffers. The received signal is said to suffer from slope-overload distortion. An intuitive remedy for this problem is to increase the step-size δ but that approach has another serious lacuna as noted in b).

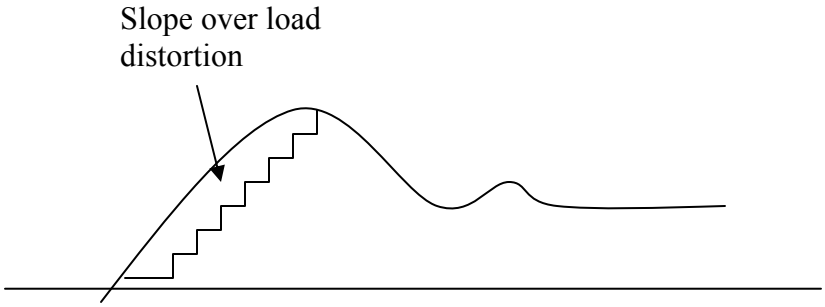


Fig. 3.14.5 *A sketch indicating slope-overload problem. The horizontal axis represents time. The continuous line represents the analog input signal, before sampling and the stair-case represents the output* $\hat{x}(kT_s)$ *of the delay element.*

b) **Granular noise:** If the step-size is made arbitrarily large to avoid slope-overload distortion, it may lead to 'granular noise'. Imagine that the input speech signal is fluctuating but very close to zero over limited time duration. This may happen due to pauses between sentences or else. During such moments, our delta modulator is likely to produce a fairly long sequence of 101010…., reflecting that the accumulator output is close but alternating around the input signal. This phenomenon is manifested at the output of the delta demodulator as a small but perceptible noisy background. This is known as 'granular noise'. An expert listener can recognize the crackling sound. This noise should be kept well within a tolerable limit while deciding the step-size. Larger step-size increases the granular noise while smaller step size increases the degree of slope-overload distortion. In the first level of design, more care is given to avoid the slope-overload distortion. We will briefly discuss about this approach while keeping the step-size fixed. A more efficient approach of adapting the step-size, leading to Adaptive Delta Modulation (ADM) , is excluded.

Condition for avoiding slope overload: From **Fig. 3.14.3** we may observe that if an input signal changes more than half of the step size (i.e. by 's') within a sampling interval, there will be slope-overload distortion. So, the desired limiting condition on the input signal x(t) for avoiding slope-overloading is,

$$\frac{dx(t)}{dt}\Big|_{\max} \leq \frac{s}{T_s} \qquad 3.14.10$$

Quantization Noise Power

Let us consider a sinusoid representing a narrow band signal $x(t) = a_m \cos(2\pi ft)$ where 'f' represents the maximum frequency of the signal and 'a_m ' its peak amplitude. There will be no slope-overload error if

$$\frac{s}{T_s} \geq 2\pi a_m f \quad \text{or} \quad a_m \leq \frac{s}{2\pi f T_s}$$

The above condition effectively limits the power of x(t). The maximum allowable power of x(t) = $P_{max} = \frac{a_m^2}{2} = \frac{s^2}{8\pi^2 f^2 T_s^2}$.

Once the slope overload distortion has been taken care of, one can find an estimate of $SQNR_{max}$. Assuming uniform quantization noise between +s and –s, the quantization noise power is

$$N_Q = \frac{4s^2}{12} = \frac{s^2}{3}$$

Let us now recollect that the sampling frequency $f_s = 1/T_s$ is much greater than 'f' . The granular noise due to the quantizer can be approximated to be of uniform power spectral density over a frequency band upto f_s **(Fig. 3.14.6)**. The low pass filter at the output end of the delta demodulator is designed as per the bandwidth of x(t) and much of the quantization noise power is filtered off. Hence, we may write,

the in-band quantization noise power $\approx \frac{f}{f_s}.N_Q$

Therefore, $SQNR_{max}$= (Maximum signal power) / (In-band quantization noise power)

$$= \left(\frac{3}{8\pi^2}\right).\left(\frac{f_s}{f}\right)^3$$

The above expression indicates that we can expect an improvement of about 9dB by doubling the sampling rate and it is not a very impressive feature when compared with a PCM scheme. Typically, when the permissible data rate after quantization and coding of speech signal is more than 48 Kbps, PCM offers better SQNR compared to linear DM.

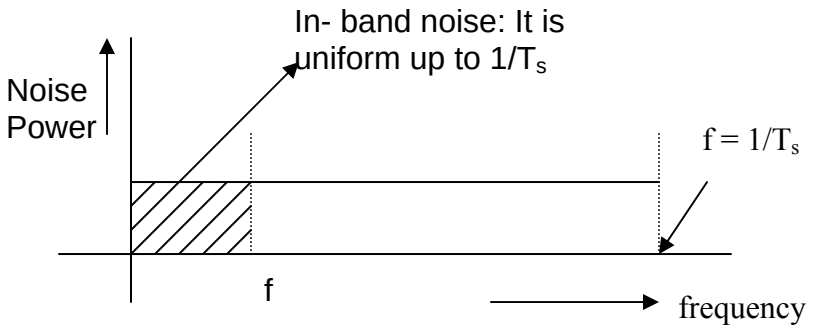


Fig. 3.14.6 *In-band noise power at the output of the low pass filter in a delta demodulator is shown by the shaded region*

Problems

Q3.14.1) Comment if a delta modulator can also be called as a 1-bit DPCM scheme.

Q3.14.2) Mention two differences between DPCM and Delta Modulator

Q3.14.3) Suggest a solution for controlling the granular noise at the output of a delta modulator.

Q3.14.4) Let $x(t) = 2 \cos (2\pi \times 100t$. If this signal is sampled at 1 KHz for delta modulator, what is the maximum achievable SQNR in dB?

Module 4

Signal Representation and Baseband Processing

Lesson 15

Orthogonality

After reading this lesson, you will learn about

- ***Basic concept of orthogonality and orthonormality;***
- ***Strum - Lion;***
- ***Slope overload distortion;***
- ***Granular Noise;***
- ***Condition for avoiding slope overloading;***

The Issue of Orthogonality

Let $f_m(x)$ and $f_n(x)$ be two real valued functions defined over the interval $a \le x \le b$. If the product $[f_m(x) \times f_n(x)]$ exists over the interval, the two functions are called orthogonal to each other in the interval $a \le x \le b$ when the following condition holds:

$$\int_b^a f_m(x) f_n(x) dx = 0, \qquad m \neq n \qquad 4.15.1$$

A set of real valued functions $f_1(x)$, $f_2(x) \ldots f_N(x)$ is called an orthogonal set over an interval $a \le x \le b$ if

(i) all the functions exist in that interval and

(ii) all distinct pairs of the functions are orthogonal to each other over the interval, i.e.

$$\int_b^a f_i(x) f_j(x) dx = 0 \ , \ i = 1,2,\ldots; \qquad j = 1,2,\ldots \text{ and } i \neq j \qquad 4.15.2$$

The norm $\| f_m(x) \|$ of the function $f_m(x)$ is defined as,

$$\| f_m(x) \| = \sqrt{\int_b^a f_m^2(x) dx} \qquad 4.15.3$$

An orthogonal set of functions $f_1(x)$, $f_2(x) \ldots f_N(x)$ is called an orthonormal set if,

$$\int_b^a f_m(x) . f_n(x) = \begin{cases} 0, m \neq n \\ 1, m = n \end{cases} \qquad 4.15.4$$

An orthonormal set can be obtained from a corresponding orthogonal set of functions by dividing each function by its norm. Now, let us consider a set of real functions $f_1(x)$, $f_2(x) \ldots f_N(x)$ such that, for some non-negative weight function $w(x)$ over the interval $a \le x \le b$

$$\int_b^a f_m(x) . f_n(x) . w(x) dx = 0, \qquad m \neq n \qquad 4.15.5$$

Do f_i -s form an orthogonal set? We say that the f_i-s form an orthogonal set with respect to the weight function $w(x)$ over the interval $a \le x \le b$ by defining the norm as,

$$\left\| f_m(x) \right\| = \sqrt{\int_b^a f_m^2(x).w(x)dx} \,. \qquad 4.15.6$$

The set of f_i-s is orthonormal with respect to w(x) if the norm of each function is 1.
The above extension of the idea of orthogonal set makes perfect sense. To see this, let

$$g_m(x) = \sqrt{w(x)}\, f_m(x) \text{, where w(x) is a non-negative function.} \qquad 4.15.7$$

It is now easy to verify that,

$$\int_b^a f_m(x).f_n(x).w(x)dx = \int_b^a g_m(x).g_n(x)dx = 0 \,. \qquad 4.15.8$$

This implies that if we have orthogonal f_i-s over $a \le x \le b$, with respect to a non-negative weight function w(x), then we can form an usual orthogonal set of f_i –s over the same interval $a \le x \le b$ by using the substitution,

$$g_m = \sqrt{w(x)}\, f_m(x)$$

Alternatively, an orthogonal set of g_i-s can be used to get an orthogonal set of f_i-s with respect to a specific non-negative weight function w(x) over $a \le x \le b$ by the following substitution (provided $\sqrt{w(x)} \neq 0$, $a \le x \le b$):

$$f_m(x) = \frac{g_m(x)}{\sqrt{w(x)}} \,. \qquad 4.15.9$$

A real orthogonal set can be generated by using the concepts of Strum-Liouville (S-L) equation. The S-L problem is a boundary value problem in the form of a second order differential equation with boundary conditions. The differential equation is of the following form:

$$\frac{d}{dx}\left[p(x)\frac{dy}{dx} \right] + \left[q(x) + \lambda.\omega(x) \right] y = 0 \,, \text{ for } a \le x \le b; \qquad 4.15.10$$

It satisfies the following boundary conditions:

i) $c_2 \dfrac{dy}{dx} + c_1 y = 0$; at x = a;

ii) $d_2 \dfrac{dy}{dx} + d_1 y = 0$; at x = b;

Here c_1, c_2, d_1 and d_2 are real constants such that at least one of c_1 and c_2 is non zero and at least one of d_1 and d_2 is non zero.

The solution y = 0 is a trivial solution. All other solutions of the above equation subject to specific boundary conditions are known as characteristic functions or eigen-functions of the S-L problem. The values of the parameter 'λ' for the non trivial solutions are known as characteristic values or eigen values. A very important property of the eigen-functions is that they are orthogonal.

Orthogonality Theorem:

Let the functions p(x), q(x) and ω(x) in the S-L equation (4.15.10) be real valued and continuous in the interval $a \le x \le b$. Let $y_m(x)$ and $y_n(x)$ be eigen functions of the S-L problem corresponding to distinct eigenvalues λ_m and λ_n respectively. Then, $y_m(x)$ and $y_n(x)$ are orthogonal over $a \le x \le b$ with respect to the weight function w(x).

Further, if p(x = a) = 0, then the boundary condition (i) may be omitted and if p(x = b) = 0, then boundary condition (ii) may be omitted from the problem. If p(x = a) = p(x = b), then the boundary condition can be simplified as,

$$y(a) = y(b) \text{ and } \frac{dy}{dx}\Big|_{x=a} = y'(a) = y'(b) = \frac{dy}{dx}\Big|_{x=b}$$

Another useful feature is that, the eigenvalues in the S-L problem, which in general may be complex based on the forms of p(x), q(x) and w(x), are real valued when the weight function ω(x) is positive in the interval $a \le x \le b$ or always negative in the interval $a \le x \le b$

Examples of orthogonal sets:

Ex#1: We know that, for integer 'm' and 'n',

$$\int_{-\pi}^{\pi} \cos mx.\cos nx dx = \begin{cases} 0, & m \neq n \\ \pi, & m = n \end{cases} \qquad \text{E4.15.1}$$

$$\int_{-\pi}^{\pi} \sin mx.\sin nx dx = \begin{cases} 0, & m \neq n \\ \pi, & m = n \end{cases} \qquad \text{E4.15.2}$$

and $$\int_{-\pi}^{\pi} \cos mx.\sin nx dx = 0 \qquad \text{E4.15.3}$$

Let us consider equation E4.15.1 and rewrite it as:

$$\int_{-1/2f}^{1/2f} (\cos 2\pi mft).(\cos 2\pi nft) dt = \begin{cases} 0, & m \neq n \\ \pi, & m = n \end{cases} \qquad \text{E4.15.4}$$

by substituting $x = 2\pi ft = \omega t$ and $dx = 2\pi f dt = \omega dt$

Note that the functions 'cosmx' and 'cosnx' are orthogonal over the range 2π of the independent variable x and its integral multiple, i.e. M. 2π, in general, where 'M' is an integer. This implies that equation (E4.15.4) is orthogonal in terms of the independent

variable 't' over the fundamental range $\frac{1}{f}$ and, in general, over M$\frac{1}{f}$ = M T_0, where 'T_0' indicates the fundamental time interval over which cos2πmft and cos2πnft are orthogonal to each other. Now 'm' and 'n' can have a minimum difference '1' if

$$\int_{-T_0}^{T_0} (\cos 2\pi mft).(\cos 2\pi nft)dt = 0 \qquad \text{E4.15.5}$$

i.e., mf –nf = f = $\frac{1}{T_0}$

So, if two cosine signals have a frequency difference 'f', then we may say,

$$\int_{-1/2f}^{1/2f} \cos 2\pi(f_c + \frac{f}{2})t.\cos 2\pi(f_c - \frac{f}{2})t.dt = 0 \qquad \text{E4.15.6}$$

Re-writing equation (E4.15.6)

$$\int_{-T_0}^{T_0} \cos 2\pi(f_c + \frac{f}{2})t.\cos 2\pi(f_c - \frac{f}{2})t.dt = 0 \qquad \text{where, } T_0 = \frac{1}{f}$$

Looking back at equation E4.15.5, we may write a general form for equation (E4.15.6):

$$\int_{-T_0/2}^{T_0/2} \cos 2\pi(f_c + p\frac{f}{2})t.\cos 2\pi(f_c - p\frac{f}{2})t.dt = 0 \qquad \text{E 4.15.7}$$

where mf = (n+p)f and 'p' is an integer.

Following similar observations on equation E4.15.2, one can say,

$$\int_{-T_0/2}^{T_0/2} \sin 2\pi(f_c + p\frac{f}{2})t.\sin 2\pi(f_c - p\frac{f}{2})t.dt = 0 \qquad \text{E4.15.8}$$

Equation E4.15.3 may also be expressed as,

$$\int_{-T_0/2}^{T_0/2} \cos 2\pi(f_c + p\frac{f}{2})t.\sin 2\pi(f_c - p\frac{f}{2})t.dt$$

$$= \int_{-T_0/2}^{T_0/2} \sin 2\pi(f_c + p\frac{f}{2})t.\cos 2\pi(f_c - p\frac{f}{2})t.dt \quad = 0 \qquad \text{E4.15.9}$$

Let us define $s_1 = \cos 2\pi\left(f_c + p\frac{f}{2}\right)t$, $s_2 = \cos 2\pi\left(f_c - p\frac{f}{2}\right)t$, $s_3 = \sin 2\pi\left(f_c + p\frac{f}{2}\right)t$ and $s_4 = \sin 2\pi\left(f_c - p\frac{f}{2}\right)t$. Can we use the above observations on orthogonality to distinguish among 's_i-s' over a decision interval of $T_5 = T_0 = \frac{1}{f}$?

■

Ex#2: $x_1(t) = 1.0$ for $0 \le t \le T/2$ and zero elsewhere,
$x_2(t) = 1.0$ for $T/2 \le t \le T$ and zero elsewhere, ■

Ex#3: $x_1(t) = 1.0$ for $0 \le t \le T/2$ and $x_1(t) = -1.0$ for $T/2 < t \le T$, while $x_2(t) = -1.0$ for $0 \le t \le T$ ■

Importance of the concepts of Orthogonality in Digital Communications

a. In the design and selection of information bearing pulses, orthogonality over a symbol duration may be used to advantage for deriving efficient symbol-by-symbol demodulation scheme.
b. Performance analysis of several modulation demodulation schemes can be carried out if the information-bearing signal waveforms are known to be orthogonal to each other.
c. The concepts of orthogonality can be used to advantage in the design and selection of single and multiple carriers for modulation, transmission and reception.

Orthogonality in a complex domain

Let, $z_1(t) = x_1(t) + jy_1(t)$ and $z_2(t) = x_2(t) + jy_2(t)$

Now, $x_1(t) = \dfrac{z_1(t) + z_1^*(t)}{2}$ and $x_2(t) = \dfrac{z_2(t) + z_2^*(t)}{2}$

If x_1 and x_2 are orthogonal to each other over $a \le t \le b$,

$$\int_a^b x_1(t).x_2(t)\,dt = 0$$

i.e., $$\int_a^b \left[z_1(t) + z_1^*(t)\right]\left[z_2(t) + z_2^*(t)\right]dt = 0$$

or, $$\int_a^b \left[z_1(t).z_2(t) + z_1(t).z_2^*(t) + z_1^*(t).z_2(t) + z_1^*(t).z_2^*(t)\right]dt = 0$$

Let us consider a complex function

$$z_1(t) = x(t) + jy(t), \quad a \le t \le b$$
$$= r(t)\left[\cos\Phi(t) + j\sin\Phi(t)\right]$$

where, $r(t) = \left|\tilde{z}(t)\right|$, a non – negative function of 't'.

$$\therefore x(t) = r(t)\cos\Phi(t) \quad \text{and} \quad y(t) = r(t)\sin\Phi(t)$$

Now, $$\int_a^b x(t).y(t)\,dt = \int_a^b r^2(t).\cos\Phi(t).\sin\Phi(t)dt$$

We know that $\cos\theta$ & $\sin\theta$ are orthogonal to each other over $-\pi \le \theta < \pi$, i.e.,

$$\int_{-\pi}^{\pi} \cos\theta.\sin\theta d\theta = 0$$

So, using a constant weight function w = r, which is non-negative, we may say

$$\int_{-\pi}^{\pi} r^2 \cos\theta.\sin\theta d\theta = 0$$

Now, x = r cos θ and y = r sin θ are also orthogonal over $-\pi \le \theta < \pi$.
Now, let 'θ' be a continuous function of 't' over $-\pi \le \theta < \pi$. And,

$\theta\big|_{t=a} = \theta_a = -\pi$ and $\theta\big|_{t=b} = \theta_b = \pi$

Assuming a linear relationship, let, $\theta(t) = 2\pi ft$

$\therefore d\theta(t) = 2\pi fdt$

Under these conditions, we see,

$$\int_a^b r^2(t).\cos\Phi(t).\sin\Phi(t)dt = \frac{1}{2\pi f}\int_{-\pi}^{\pi} r^2(t).\cos\Phi(t).\sin\Phi(t)d\Phi$$

$$= \frac{1}{2\pi f}\int_{-\pi}^{\pi} r^2.\cos\Phi(t).\sin\Phi(t)d\Phi = 0$$

i.e., x(t) and y(t) are orthogonal over the interval $-\frac{1}{2f} \le t \le \frac{1}{2f}$ or $\frac{T}{2} \le t \le \frac{T}{2}$

So, if $\tilde{z}(t) = x(t) + jy(t)$ represents a phasor in the complex plane rotating at a uniform frequency of 'f' , then x(t) and y(t) are orthogonal to each other over the interval $-\frac{1}{2f} \le t \le \frac{1}{2f}$ or, equivalently $-\frac{T}{2} \le t \le \frac{T}{2}$ where $T = \frac{1}{f}$. i.e., $\int_{-T/2}^{T/2} x(t).y(t)dt = 0$

Now, let us consider two complex functions:

$\tilde{z}_1(t) = x_1(t) + jy_1(t) = \left|\tilde{z}_1(t)\right| e^{j\Phi_1(t)}$

and $\tilde{z}_2(t) = x_2(t) + jy_2(t) = \left|\tilde{z}_2(t)\right| e^{j\Phi_2(t)}$

$[x_1(t),y_1(t)]$ and $[x_2(t),y_2(t)]$ are orthogonal pairs over the interval $-\frac{T}{2} \le t \le \frac{T}{2}$. So, $\tilde{z}_1(t)$ and $\tilde{z}_2(t)$ may be viewed as two phasors rotating with equal speed.

Now, two static phasors are orthogonal to each other if their dot or scalar product is zero, i.e.,

$\overline{A}.\overline{B} = |A||B|\cos\gamma = A_x.B_x + A_y.B_y = 0$, where 'γ is the angle between $\overline{A}$ *and* $\overline{B}$

In general, two complex functions $\tilde{z}_1(t)$ *and* $\tilde{z}_2(t)$ with finite energy are said to be orthogonal to each other over an interval $a \le t \le b$, if

$$\int_a^b \tilde{z}_1(t).\tilde{z}_2^{\,*}(t)dt = 0$$

Problems

Q4.15.1) Verify whether two signals are orthogonal over one time period of the signal with smallest frequency signal.

i) $X_1(t) = \text{Cos } 2\pi ft$ and $X_2(t) = \text{Sin } 2\pi ft$

ii) $X_1(t) = \text{Cos } 2\pi ft$ and $X_2(t) = \text{Cos } (2\pi ft + \frac{\pi}{3})$

iii) $X_1(t) = \text{Cos } 2\pi ft$ and $X_2(t) = \text{Cos } (4\pi ft + \frac{\pi}{4})$

iv) $X_1(t) = \text{Sin } 4\pi ft$ and $X_2(t) = -\text{ Cos } (\pi ft - \frac{\pi}{6})$

Module 4

Signal Representation and Baseband Processing

Lesson 16

Representation of Signals

After reading this lesson, you will learn about

- ***Representation of signals following the Gram-Schmidt orthogonalization procedure;***
- ***Signal space and signal constellation;***
- ***Use of signal space for signal detection;***
- ***The fundamental detection problem in a receiver;***

As mentioned earlier in Module #1, a digital modulator is supposed to accept stream of information-bearing symbols (usually bits) and represent them appropriately with or without the help of a carrier. So, a very important issue in-between is to represent information symbols in suitable energy signals so that the signals can be modulated, amplified and transmitted. For a continuous stream of input of information sequence what kind of strategy should we take to represent them as signals? One may think of multiple alternatives including the following:

- Consider one symbol at a time and design a signal for the symbol.
- When several bits make one symbol, consider one bit at a time and design for the symbol.
- Consider a larger group of symbols in a sequence and design signals spread over long time duration [sequence based modulation - demodulation strategy].

Let us consider a systematic approach to identify M symbols from the input information sequence. If the format of input information is known, this is not a difficult task. For example, if the information sequence is binary and if we choose M = 2, we can identify '1' as one symbol and '0' as the other. Else, if we choose M = 4 for the same binary information sequence; we may consider a group of two bits at a time to define one symbol. The duration of a symbol now is twice the duration of one information bit. If the rate of incoming information is R_b bits/sec, the symbol rate is $R_b/2$ symbols per second. Usually, for practical considerations, M is so chosen that $M = 2^m$, where 'm' is a positive integer.

The next issue is to design 'M' energy signals for these M symbols such that the energy of each signal is limited within the symbol duration. This problem is addressed in general by a scheme known as Gram-Schmidt Orthogonalization

Gram-Schmidt Orthogonalization

The principle of Gram-Schmidt Orthogonalization (GSO) states that, any set of M energy signals, $\{s_i(t)\}$, $1 \le i \le M$ can be expressed as linear combinations of N orthonormal basis functions, where $N \le M$.

If $s_1(t)$, $s_2(t)$, ….., $s_M(t)$ are real valued energy signals, each of duration 'T' sec,

$$s_i(t) = \sum_{j=1}^{N} s_{ij}\Phi_j(t); \quad \begin{cases} 0 \le t \le T \\ i = 1, 2, ..., M \ge N \end{cases} \qquad 4.16.1$$

where,

$$s_{ij} = \int_0^T s_i(t)\varphi_j(t)dt \quad ; \begin{cases} i = 1, 2, ..., M \\ j = 1, 2, ..., N \end{cases} \qquad 4.16.2$$

The $\varphi_j(t)$-s are the basis functions and 's_{ij}'-s are scalar coefficients. We will consider real-valued basis functions $\varphi_j(t)$ - s which are orthonormal to each other, i.e.,

$$\int_0^T \varphi_i(t).\varphi_j(t)dt = \begin{cases} 1, if\ i = j \\ 0, if\ i \neq j \end{cases} \qquad 4.16.3$$

Note that each basis function has unit energy over the symbol duration 'T'. Now, if the basis functions are known and the scalars are given, we can generate the energy signals, by following **Fig. 4.16.1**. Or, alternatively, if we know the signals and the basis functions, we know the corresponding scalar coefficients (**Fig. 4.16.2**).

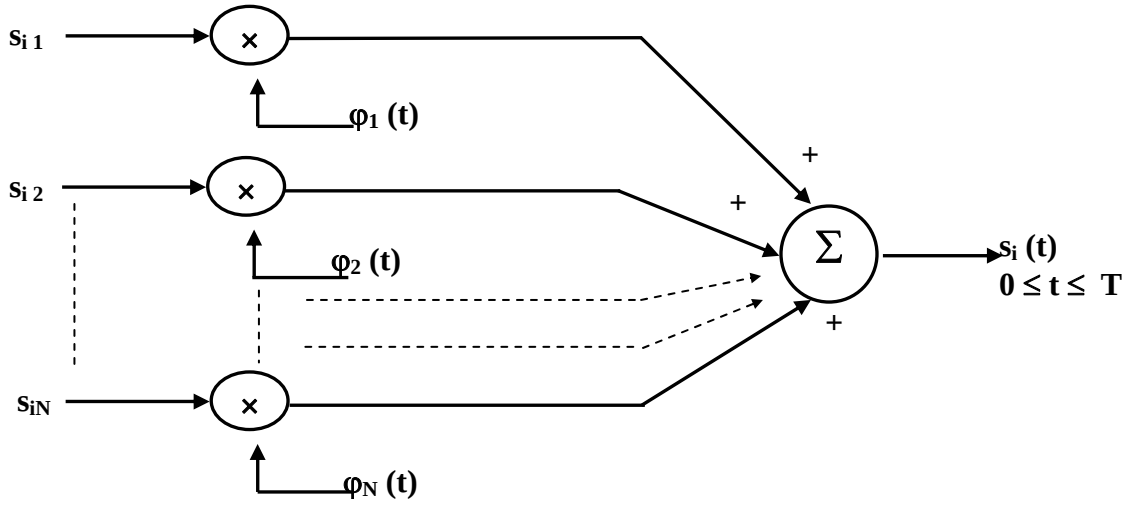


Fig. 4.16.1 *Pictorial depiction of Equation 4.16.1*

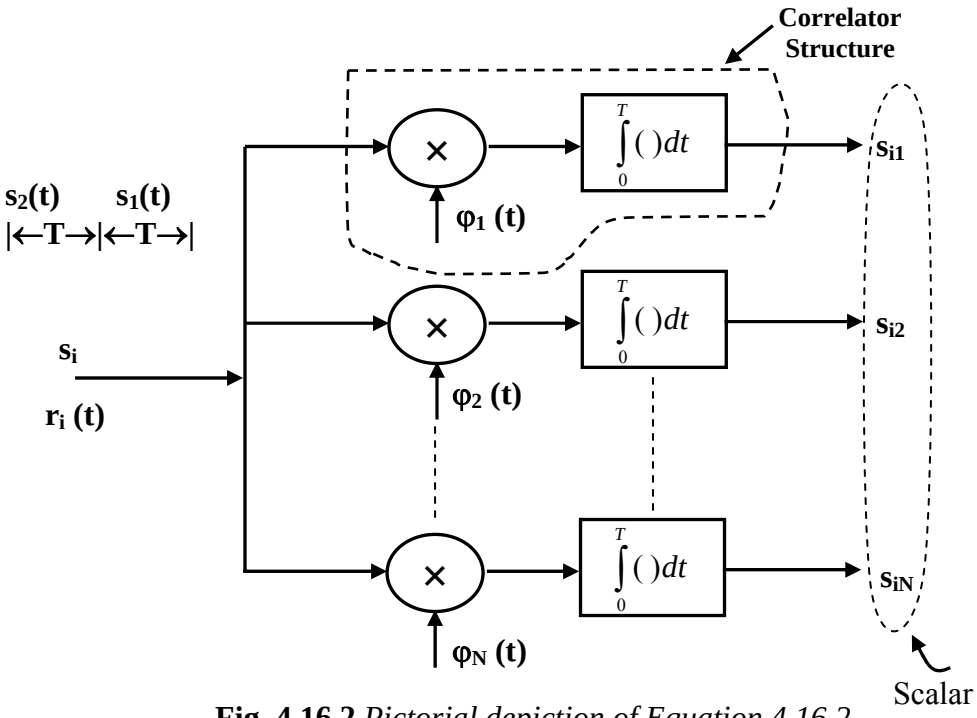


Fig. 4.16.2 *Pictorial depiction of Equation 4.16.2*

Justification for G-S-O procedure

Part – I: We show that any given set of energy signals, $\{s_i(t)\}$, $1 \le i \le M$ over $0 \le t < T$, can be completely described by a subset of energy signals whose elements are linearly independent.

To start with, let us assume that all $s_i(t)$ -s are not linearly independent. Then, there must exist a set of coefficients $\{a_i\}$, $1 < i \le M$, not all of which are zero, such that,

$$a_1 s_1(t) + a_2 s_2(t) + \ldots\ldots + a_M s_M(t) = 0, \quad 0 \le t < T \qquad 4.16.4$$

Verify that even if two coefficients are not zero, e.g. $a_1 \neq 0$ and $a_3 \neq 0$, then $s_1(t)$ and $s_3(t)$ are dependent signals.

Let us arbitrarily set, $a_M \neq 0$. Then,

$$s_M(t) = -\frac{1}{a_M}\left[a_1 s_1(t) + a_2 s_2(t) + \ldots\ldots + a_{M-1} s_{M-1}(t)\right]$$

$$= -\frac{1}{a_M}\sum_{i=1}^{M-1} a_i s_i(t) \qquad 4.16.5$$

Eq.4.16.5 shows that $s_M(t)$ could be expressed as a linear combination of other $s_i(t)$ – s, i = 1, 2, .., (M – 1).

Next, we consider a reduced set with (M-1) signals $\{s_i(t)\}$, i = 1,2,….., (M – 1).
This set may be either linearly independent or not. If not, there exists a set of $\{b_i\}$, i = 1,2…, (M – 1), not all equal to zero such that,

$$\sum_{i=1}^{M-1} b_i s_i(t) = 0, \quad 0 \le t < T \qquad 4.16.6$$

Again, arbitrarily assuming that $b_{M-1} \neq 0$, we may express $s_{M-1}(t)$ as:

$$s_{M-1}(t) = -\frac{1}{b_{M-1}} \sum_{i=1}^{M-2} b_i s_i(t) \qquad 4.16.7$$

Now, following the above procedure for testing linear independence of the remaining signals, eventually we will end up with a subset of linearly independent signals. Let $\{s_i(t)\}$, i = 1, 2, …., N ≤ M denote this subset.

Part – II : We now show that it is possible to construct a set of 'N' orthonormal basis functions $\varphi_1(t)$, $\varphi_2(t)$, ….., $\varphi_N(t)$ from $\{s_i(t)\}$, i = 1, 2, ….., N.

Let us choose the first basis function as, $\varphi_1(t) = \frac{s_1(t)}{\sqrt{E_1}}$, where E_1 denotes the energy of the first signal $s_1(t)$, i.e., $E_1 = \int_0^T s_1^2(t)dt$:

$$\therefore s_1(t) = \sqrt{E_1}.\varphi_1(t) = s_{11}\varphi_1(t) \qquad 4.16.8$$

Where, $s_{11} = \sqrt{E_1}$

Now, from Eq. 4.16.2, we can write

$$s_{21} = \int_0^T s_2(t)\phi_1(t)dt \qquad 4.16.9$$

Let us now define an intermediate function:

$$g_2(t) = s_2(t) - s_{21}\varphi_1(t); \quad 0 \le t < T \qquad 4.16.10$$

Note that,

$$\int_0^T g_2(t)\varphi_1(t)dt = \int_0^T s_2(t)\varphi_1(t)dt - s_{21}\int_0^T \varphi_1(t)\varphi_1(t)dt$$

$$= s_{21} - s_{21} = 0 \rightarrow g_2(t) \text{ Orthogonal to } \varphi_1(t);$$

So, we verified that the function $g_2(t)$ is orthogonal to the first basis function. This gives us a clue to determine the second basis function.
Now, energy of $g_2(t)$

$$= \int_0^T g_2^2(t)dt$$

$$= \int_0^T \left[s_2(t) - s_{21}\varphi_1(t) \right]^2 dt$$

$$= \int_0^T s_2{}^2(t)dt - 2.s_{21}\int_0^T s_2(t)\varphi_1(t)dt + s_{21}{}^2\int_0^T \varphi_1{}^2(t)dt$$

$$= E_2 - 2.s_{21}.s_{21} + s_{21}{}^2 = E_2 - s_{21}{}^2 \quad (Say) \qquad 4.16.11$$

So, we now set,

$$\varphi_2(t) = \frac{g_2(t)}{\sqrt{\int_0^T g_2{}^2(t)dt}} = \frac{s_2(t) - s_{21}\varphi_1(t)}{\sqrt{E_2 - s_{21}{}^2}} \qquad 4.16.12$$

and $E_2 = \int_0^T s_2{}^2(t)dt$: Energy of $s_2(t)$

Verify that:

$\int_0^T \varphi_2{}^2(t)dt = 1$, i.e. $\varphi_2(t)$ is a time limited energy signal of unit energy.

and $\int_0^T \varphi_1(t).\varphi_2(t)dt = 0$, i.e. $\varphi_1(t)$ and $\varphi_2(t)$ are orthonormal to each other.

Proceeding in a similar manner, we can determine the third basis function, $\varphi_3(t)$. For i=3,

$$g_3(t) = s_3(t) - \sum_{j-1}^{2} s_{3j}\varphi_J(t); \qquad 0 \le t < T$$

$$= s_3(t) - \left[s_{31}\varphi_1(t) + s_{32}\varphi_2(t) \right]$$

where,

$$s_{31} = \int_0^T s_3(t)\varphi_1(t)dt \quad \text{and} \quad s_{32} = \int_0^T s_3(t)\varphi_2(t)dt$$

It is now easy to identify that,

$$\varphi_3(t) = \frac{g_3(t)}{\sqrt{\int_0^T g_3{}^2(t)dt}} \qquad 4.16.13$$

Indeed, in general,

$$\varphi_i(t) = \frac{g_i(t)}{\sqrt{\int_0^T g_i{}^2(t)dt}} = \frac{g_i(t)}{\sqrt{Eg_i}} \qquad 4.16.14$$

for i = 1, 2,….., N, where

$$g_i(t) = s_i(t) - \sum_{j=1}^{i-1} s_{ij}\varphi_j(t) \qquad 4.16.15$$

and $$s_{ij} = \int_0^T s_i(t).\varphi_j(t)dt \qquad 4.16.16$$

for i = 1, 2,….., N and j = 1, 2, …, M

Let us summarize the steps to determine the orthonormal basis functions following the Gram-Schmidt Orthogonalization procedure:

If the signal set $\{s_j(t)\}$ is known for j = 1, 2,….., M, $0 \le t < T$,

- Derive a subset of linearly independent energy signals, $\{s_i(t)\}$,i = 1, 2,….., N ≤ M.
- Find the energy of $s_1(t)$ as this energy helps in determining the first basis function $\varphi_1(t)$, which is a normalized form of the first signal. Note that the choice of this 'first' signal is arbitrary.
- Find the scalar 's_{21}', energy of the second signal (E_2), a special function '$g_2(t)$' which is orthogonal to the first basis function and then finally the second orthonormal basis function $\varphi_2(t)$
- Follow the same procedure as that of finding the second basis function to obtain the other basis functions.

Concept of signal space

Let, for a convenient set of $\{\varphi_j(t)\}$, j = 1,2,…,N and $0 \le t < T$,

$$s_i(t) = \sum_{j=1}^{N} s_{ij}\varphi_j(t), \quad \text{i= 1, 2,……, M and } 0 \le t < T, \text{ such that,}$$

$$s_{ij} = \int_0^T s_i(t)\varphi_j(t)dt$$

Now, we can represent a signal $s_i(t)$ as a column vector whose elements are the scalar coefficients s_{ij}, j = 1, 2, ….., N :

$$\underline{s_i} = \begin{bmatrix} s_{i1} \\ s_{i2} \\ \vdots \\ s_{iN} \end{bmatrix}_{1\times N} \quad ; \text{i= 1, 2,……, M} \qquad 4.16.17$$

These M energy signals or vectors can be viewed as a set of M points in an N – dimensional Euclidean space, known as the '*Signal Space*' (**Fig.4.16.3**). *Signal Constellation* is the collection of M signals points (or messages) on the signal space.

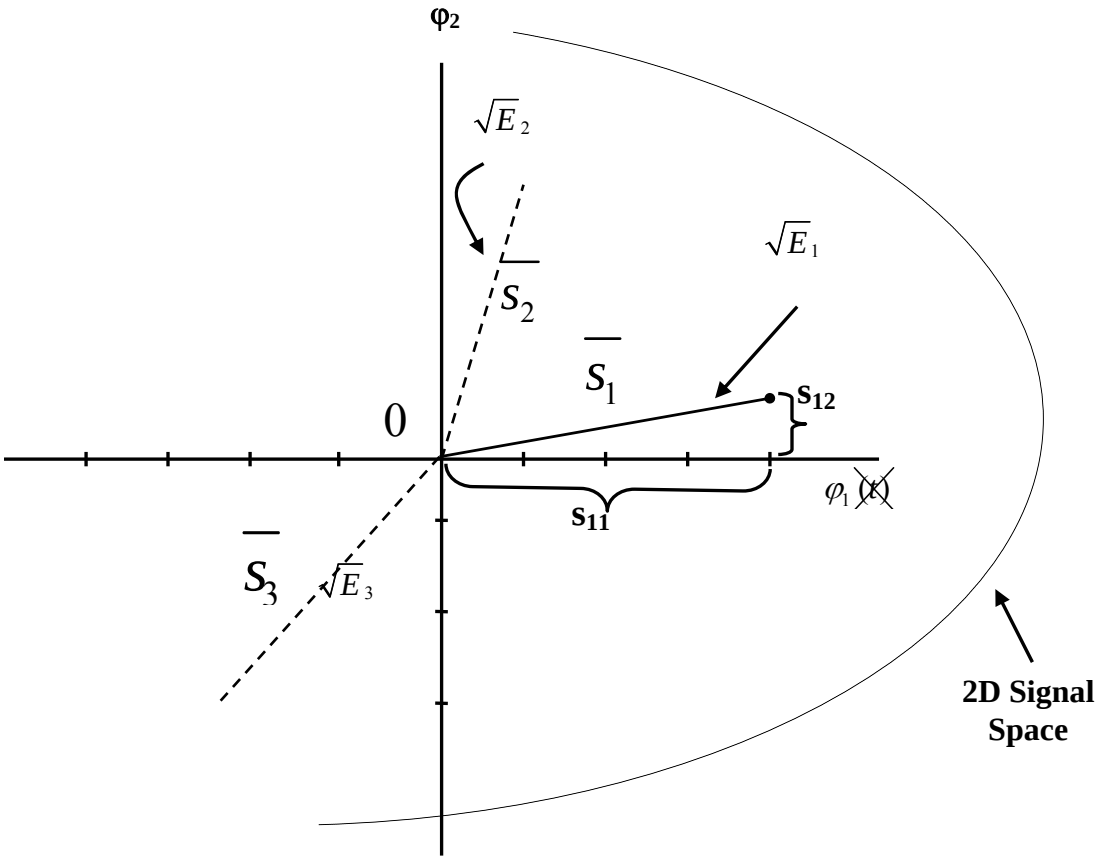


Fig. 4.16.3 *Sketch of a 2-dimentional-signal space showing three signal vectors* $\overline{s_1}$, $\overline{s_2}$ and $\overline{s_3}$

Now, the length or *norm* of a vector is denoted as $\left\|\overline{s_i}\right\|$. The squared norm is the inner product of the vector:

$$\left\|\overline{s_i}\right\|^2 = \left(\overline{s_i}, \overline{s_i}\right) = \sum_{j=1}^{N} s^2{}_{ij} \qquad 4.16.18$$

The cosine of the angle between two vectors is defined as:

$$\cos\left(\text{angle between } \vec{s_i} \ \& \ \vec{s_j}\right) = \frac{\left(\vec{s_i}, \vec{s_j}\right)}{\left\|\vec{s_i}\right\|\left\|\vec{s_j}\right\|} \qquad 4.16.19$$

$\therefore \vec{s_i}$ & $\vec{s_j}$ are orthogonal to each other if $\left(\vec{s_i}, \vec{s_j}\right) = 0$.

If E_i is the energy of the i-th signal vector,

$$E_i = \int_0^T s_i^2(t)\,dt = \int_0^T \left[\sum_{j=1}^{N} s_{ij}\varphi_j(t)\right]\left[\sum_{k=1}^{N} s_{ik}\varphi_k(t)\right] dt$$

$$= \sum_{j=1}^{N}\sum_{k=1}^{N} s_{ij}s_{ik}\int_0^T \varphi_j(t)\varphi_k(t)\,dt \quad \text{as } \{\varphi_j(t)\} \text{ forms an ortho-normal set}$$

$$= \sum_{j=1}^{N} s_{ij}^2 = \left\|\vec{s_i}\right\|^2 \qquad 4.16.20$$

For a pair of signals $s_i(t)$ and $s_k(t)$, $\left\|\vec{s_i}-\vec{s_k}\right\|^2 = \sum_{j=1}^{N}\left(s_{ij}-s_{kj}\right)^2 = \int_0^T \left[s_i(t)-s_k(t)\right]^2 dt$

It may now be guessed intuitively that we should choose $s_i(t)$ and $s_k(t)$ such that the Euclidean distance between them, i.e. $\left\|\vec{s_i}-\vec{s_k}\right\|$ is as much as possible to ensure that their detection is more robust even in presence of noise. For example, if $s_1(t)$ and $s_2(t)$ have same energy E, (i.e. they are equidistance from the origin), then an obvious choice for maximum distance of separation is, $s_1(t) = -s_2(t)$.

Use of Signal Space for Signal Detection in a Receiver

The signal space defined above, is very useful for designing a receiver as well. In a sense, much of the features of a modulation scheme, such as the number of symbols used and the energy carried by the symbols, is embedded in the description of its signal space. So, in absence of any noise, the receiver should detect one of these valid symbols only. However, the received symbols are usually corrupted and once placed in the signal space, they may not match with the valid signal points in some respect or the other. Let us briefly consider the task of a good receiver in such a situation. Let us assume the following:

1. One of the M signals $s_i(t)$, i=1,2,….,M is transmitted in each time slot of duration 'T' sec.
2. All symbols are equally probable, i.e. the probability of occurrence of $s_i(t) = 1/M$, for all 'i'.
3. Additive White Gaussian Noise (AWGN) processes W (t) is assumed with a noise sample function w (t) having mean = 0 and power spectral density $\frac{N_0}{2}$ [N_0: single sided power spectral density of additive white Gaussian noise. Noise is discussed more in next two lessons]
4. Detection is on a symbol-by-symbol basis.

Now, if R (t) denotes the received random process with a sample function r (t), we may write,

$$r(t) = s_i(t) + w(t) \quad ; \qquad 0 \le t < T \quad \text{and} \quad i = 1,2,\ldots,M.$$

The job of the receiver is to make "best estimate" of the transmitted signal $s_i(t)$ (or, equivalently, the corresponding message symbol m_i) upon receiving r(t). We map the received sample function r (t) on the signal space to include a 'received vector' or

'received signal point'. This helps us to identify a noise vector, w(t) ,also. The detection problem can now be stated as:

'*Given an observation / received signal vector ($\bar{r}$), the receiver has to perform a mapping from $\bar{r}$ to an estimate $\hat{m}$ for the transmitted symbol m_i in a way that would minimize the average probability of symbol error'.*

Maximum Likelihood Detection scheme provides a general solution to this problem when the noise is additive and Gaussian. We discuss this important detection scheme in Lesson #19.

Problems

Q4.16.1) Sketch two signals, which are orthonormal to each other over 1 sec. Verify that Eq4.16.3 is valid.

Q4.16.2) Let, $S_1(t) = \text{Cos}2\pi ft$, $S_2(t) = \text{Cos}\,(2\pi ft + \pi/3)$ and $S_3(t) = \text{Sin}\,2\pi ft$. Comment whether the three signals are linearly independent?

Q4.16.3) Consider a binary random sequence of 1 and 0. Draw a signal constellation for the same.

Module 4

Signal Representation and Baseband Processing

Lesson 17

Noise

After reading this lesson, you will learn about

- ***Basic features of Short Noise;***
- ***Thermal (Johnson) Noise;***
- ***Various other forms of Noise;***
- ***Shannon's channel capacity equation and its interpretation;***

As noted earlier, when send some information-bearing signal through a physical channel, the signal undergoes changes in several ways. Some of the ways are the following:

- The signal is usually reduced or attenuated in strength (measured in terms of received power or energy per symbol)
- The signal propagates at a speed comparable to the speed of light, which is high but after all, finite. This means, the channel delays the transmission
- The physical channel may, occasionally introduce additive noise. A transmission cable, for example, may be a source of noise.
- The physical channel may also allow some interfering signals, which are undesired
- The channel itself may have a limitation in bandwidth which may lead to some kind of distortion of the transmitted signal.

Usually the strength of the signal at the receiver input is so low that it needs amplification before any significant signal processing can be carried out. However, the amplifier, while trying to boost the strength of the weak signal, also generates noise within. The power of this noise (and in some other modules down the line such as the frequency down converter in a heterodyne radio receiver) is not negligible. This internally generated noise is always present in a communication receiver. Various mathematical models exist to depict different noise processes that originate in the receiver and affect the transmitted signal. We will consider a simple additive noise model wherein an 'equivalent noise source' will be assumed ahead of a 'noise-less receiver' [n(t) in **Fig. 4.17.1**]. This noise, sometimes referred as 'channel noise', is additive in the sense that the instantaneous noise amplitude is added with the instantaneous amplitude of the received signal.

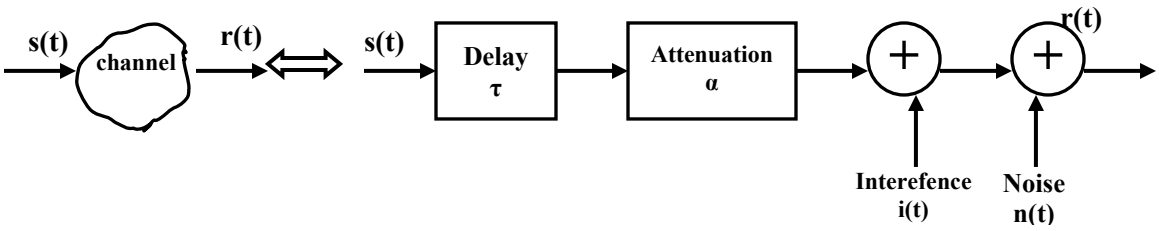


Fig. 4.17.1: *An equivalent model for the physical propagation channel, including the noise generated by the receiver front end*

If s(t) is the transmitted signal and 'α' is the attenuation introduced by the channel, the received signal r(t) can be expressed as,

$$r(t) = \alpha s(t-\tau) + I(t) + n(t) \qquad 4.17.1$$

$I(t)$ represents the interfering signal, if any.

In this lesson, we briefly discuss about the physical features of noise and a short discussion on a baseband channel model for additive white Gaussian noise (AWGN) under certain assumptions.

It is a common knowledge that movable electrons within a passive or active electronic component are responsible for current when excited by external voltage. However, even when no voltage is applied externally, electrons are always in random motion, interacting with other electrons and the material's lattice sites and impurities. The average velocity in any direction remains zero. This statistically random electron motion creates a noise voltage.

Noise is a very important factor that degrades the transmitted signal in a receiver. It is necessary to know the noise level. Two important and relevant forms of noise are, a) *thermal noise* produced by random, thermally produced, motions of carriers in metals and semiconductors and b) *shot noise* produced by 'particle-like' behavior of electrons and photons when an external excitation is available to produce current. Shot noise is avoidable only if we reduce all current to zero.

Shot Noise

Let us consider a 'steady' or dc electric current I between two points A and B with each electron carrying a charge 'q'. On an average, the number of charges moving from A to B during time 't' is

$$n_{av} = \frac{I.t}{q} \qquad 4.17.2$$

Now, at the microscopic level, the electrons do not move in a perfectly regular fashion. The rate of flow varies unpredictably within short spans of time. This means that the instantaneous current is usually different from 'I'. This fluctuation around a nominal average value of 'I' is modeled as a noise current (i_n). It has been established that the observed mean squared value of this fluctuating current is,

$$E\left[i_n^2\right] = 2.q.I.B \qquad 4.17.3$$

where B is the bandwidth of the system used for measurement. Interestingly, the mean squared value of the noise current is proportional to the gross current 'I'. So, if the average (bias) current in a photo detector is high, there is a possibility of considerable shot noise. This is an important issue in the design of optical detectors in fiber optic communication. Shot noise in optical devices is widely called as 'quantum noise'. Low noise active electronic amplifiers for wireless receivers are intelligently designed to suppress the shot noise by electrostatic repulsion of charge carriers.

Shot noise is closely described and modeled as a Poisson process. The charge carriers responsible for the shot noise follow Poisson distribution [Lesson #7]. Analytically, the noise power may be obtained from the Fourier transform of the auto-correlation of this random process.

Thermal Noise (also known as Johnson-Nyquist noise and Johnson noise) :

Thermal noise is generated by the equilibrium fluctuations of the carriers in electronic components, even in absence of an applied voltage. It originates due to random thermal motion of the charge carriers. It was first measured by J. B. Johnson in 1928 and theoretically established by H. Nyquist through a fluctuation – dissipation relationship of statistical thermodynamics. Thermal noise is different from shot noise, which is due to current fluctuations that occur only when a macroscopic current exists.

The thermal noise power P, in watts, is given by $P = 4kT\Delta f$, where k is Boltzmann's Constant [k = 1.380 6505(24) × 10^{-23} J/K], T is the component temperature in Kelvin and Δf is the bandwidth in Hz. Thermal noise power spectral density, Watt per Hz, is constant throughout the frequency spectrum of interest (typically upto 300 GHz). It depends only on k and T. That is why thermal noise is often said to be a white noise in the context of radio communication.
A quick and good estimate of thermal noise, in dBm [0 dBm = 1 mWatt], at room temperature (about 27^0C) is:

$$P = -174 + 10\log(\Delta f) \qquad 4.17.4$$

A quick calculation reveals that the total noise power in a receiver, with a bandwidth of 1 MHz and equivalent noise temperature of 27^0C, may be about -114 dBm.
The thermal noise voltage, v_n, that is expected across an 'R' Ohm resistor at an absolute temperature of 'T' K is given by:

$$v_n = \sqrt{4kT\Delta f} \qquad 4.17.5$$

So, thermal noise in a receiver can be made very low by cooling the receiver subsystems, which is a costly proposition.

Colour of noise

Several possible forms of noise with various frequency characteristics are some times named in terms of colors. It is assumed that such noise has components at all frequencies, with a spectral density proportional to $\frac{1}{f^{\alpha}}$.

White noise

It is a noise process with a flat spectrum vs. frequency, i.e. with same power spectral density, $\frac{N_o}{2}$ W/Hz. This means, a 1 KHz frequency range between 2 KHz and 3KHz contains the same amount of power as the range between 2 MHz and 2.001 MHz. Let us note here that the concept of an infinite-bandwidth white noise is only theoretical as the noise power is after all, finite in a physically realizable receiver. The additive Gaussian noise process is white.

Pink noise [flicker noise, 1/f noise]

The frequency spectrum of flicker noise is flat in logarithmic space, i.e., it has same power in frequency bands that are proportionally wide. For example, flicker noise in a system will manifest equal power in the range from 30 to 50 Hz and in the band from 3KHz to 5KHz. Interestingly, the human auditory system perceives approximately equal magnitude on all frequencies.

Brown noise

Similar to pink noise, but with a power density decrease of 6 dB per octave with increasing frequency (density proportional to $\frac{1}{f^2}$) over a frequency range which does not include DC. It can be generated by simulating Brownian motion and by integration*Blue noise:* Power spectral density of blue noise increases 3 dB per octave with increasing frequency ($\alpha = -1$) over a finite frequency range. This kind of noise is sometimes useful for dithering.

Shannon's Channel Capacity Equation

The amount of noise present in the receiver can be represented in terms of its power $N = \frac{v_n^2}{R_{ch}}$, where R_{ch} is the characteristic impedance of the channel, as seen by the receiver and v_n is the rms noise voltage. Similarly, the message bearing signal can be represented by its power we can represent a typical message in terms of its average signal power $S = \frac{v_s^2}{R_{ch}}$, where v_s is the rms voltage of the signal. Now, it is reasonable to assume that the signal and noise are uncorrelated i.e., they are not related in any way and we cannot predict one from the other. If 'P_r' is the total power received due to the combination of signal and noise, which are uncorrelated random processes, we can write $v_r^2 = v_s^2 + v_n^2$, i.e.,

$$P_r = S + N \qquad 4.17.6$$

Now, let the received signal with rms voltage 'v_s' contain 'b' bits of information per unit time and noise with rms voltage 'v_n'. If, for the sake of simplicity, we decide to sample the received signal once per unit time, we can hope to recover the b bits of information correctly from the received signal sample by adopting the following strategy:

We quantize the sample in a manner such that the noise is not likely to make our decision about b-bits of information wrong. This is achievable if we adopt a b-bit quantizer(i.e. 2^b quantizer levels) and the noise sample voltage is less than half the step size. The idea then, is simply to read the quantizer output as the received b-bit information. So, the limiting condition may be stated as:

$2^b = \frac{V_{r\,max}}{V_{n\,max}}$, where $V_{r\,max}$ is the maximum allowable received signal amplitude and $V_{n\,max}$ is the maximum allowable noise amplitude. With this quantizer, our decision will be correct when

$2^b \geq \frac{V_r}{V_n}$ and our decision will be erroneous if $2^b \leq \frac{V_r}{V_n}$. So, the limiting condition for extracting b-bits of information from noise-corrupted received signal is,

$$2^b = \frac{V_r}{V_n} \qquad 4.17.7$$

Now, we can write,

$$2^b = \frac{V_r}{V_n} = \sqrt{\frac{v_r^2}{v_n^2}} = \sqrt{\frac{v_s^2 + v_n^2}{v_n^2}} = \sqrt{1 + \left(\frac{S}{N}\right)} \qquad 4.17.8$$

Or, equivalently, $$\log_2\left(\sqrt{1 + \left(\frac{S}{N}\right)}\right) \qquad 4.17.9$$

Now, from Nyquist's sampling theorem, we know that, for a signal of bandwidth 'B', the maximum number of such samples that can be obtained per unit time is 2B and hence, the maximum amount of information (in bits) that can be obtained per unit time, is,

$$I_{max} = 2Bb = 2B \log_2\left(\sqrt{1 + \left(\frac{S}{N}\right)}\right) = 2B \log_2\left(1 + \frac{S}{N}\right). \qquad 4.17.10$$

Eq. 4.17.10 is popularly expressed as,

$$C = B \log_2\left(1 + \frac{S}{N}\right) \qquad 4.17.11$$

'C' indicates the 'capacity of the waveform channel', i.e. the maximum amount of information that can be transmitted through a channel with bandwidth 'B' and enjoying signal-to-noise ratio of S/N. Eq. 4.17.11 is popularly known as *Shannon-Hartley Channel Capacity Equation* for additive white Gaussian noise waveform channel.

Interpretation of Shannon-Hartley Channel Capacity Equation

a) We observe that the capacity of a channel can be increased by either i) increasing the channel bandwidth or ii) increasing the signal power or iii) reducing the in-band noise power or iv) any judicious combination of the three. Each approach in practice has its own merits and demerits. It is indeed, interesting to note that, all practical digital communication systems, designed so far, operate far below the capacity promised by Shannon-Hartley equation and utilizes only a fraction of the capacity. There are multiple yet interesting reasons for this. One of the overriding requirements in a practical system is sustained and reliable performance within the regulations in force. However, advances in coding theory (especially turbo coding), signal processing techniques and VLSI techniques are now making it feasible to push the operating point closer to the Shannon limit.

b) If, B → ∞, we apparently have infinite capacity but it is not true. As B → ∞, the in-band noise power, N also tends to infinity [N = N_o.B, N_o : single-sided noise power spectral density, a constant for AWGN] and hence, S/N → 0 for any finite signal power 'S' and $\log_2\left(1+\frac{S}{N}\right)$ also tends to zero. So, it needs some more careful interpretation and we can expect an asymptotic limit.

At capacity, the bit rate of transmission $R_b = C$ and the duration of a bit = $T_b = \frac{1}{R_b} = \frac{1}{C}$. If the energy received per information bit is E_b, the signal power S can be expressed as, S = energy received per unit time = $E_b.R_b = E_b.C$. So, the signal-to-noise ratio $\frac{S}{N}$ can be expressed as,

$$\frac{S}{N} = \frac{E_b C}{N_0 B} \qquad 4.17.12$$

Now, from Eq. 4.17.11, we can write,

$$\frac{C}{B} = \log_2\left(1+\frac{E_b C}{N_0 B}\right) \qquad 4.17.13$$

This implies,

$$\frac{E_b}{N_0} = \frac{B}{C}\left(2^{\frac{C}{B}} - 1\right)$$

$$\cong \frac{B}{C}\left[\left(1+\frac{C}{B}\ln 2\right) - 1\right], \quad \text{for } B >> C$$

$$= \log_e 2, \quad \text{for } B >> C$$

$$= -1.6 \text{ dB} \qquad 4.17.14$$

So, the limiting $\frac{E_b}{N_0}$, in dB is -1.6 dB. So, ideally, a system designer can expect to achieve almost errorless transmission only when the $\frac{E_b}{N_0}$ is more than -1.6 dB and there is no constraint in bandwidth.

c) In the above observation, we set $R_b = C$ to appreciate the limit in $\frac{E_b}{N_0}$ and we also saw that if $R_b > C$, the noise v_n is capable of distorting the group of 'b' information bits. We say that the bit rate has exceeded the capacity of the channel and hence errors are not controllable by any means.

To reiterate, all practical systems obey the inequality $R_b < C$ and most of the civilian digital transmission systems utilize the available bandwidth efficiently, which means B (in Hz) and C (in bits per second) are comparable. For bandwidth efficient transmission, the strategy is to

increase the bandwidth factor $\frac{R_b}{B}$ while $R_b < C$. This is achieved by adopting suitable modulation and reception strategies, some of which will be discussed in Module #5.

Problems

Q4. 17.1) Name two passive electronic components, which may produce noise.

Q4. 17.2) If a resistor generates 1 nano-Watt/Hz, determine the temperature of the resistor.

Q4. 17.3) Determine the capacity of a waveform channel whose bandwidth is 10 MHz and signal to noise rotation is 10dB.

Module 4

Signal Representation and Baseband Processing

Lesson 18

Response of Linear System to Random Processes

After reading this lesson, you will learn about

- ***Modeling of thermal noise and power spectral density;***
- ***Time domain analysis of a linear filter for random input;***
- ***Representation of narrow-band Gaussian noise;***
- ***Low-pass equivalent components of narrow-band noise;***
- ***Band-pass Gaussian noise and its spectral density;***

A noise waveform is a sample function of a random process. Thermal noise is expected to manifest in a communication receiver for an infinite time and hence theoretically noise may have infinite energy. Thermal noise is typically modeled as a power signal. Usually, some statistical properties of thermal noise, such as its mean, variance, auto – correlation function and power spectrum are of interest.

Thermal noise is further modeled as a wide-sense-stationary (WSS) stochastic process. That is, if n(t) is a sample function of noise, a) the sample mean of $n(t_1)$, i.e. n(t) at $t = t_1$, is independent of the choice of sampling instant 't_1' and b) the correlation of two random samples, $n(t_1)$ and $n(t_2)$ depends only on the interval / delay (t_2-t_1), i.e., $E[n(t_1)n(t_2)] = R_n(t_2 - t_1) = R_n(\tau)$

The auto-correlation function (ACF), $R_x(\tau)$ of a WSS process, x(t) is defined as: $ACF = R_x(\tau) = E[x(t)x(t+\tau)]$. $R_x(\tau)$ indicates the extent to which two random variables separated in time by 'τ' vary with each other. Note that, $R_x(0) = E[x(t)x(t)] = \overline{x^2}$, the mean square of x (t).

Power Spectral Density (psd)

a. Specifies distribution of power of the random process over frequency 'f'. If $S_x(f)$ is the two-sided psd of x(t), the power in a small frequency band Δf at f_1 is $[S_x(f_1).\ \Delta f]$;

b. psd $S_x(f)$ of thermal noise is a real, positive even function of frequency. The power in a band f_1 to f_2 is:

$$\int_{-f_2}^{-f_1} S_x(f)df + \int_{f_1}^{f_2} S_x(f)df = 2\int_{f_1}^{f_2} S_x(f)df\ ; \qquad \text{in Volt}^2/\text{Hz}$$

For a deterministic waveform, the psd and ACF form a Fourier transform pair. The concept is extended to random processes and we may write for thermal noise process,

$$S_x(f) = F\left[R_x(\tau)\right] = \int_{-\alpha}^{\alpha} R_x(\tau)e^{-j\varpi\tau}d\tau \qquad 4.18.1$$

$$\text{and} \quad R_x(\tau) = F^{-1}\left[S_x(f)\right] = \int_{-\alpha}^{\alpha} S_x(\tau)e^{j\varpi\tau}df$$

Now, as noted in Lesson #17, the psd for white noise is constant:

$$S_n(f) = \frac{N_0}{2} \qquad 4.18.2$$

Hence, the ACF for such noise process is,

$$R_n(\tau) = F^{-1}\left\{\frac{N_0}{2}\right\} = \frac{N_0}{2}.\delta(\tau) \qquad 4.18.3$$

As we know, the signal, carrying information, occupies a specific frequency band and it is sufficient to consider the effect of noise, which manifests within this frequency band. So, it is useful to study the features of 'band-limited noise'. For a base band additive white Gaussian noise (AWGN) channel of bandwidth 'W' Hz,

$$S_n(f) = \frac{N_0}{2} \;,\; |f| < W$$
$$= 0 \;, \quad \text{Elsewhere} \qquad 4.18.4$$

Simple calculation now shows that, the auto-correlation function for this base band noise is:

$$R_n(\tau) = WN_0 \sin c(2W\tau) \qquad 4.18.5$$

For pass-band thermal noise of bandwidth 'B' around a centre frequency f_c, the results can be extended:

$$S_n(f) = \frac{N_0}{2} \;,\; |f - f_c| < \frac{B}{2}$$
$$= 0 \quad , \quad \text{Otherwise} \qquad 4.18.6$$

The ACF now is given by, $R_n(\tau) = BN_0(\sin cB\tau).\cos 2\pi f_c\tau \qquad 4.18.7$

In many situations, it is necessary to analyze the characteristics of a noise process at the output of a linear system, which transforms some excitation given at its input. This is important because, the system being linear in nature, obeys the principle of superposition and if we excite the system with a noise process and analyze the response noise process, we can use this knowledge for multiple situations. For example, a specific case of interest may be to analyze the output of a linear filter when a noisy received signal is fed to it. For simplicity, we discuss about response of linear systems which are time-invariant. Though such analysis is more elegant when carried out in the frequency domain, we start with a time-domain analysis to provide some insight.

Time-domain analysis for random input to a linear filter

Let us consider a linear lowpass filter whose impulse response is h(t) and let us excite the filter with white Gaussian noise. The input being a random process, it is not so important to get only an expression for the filter output y(t). It is statistically more significant to obtain expressions for the mean, variance, ACF and other parameters of the output signal. Now, in general, if x(t) indicates the input to a linear system, the mean of the output y(t)

is, $\bar{y} = \bar{x}\int_0^{\alpha} h(t)dt$, where $\bar{x}$ is the mean of the input process. The mean square value of the output is: $\overline{y^2} = \int_0^{\alpha}\int_0^{\alpha} R_x(\lambda_2 - \lambda_1)h(\lambda_1)h(\lambda_2)d\lambda_2 d\lambda_1$

When the input is white noise, we know

$$R_n(\tau) = \frac{N_0}{2}\delta(\tau), \quad \text{and} \quad \bar{y} = 0 \qquad 4.18.8$$

So, the mean square of the output noise process is:

$$\overline{y^2} = \int_0^{\alpha}\int_0^{\alpha} \frac{N_0}{2}\delta(\lambda_2 - \lambda_1)h(\lambda_1)h(\lambda_2)d\lambda_2 d\lambda_1$$

$$= \frac{N_0}{2}\int_0^{\alpha} h^2(\lambda)d\lambda \qquad 4.18.9$$

Ex 4.18.1: Let us consider a single-stage passive R-C lowpass filter whose impulse response is well known:

$h(t) = \alpha e^{-\alpha t}u(t)$, where $\alpha = \frac{1}{\text{R.C}}$. The 3 dB cutoff frequency of the filter is $f_{cutoff} = \frac{1}{2\pi RC}$. It is straight forward to see that the average of noise at the output of this low-pass filter is zero: $\bar{y} = \bar{x}\int_0^{\alpha} \alpha e^{-\alpha t}d\lambda = \bar{n} = 0$

Further, $\overline{y^2} = \frac{N_0}{2}\int_0^{\alpha} \alpha^2 e^{-2\alpha\lambda}d\lambda = \frac{\alpha N_0}{4} = \frac{N_0}{4RC} = \frac{\pi}{2}\times N_0 \times f_{cutoff}$

We observe that $\overline{y^2}$, the noise power at the output of the filter, is proportional to the filter BW.

Auto-correlation function (ACF)

In general, the autocorrelation of a random process at the output of a linear two-port network is:

$$R_y(\tau) = \int_0^{\alpha}\int_0^{\alpha} R_x(\lambda_2 - \lambda_1 - \tau)h(\lambda_1).h(\lambda_2)d\lambda_2 d\lambda_1 \qquad 4.18.10$$

Specifically, for white noise,

$$R_y(\tau) = \int_0^{\alpha}\int_0^{\alpha} \frac{N_0}{2}\delta(\lambda_2 - \lambda_1 - \tau)h(\lambda_1).h(\lambda_2)d\lambda_2 d\lambda_1$$

$$= \frac{N_0}{2}\int_0^{\alpha} h(\lambda_1).h(\lambda_1+\tau)d\lambda_1 \qquad 4.18.11$$

Considering the RC lowpass filter of *Ex #4.18.1,* we see,

$$R_y(\tau) = \frac{N_0}{2}\int_0^{\alpha} \alpha e^{-\alpha\lambda}\alpha e^{-\alpha(\lambda+\tau)}d\lambda$$

$$= \frac{\alpha N_0}{4}e^{-\alpha\tau}\ , \tau \geq 0$$

As, R(τ) is an even function , R(τ) = R(- τ) and hence,

$$R_y(\tau) = \frac{\alpha N_0}{4}e^{-\alpha|\tau|} \qquad 4.18.12$$

This is an exponentially decaying function of τ with a peak value of $\frac{\alpha N_o}{4}$ at τ = 0.

As $R_y(\tau)$ and the power spectrum $S_y(\omega)$ are Fourier transform pair, we see that the power spectrum of the output noise is:

$$S_y(\omega) = \frac{N_0}{2}\left(\frac{\alpha^2}{\omega^2+\alpha^2}\right) \qquad 4.18.13$$

Representation of Narrow-band Gaussian Noise

Representation and analysis of narrow pass band noise is of fundamental importance in developing insight into various carrier-modulated digital modulation schemes which are discussed in Module #5. The following discussion is specifically relevant for narrowband digital transmission schemes.

Let, x(t) denote a zero-mean Gaussian noise process band-limited to ± B/2 around centre frequency 'f_c'. There are several ways of analyzing such narrowband noise process and we choose an easy-to-visualize approach, which somewhat approximate. To start with, we consider a sample of a noise process over a finite time interval and apply Fourier series expansion while stretching the time interval to ∞. If x(t) is observed over an interval $-\frac{T}{2} \leq t \leq \frac{T}{2}$, we may write

$$x(t) = \sum_{n=1}^{\alpha}(x_{cn}\cos nw_0t + x_{sn}\sin w_0t) \qquad \text{where,} \quad w_0 = \frac{2\pi}{T}$$

and $x_{cn} = \frac{2}{T}\int_{-T/2}^{T/2} x(t)\cos nw_0t\,dt$, n = 1,2,……

and $x_{sn} = \frac{2}{T}\int_{-T/2}^{T/2} x(t)\sin nw_0t\,dt$, n = 1,2,…..

4.18.14

It can be shown that x_{cn} and x_{sn} are Gaussian random variables.
The centre frequency 'f_c' can now be brought in by the following substitution:

$$nw_0 = (nw_0 - w_c) + w_c$$

$$\therefore \quad x(t) = \sum_{n=1}^{\alpha} \{x_{cn} \cos[(nw_0 - w_c)t + w_c t] + x_{sn} \sin[(nw_0 - w_c)t + w_c t]\}$$

$$\equiv x_c(t) \cos w_c t - x_s(t) \sin w_c t \; ; \qquad 4.18.15$$

where, $x_c(t) = \sum_{n=1}^{\alpha} x_{cn} \cos(nw_0 - w_c)t + x_{sn} \sin(nw_0 - w_c)t$ 4.18.16

and $x_s(t) = \sum_{n=1}^{\alpha} x_{cn} \cos(nw_0 - w_c)t + x_{sn} \sin(nw_0 - w_c)t$ 4.18.17

Another elegant and equivalent expression for x(t) is:

$$x(t) = x_c(t) \cos w_c t - x_s(t) \sin w_c t$$

$$= r(t) \cos[w_c t + \Phi(t)] \qquad 4.18.18$$

where, $r(t) = \sqrt{x_c^{\,2}(t) + x_s^{\,2}(t)}$;

and $\Phi(t) = \tan^{-1}\left[\frac{x_s(t)}{x_c(t)}\right]$; $0 \le \Phi(t) < 2\pi$

It is easy to recognize that, $x_c(t) = r(t)\cos\Phi(t)$ and $x_s(t) = r(t)\sin\Phi(t)$

Low pass equivalent components of narrow band noise

Let, x_{ct} and x_{st} represent samples
of $x_c(t)$ and $x_s(t)$. These are Gaussian distributed random variables with zero mean as the original noise process has zero mean.

$$\therefore E[x_{ct}] = E[x_{st}] = 0 \qquad 4.18.19$$

Now, an expression for variance of x_{ct}, by definition, looks like the following:

$$E[x_{ct}^{\,2}] = E\left[\sum_{n=1}^{\alpha}\sum_{m=1}^{\alpha}[x_{cn}\cos(nw_0 - w_c)t + x_{sn}\sin(nw_0 - w_c)t]\times [x_{cm}\cos(mw_0 - w_c)t + x_{sm}\sin(mw_0 - w_c)t]\right] \qquad 4.18.20$$

However, after some manipulation, the above expression can be put in the following form:

$$E[x_{ct}^{\,2}]$$

$$= \left\{\sum_{n=1}^{\alpha}\sum_{m=1}^{\alpha} \begin{matrix} [\overline{x_{cn}.x_{cm}}\cos(nw_0 - w_c)t.\cos(mw_0 - w_c)t + \overline{x_{cn}.x_{sm}}\cos(nw_0 - w_c)t. \\ \sin(mw_0 - w_c)t + \ldots + \overline{x_{sn}.x_{cm}}\sin(nw_0 - w_c)t.\cos(mw_0 - w_c)t \\ + \overline{x_{sn}.x_{sm}}\sin(nw_0 - w_c)t.\sin(mw_0 - w_c)t] \end{matrix}\right\}$$

4.18.21

Here,

$$\overline{x_{cn}x_{cm}} = E\left[\left(\frac{2}{T}\int_{-T/2}^{T/2} x(t).\cos nw_0 t dt\right).\left(\frac{2}{T}\int_{-T/2}^{T/2} x(t).\cos mw_0 t dt\right)\right]$$

$$= \frac{4}{T^2}\int_{-T/2}^{T/2}\int_{-T/2}^{T/2} \overline{x(t_1)x(t_2)}\cos nw_0 t_1.\cos mw_0 t_2 dt_1 dt_2$$

$$= \frac{4}{T^2}\int_{-T/2}^{T/2}\int_{-T/2}^{T/2} R_x(t_2 - t_1)\cos nw_0 t_1.\cos mw_0 t_2 dt_1 dt_2 \qquad 4.18.22$$

$R_x(t_2\text{-}t_1)$ in the above expression is the auto-correlation of the noise process x(t). Now, putting $t_2 - t_1 = u$ and $\frac{t_1}{T} = v$, we get,

$$\overline{x_{cn}x_{cm}} = \frac{4}{T}\int_{-1/2}^{1/2} \cos nw_0 Tv\left\{\int_{-T(\frac{1}{2}+v)}^{T(\frac{1}{2}+v)} R_x(u).\cos mw_0(u + v.T)du\right\}dv$$

$$= \frac{4}{T}\int_{-1/2}^{1/2} \cos 2\pi nv\left[\cos 2\pi mv\int_{-T(\frac{1}{2}+v)}^{T(\frac{1}{2}+v)} R_x(u).\cos\frac{2\pi mu}{T}du\right.$$

$$\left. - \sin 2\pi mv\int_{-T(\frac{1}{2}+v)}^{T(\frac{1}{2}+v)} R_x(u).\sin\frac{2\pi mu}{T}du\right]dv \qquad 4.18.23$$

Now for $T \to \infty$ and putting $f_m = \frac{m}{T}$, where f_m tends to 0 for all 'm', the inner integrands are:

$$\lim_{T\to\alpha}\int_{-T(\frac{1}{2}+v)}^{T(\frac{1}{2}+v)} R_x(u).\cos 2\pi f_m u du = S_x(f_m)\ \text{, say}$$

and $$\lim_{T\to\alpha}\int_{-T(\frac{1}{2}+v)}^{T(\frac{1}{2}+v)} R_x(u).\sin 2\pi f_m u du = 0$$

Let us choose to write, $\lim_{T\to\alpha} f_m\left(=\frac{m}{T}\right) = f \to 0$

Now from Eq. 4.18.23, we get a cleaner expression in the limit:

$$\lim_{T\to\alpha} T\overline{x_{cn}x_{cm}} = 4S_x(f)\int_{-1/2}^{1/2}\cos 2\pi nv\cos 2\pi mv\,dv$$
$$= 2S_x(f), \quad m = n$$
$$= 0, \quad m \neq n \qquad 4.18.24$$

Following similar procedure as outlined above, it can be shown that,

$$\lim_{T\to\alpha} T\overline{x_{sn}x_{sm}} = 2S_x(f), \quad m = n$$
$$= 0, \quad m \neq n \qquad 4.18.25$$

and $\lim_{T\to\alpha} T\overline{x_{cn}x_{sm}} = 0$, all m, n 4.18.26

Eq. 4.18.24 – 26 establish that the coefficients x_c-s and x_s-s are uncorrelated as T approaches ∞.

Now, referring back to Eq. 4.18.21, we can see that,

$$E\left[x_{ct}^2\right] = \lim_{T\to\alpha}\sum_{n=1}^{\alpha} x_{cn}{}^2\left[\cos^2(nw_0 - w_c)t + \sin^2(nw_0 - w_c)t\right] \qquad 4.18.27$$

$$= \lim_{T\to\alpha}\sum_{n=1}^{\alpha} S_x(f_m)\left(\frac{2}{T}\right) = 2\int_0^{\alpha} S_x(f)df = \overline{x_t^2} \qquad 4.18.28$$

Here $\overline{x_t^2}$ denotes the mean square value of x(t).

Similarly, it can be shown that,

$$E\left[x_{st}{}^2\right] = E\left[x_{ct}{}^2\right] = \overline{x_t{}^2} \qquad 4.18.29$$

Since, $\overline{x_{st}} = \overline{x_{ct}} = 0$, we finally get, $\sigma_{st}{}^2 = \sigma_{ct}{}^2 = \sigma_x{}^2$, the variance of x(t).

It may also be shown that the covariance of x_{ct} and x_{st} approach 0 as T approaches ∞. Therefore, ultimately it can be shown that x_{ct} and x_{st} are statistically independent.

So, x_{ct} and x_{st} are uncorrelated Gaussian distributed random variables and they are statistically independent. They have zero mean and a variance equal to the variance of the original bandpass noise process**.** This is an important observation. 'x_{ct}' is called the in-phase component and 'x_{st}' is called the quadrature component of the noise process.

Spectral Density of In-phase and Quadrature Component of Bandpass Gaussian Noise

Following similar procedures as adopted for determining mean square values of $x_c(t)$ and $x_s(t)$, we can compute their auto-correlation and cross-correlation functions as below:

ACF of $x_{ct} = R_{x_c}(\tau) = 2\int_0^{\alpha} S_x(f).cos2\pi(f - f_c)\tau df$ 4.18.30

$R_{x_s}(\tau) = \text{ACF of } x_{st} = 2\int_0^{\alpha} S_x(f).cos2\pi(f - f_c)\tau df = R_{xc}(\tau)$ 4.18.31

Cross Correlation Function (CCF) between x_{ct} and x_{st} is:

$$R_{x_c x_s}(\tau) = 2\int_0^{\alpha} S_x(f).\sin 2\pi(f - f_c)\tau df \quad 4.18.32$$

and $$R_{x_c x_s}(\tau) = -R_{x_c x_s}(\tau) \quad 4.18.33$$

Eq.4.18.30 can be expressed in the following convenient manner:

$$\begin{aligned} R_{x_c}(\tau) &= \int_0^{\alpha} S_x(f)\cos 2\pi(f - f_c)\tau df + \int_0^{-\alpha} S_x(-f)\cos 2\pi(-f - f_c)\tau(-df) \\ &= \int_{-f_c}^{\alpha} S_x(f + f_c)\cos 2\pi f\tau df + \int_0^{f_c} S_x(f - f_c)\cos 2\pi f\tau df \\ &= \int_{-f_c}^{f_c} \left[S_x(f + f_c) + S_x(f - f_c)\right]\cos 2\pi f\tau df \end{aligned} \quad 4.18.34$$

From this, using inverse Fourier Transform, one gets,

$$S_{x_c}(f) = S_x(f + f_c) + S_x(f - f_c) \quad 4.18.35$$

Moreover, $$S_{x_c}(f) = S_{x_s}(f) = S_x(f + f_c) + S_x(f - f_c) \quad 4.18.36$$

Note that the power spectral density of $x_c(t)$ is the sum of the negative and positive frequency components of $S_x(f)$ after their translation to the origin.

The following steps summarize the method to construct psd of $S_{x_c}(f)$ or $S_{x_s}(f)$ from $S_x(f)$:

a. Displace the +ve frequency portion of the plots of $S_x(f)$ to the left by 'f_c'.
b. Displace the –ve frequency portion of $S_x(f)$ to the right by 'f_c'.
c. Add the two displaced plots.

If 'f_c' is not the centre frequency, the psd of $x_c(t)$ or $x_s(t)$ may be significantly different from what may be guessed intuitively.

Problems

Q4.18.1) Consider a pass band thermal noise of bandwidth 10 MHz around a center frequency of 900 MHz. Sketch the auto co-relation function of this pass band thermal noise normalized to its PSD.

Q4.18.2) Sketch the pdf of typical narrow band thermal noise.

Module 4

Signal Representation and Baseband Processing

Lesson 19

Maximum Likelihood Detection and Correlation Receiver

After reading this lesson, you will learn about

- ***Principle of Maximum Likelihood (ML) detection;***
- ***Likelihood function;***
- ***Correlation receiver;***
- ***Vector receiver;***

Maximum likelihood (ML) detection:

We start with the following assumptions:

- Number of information-bearing signals (symbols), designed after the G-S-O approach, is 'M' and one of these 'M' signals is received from the AWGN channel in each time slot of 'T'-sec. Let the messages be denoted by m_i, i = 1, 2, …, M. Each message, as discussed in an earlier lesson, may be represented by a group of bits, e.g. by a group of 'm' bits each such that $2^m = M$.

- All symbols are equi- probable. This is not a grave assumption since, if the input message probabilities are different and known, they can be incorporated following Bayesian approach. However, for a bandwidth efficient transmission scheme, as is often needed in wireless systems, the source coding operation should be emphasized to ensure that all the symbols are independent and equally likely. Alternatively, the number of symbols, M, may also be decided appropriately to approach this desirable condition.

- AWGN process with a mean = 0 and double-sided psd No/2. Let w(t) denote a noise sample function over $0 \le t < T$.

Let 'R(t)' denote the received random process with sample function over a symbol duration denoted as r(t), $0 \le t \le T$. Now, a received sample function can be expressed in terms of the corresponding transmitted information-bearing symbol, say $s_i(t)$, and a sample function w(t) of the Gaussian noise process simply as:

$$r(t) = s_i(t) + w(t), \quad 0 \le t < T \qquad 4.19.1$$

At the receiver, we do not know which $s_i(t)$ has been transmitted over the interval $0 \le t < T$. So, the job of an efficient receiver is to make 'best estimate' of transmitted signal $[s_i(t)]$ upon receiving r(t) and to repeat the same process during all successive symbol intervals. This problem can be explained nicely using the concept of '*signal space*', introduced earlier in Lesson #16. Depending on the modulation and transmission strategy, the receiver usually has the knowledge about the signal constellation that is in use. This also means that the receiver knows all the nominal basis functions used by the transmitter. For convenience, we will mostly consider a transmission strategy involving two basis functions, $\boldsymbol{\varphi_1}$ and $\boldsymbol{\varphi_2}$ (described now as unit vectors) for explanation though most of the discussion will hold for any number of basis functions. **Fig.4.19.1** shows a two-dimensional signal space showing a signal vector $\overline{s_i}$ and a received vector $\overline{r}$. Note the noise vector $\overline{\omega}$. as well.

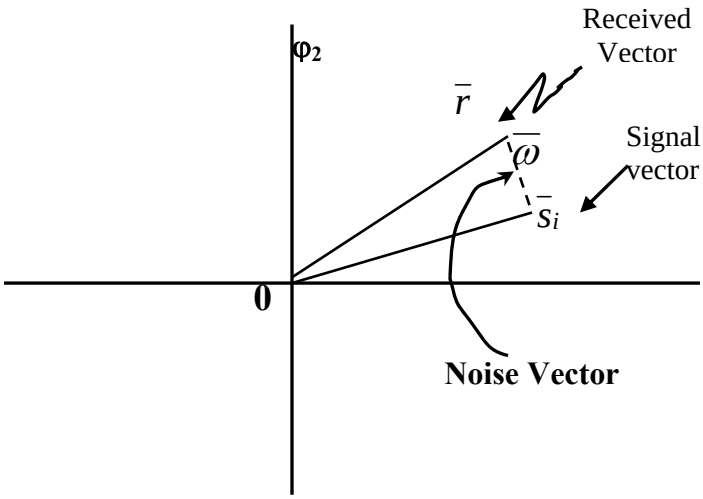


Fig. 4. 19.1 *Signal space showing a signal vector* $\bar{s}_i$ *and a received vector* $\bar{r}$

The job of the receiver can now be formally restated as: Given received signal vectors $\bar{r}$, find estimates $\hat{m}_i$ for all valid transmit symbols 'm_i-s' once in each symbol duration in a way that would minimize the probability of erroneous decision of a symbol on an average (continuous transmission of symbols is implicit).

The principle of Maximum Likelihood (ML) detection provides a general solution to this problem and leads naturally to the structure of an optimum receiver. When the receiver takes a decision that $\hat{m} = m_i$, the associated probability of symbol decision error may be expressed as: $Pe\left(m_i, \bar{r}\right)$ = probability of decision on receiving $\bar{r}$ that 'm_i' was transmitted = Pr (m_i not sent $\left|\bar{r}\right.$) = 1 – Pr (m_i sent $\left|\bar{r}\right.$).

In the above, Pr (m_i not sent $\left|\bar{r}\right.$) denotes the probability that 'm_i' was not transmitted while $\bar{r}$ is received. So, an optimum decision rule may heuristically be framed as:

Set $\hat{m} = m_i$ if Pr (m_i sent $\left|\bar{r}\right.$) ≥ Pr (m_k sent $\left|\bar{r}\right.$), for all $k \neq i$ 4.19.2

This decision rule is known as *maximum a posteriori probability* rule. This rule requires the receiver to determine the probability of transmission of a message from the received vector. Now, for practical convenience, we invoke Bayes' rule to obtain an equivalent statement of optimum decision rule in terms of a priori probability:

$$\underbrace{Pr\left(m_i\left|\bar{r}\right.\right)}_{\substack{\text{a posteriori prob.}\\ \text{of '}m_i\text{' given }\bar{r}}} \underbrace{Pr\left(\bar{r}\right)}_{\substack{\text{joint probability}\\ \text{of }\bar{r}}} = \underbrace{Pr\left(\bar{r}\left|m_i\right.\right)}_{\substack{\text{a priori prob.}\\ \text{of }\bar{r}\text{ given 'm}_i\text{'}}} \underbrace{Pr\left(m_i\right)}_{\frac{1}{M}} \qquad 4.19.3$$

$Pr\left(m_i\left|\bar{r}\right.\right)$: A posteriori probability of m_i given $\vec{r}$

$Pr(\bar{r})$: Joint pdf of $\vec{r}$, defined over the entire set of signals { $s_i(t)$ }; independent of any specific message 'm_i '

$Pr(\bar{r}\,|m_i)$: Probability that a specific $\bar{r}$ will be received if the message m_i is transmitted; known as the a priori probability of $\vec{r}$ given m_i

$Pr(m_i)$: $1/M$

From Eq. 4.19.3, we see that determination of maximum a posteriori probability is equivalent to determination of maximum a priori probability $Pr(\bar{r}\,|m_i)$. This a priori probability is also known as the '*likelihood function*'.

So the decision rule can equivalently be stated as:

Set $\hat{m} = m_i$ if $Pr(\bar{r}\,|m_i)$ is maximum for k = i

Usually, $\ln\left[p_r(\vec{r}|m_k)\right]$, i.e. natural logarithm of the likelihood function is considered. As the likelihood function is non-negative, another equivalent form for the decision rule is:

Set $\hat{m} = m_i$ if $\ln[Pr(\bar{r}\,|m_i)]$ is maximum for k = i 4.19.4

A '*Maximum Likelihood Detector*' realizes the above decision rule.Towards this, the signal space is divides in M decision regions, Z_i , i = 1, 2, …, M such that,

$$\begin{cases} \text{vector } \bar{r} \text{ lies inside } 'Z_i\,' \text{if,} \\ \ln\left[P_r(\bar{r}|m_k)\right] \text{is maximum for k = i} \end{cases} \qquad 4.19.5$$

Fig. 4.19.2 indicates two decision zones in a two-dimensional signal space. The received vector $\bar{r}$ lies inside region Z_i if $\ln\left[p_r(\vec{r}|m_k)\right]$ is maximum for k = i.

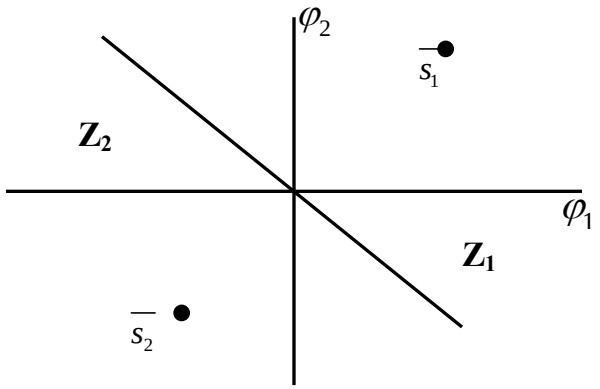


Fig. 4.19.2 *Decision zones in a two-dimensional signal space*

Now for an AWGN channel, the following statement is equivalent to ML decision:

Received vector $\vec{r}$ lies inside decision region Z_i

if, $\sum_{j=1}^{N}\left(r_j - s_{kj}\right)^2$ is minimum for k = i 4.19.6

That is, the decision rule simply is to choose the signal point $\bar{s}_i$ if the received vector $\vec{r}$ is closest to $\bar{s}_i$ in terms of Euclidean distance. So, it appears that Euclidean distances of a received vector $\vec{r}$ from all the signal points are to be determined for optimum decision-making. This can, however, be simplified. Note that, on expansion we get,

$$\sum_{j=1}^{N}\left(r_j - s_{kj}\right)^2 = \sum_{j=1}^{N} r_j^{\,2} - 2\sum_{j=1}^{N} r_j . s_{kj} + \sum_{j=1}^{N} s^2{}_{kj} \qquad 4.19.7$$

It is interesting that, the first term on the R.H.S, i.e., $\sum_{j=1}^{N} r_j^{\,2}$ is independent of 'k' and hence need not be computed for our purpose. The second term, $2\sum_{j=1}^{N} r_j . s_{kj}$ is the inner product of two vectors. The third term, i.e. $\sum_{j=1}^{N} s_{kj}^{\,2}$ is the energy of the k-th symbol. If the modulation format is so chosen that all symbols carry same energy, this term also need not be computed. We will see in Module #5 that many popular digital modulation schemes such as BPSK, QPSK exhibit this property in a linear time invariant channel.

So, a convenient observation is: the received vector $\vec{r}$ lies in decision region Z_i if, $\left(\sum_{j=1}^{N} r_j s_{kj} - \frac{1}{2}E_k\right)$ is maximum for k = i

That is, a convenient form of the ML decision rule is:

Choose $\hat{m} = m_i$ if $\left(\sum_{j=1}^{N} r_j s_{kj} - \frac{1}{2}E_k\right)$ is maximum for k = i 4.19.8

A *Correlation Receiver*, consisting of a Correlation Detector and a Vector Receiver implements the M – L decision rule [4.19.8] by, (a) first finding $\vec{r}$ with a correlation detector and then (b) computing the metric in [4.19.8] and taking decision in a vector receiver. **Fig. 4.19.3** shows the structure of a Correlation Detector for determining the received vector $\bar{r}$ from the received signal r(t). **Fig. 4.19.4** highlights the operation of a Vector Receiver.

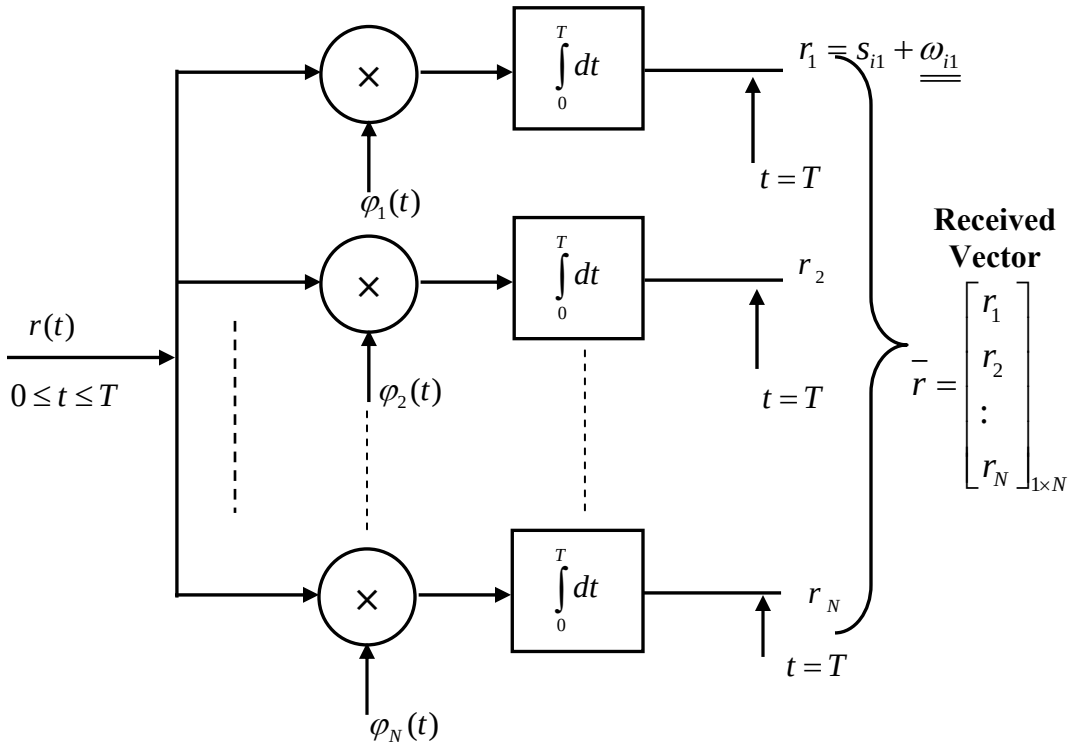


Fig. 4.19.3 *The structure of a Correlation Detector for determining the received vector* $\bar{r}$ *from the received signal r(t)*

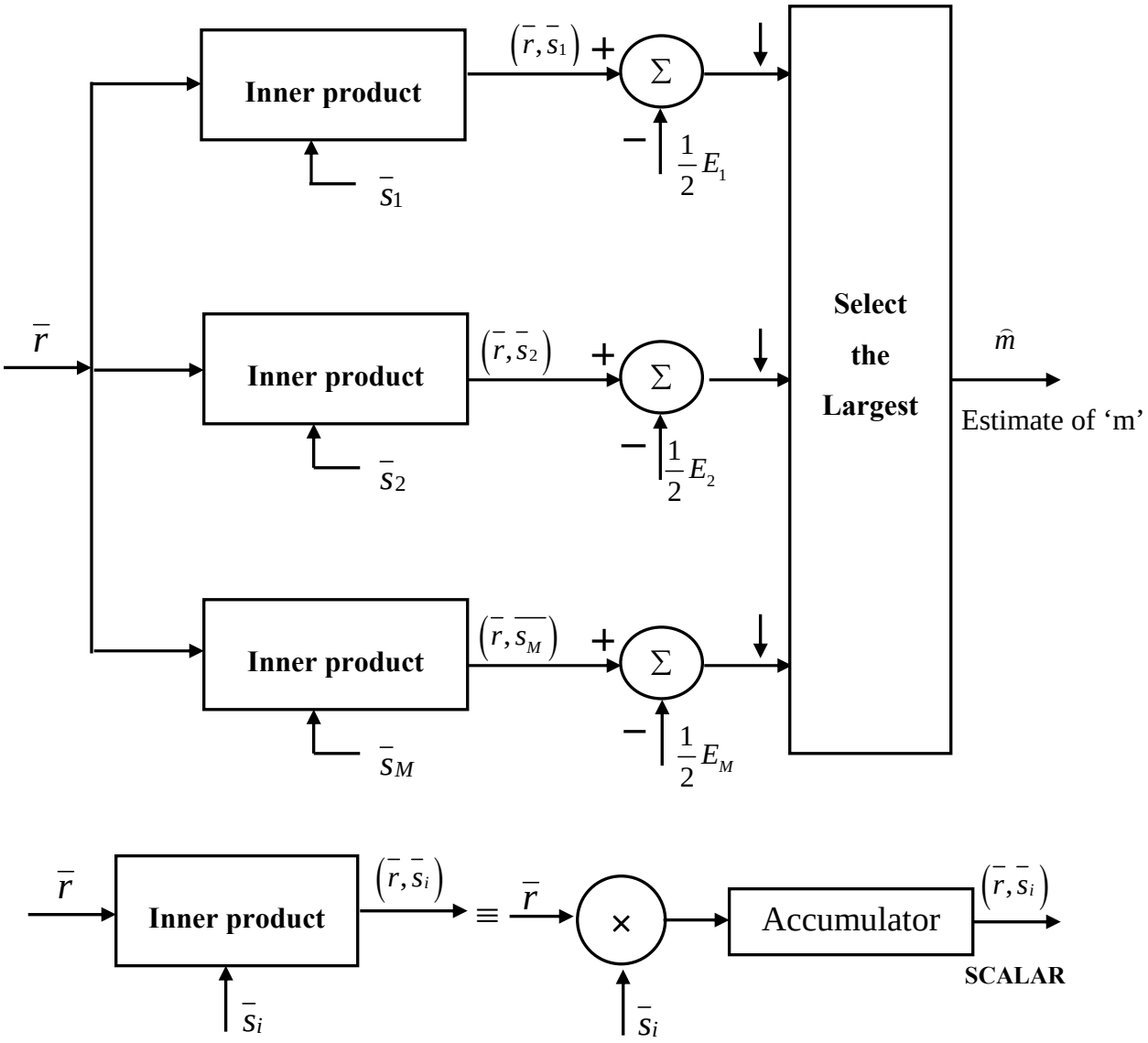


Fig. 4. 19.4 *Block schematic diagram for the Vector Receiver*

Features of the received vector $\bar{r}$

We will now discuss briefly about the statistical features of the received vector $\bar{r}$ as obtained at the output of the correlation detector [**Fig. 4.19.3**]. The j-th element of $\bar{r}$, which is obtained at the output of the j-th correlator once in T second, can be expressed as:

$$r_j = \int_0^T r(t)\Phi_j(t)dt = \int_0^T \left[s_i(t) + w(t)\right]\Phi_j(t)dt$$

$$= s_{ij} + w_j \; ; \quad j=1,2,\ldots..,\ N \qquad 4.19.9$$

Here w_j is a Gaussian distributed random variable with zero mean and s_{ij} is a scalar signal component of $\bar{s}_i$. Now, the mean of the correlator out put is, $E\left[r_j\right] = E\left[s_{ij} + w_j\right] = E\left[s_{ij}\right] = s_{ij} = m_{rj}$, say. We note that the mean of the correlator out put is independent of the noise process. However, the variances of the correlator outputs are dependent on the strength of accompanying noise:

$$Var\left[r_j\right] = \sigma_{r_j}^2 \quad = E\left[\left(r_j - s_{ij}\right)^2\right] = E\left[w_j^2\right]$$
$$= E\left[\int_0^T w(t)\Phi_j(t)dt \int_0^T w(u)\Phi_j(u)du\right]$$
$$= E\left[\int_0^T\int_0^T \Phi_j(t)\Phi_j(u).w(t)w(u)dtdu\right]$$

Taking the expectation operation inside, we can write

$$\sigma_{rj}^2 = \int_0^T\int_0^T \Phi_j(t)\Phi_j(u)E\left[w(t).w(u)\right]dtdu$$
$$= \int_0^T\int_0^T \Phi_j(t)\Phi_j(u)R_w(t,u)dtdu \qquad 4.19.10$$

Here, R_w (t-u) is the auto correlation of the noise process. As we have learnt earlier, additive white Gaussian noise process is a WSS random process and hence the auto-correlation function may be expressed as, $R_w(t,u) = R_w(t-u)$ and further, $R_w(t-u) = \frac{N_0}{2}\delta(t-u)$, where '$N_o$' is the single-sided noise power spectral density in Watt/Hz. So, the variance of the correlator output now reduces to:

$$\sigma_{rj}^2 = \frac{N_0}{2}\int_0^T\int_0^T \Phi_j(t)\Phi_j(u)\delta(t-u)dtdu$$
$$= \frac{N_0}{2}\int_0^T \Phi_j^2(t)dt = \frac{N_0}{2} \qquad 4.19.11$$

It is interesting to note that the variance of the random signals at the out puts of all N correlators are a) same, b) independent of information-bearing signal waveform and c) dependent only on the noise psd.

Now, the likelihood function for $s_i(t)$, as introduced earlier in Eq.4.19.3 and the ML decision rule [4.19.5] , can be expressed in terms of the output of the correlation detector. The likelihood function for 'm_i' = $Pr\left(\overline{r}\,|m_i\right) = f_{\bar{r}}\left(\bar{r}\middle|m_i\right) = f_{\bar{r}}\left(\bar{r}\middle|s_i(t)\right)$, where, $f_r(r|m_i)$ is the conditional pdf of 'r' given 'm_i'.

In our case, $f_{\bar{r}}\left(\bar{r}\middle|m_i\right)=\prod_{j=1}^{N} f_{r_j}\left(r_j\middle|m_i\right)$, i = 1,2,…,M 4.19.12

where, $f_{r_j}\left(r_j\middle|m_i\right)$ is the pdf of a Gaussian random variable with mean s_{ij} & var. = $\sigma_{r_j}^2$ = $\frac{N_0}{2}$, i.e., $f_{r_j}\left(r_j\middle|m_i\right)=\frac{1}{\sqrt{2\pi\sigma_{r_j}^2}}.e^{-\frac{(r_j-s_{ij})^2}{2\sigma_{rj}^2}}$ 4.19.13

Combining Eq. 4.19.12 and 4.19.13, we finally obtain,

$$f_{\bar{r}}\left(\bar{r}\middle|m_i\right)=\left(\pi N_0\right)^{-\frac{N}{2}}.\exp\left[-\frac{1}{N_0}\sum_{j=1}^{N}\left(r_j-s_{ij}\right)^2\right],\quad \text{i=1,2,……..,M} \qquad 4.19.14$$

This generic expression is of fundamental importance in analyzing error performance of digital modulation schemes [Module #5].

Problems

Q4.19.1) Consider a binary transmission scheme where a bit ‘1’ is represented by +1.0 and a bit ‘0’ is represented by –1.0. Determine the basis function if no carrier modulation scheme is used. If the additive noise is a zero mean Gaussian process, determine the mean values of r_1 and r_2 at the output of the correlation detector. Further, determine E_1 and E_2 as per Fig 4.19.4.

Module 4

Signal Representation and Baseband Processing

Lesson 20

Matched Filter

After reading this lesson, you will learn about

- ***Principle of matched filter (MF);***
- ***Properties of a matched filter;***
- ***SNR maximization and minimization of average symbol error probability;***
- ***Schwartz's Inequality;***

Certain structural modification and simplifications of the correlation receiver are possible by observing that,

(a) All orthonormal basis functions $\varphi_j - s$ are defined between $0 \leq t \leq T_b$ and they are zero outside this range .

(b) Analog multiplication, which is not always very simple and accurate to implement, of the received signal r(t) with time limited basis functions may be replaced by some filtering operation.

Let, $h_j(t)$ represent the impulse response of a linear filter to which r(t) is applied.

Then, the filter output $y_j(t)$ may be expressed as:

$$y_j(t) = \int_{\infty}^{\infty} r(\tau)\, h_j(t-\tau) d\tau \qquad 4.20.1$$

Now, let, $h_j(t) = \varphi_j(T-t)$, a time reversed and time-shifted version of $\varphi_j(t)$.

Now, $y_j(t) = \int_{-\infty}^{\infty} r(\tau).\varphi_j[T-(t-\tau)]d\tau$

$$= \int_{-\infty}^{\infty} r(\tau).\varphi_j(T+\tau-t)d\tau \qquad 4.20.2$$

If we sample this output at t = T,

$$y_j(T) = \int_{-\infty}^{\infty} (\tau).\varphi_j(\tau)d\tau \qquad 4.20.3$$

Let us recall that $\varphi_j(t)$ is zero outside the interval $0 \leq t \leq T$. Using this, the above equation may be expressed as,

$$y_j(T) = \int_0^T r(\tau)\varphi_j(\tau)d\tau$$

From our discursion on correlation receiver, we recognize that,

$$r_j = \int_0^T r(\tau)\varphi_j(\tau)d\tau = y_j(\tau) \qquad 4.20.4$$

The important expression of (Eq.4.20.4) tells us that the j – th correlation output can equivalently be obtained by using a filter with $h_j(t) = \varphi_j(T-t)$ and sampling its output at t = T.

The filter is said to be matched to the orthonormal basis function $\varphi_j(t)$ and the alternation receiver structure is known as a matched filter receiver. The detector part of the matched filter receiver is shown in [**Fig.4.20.1**].

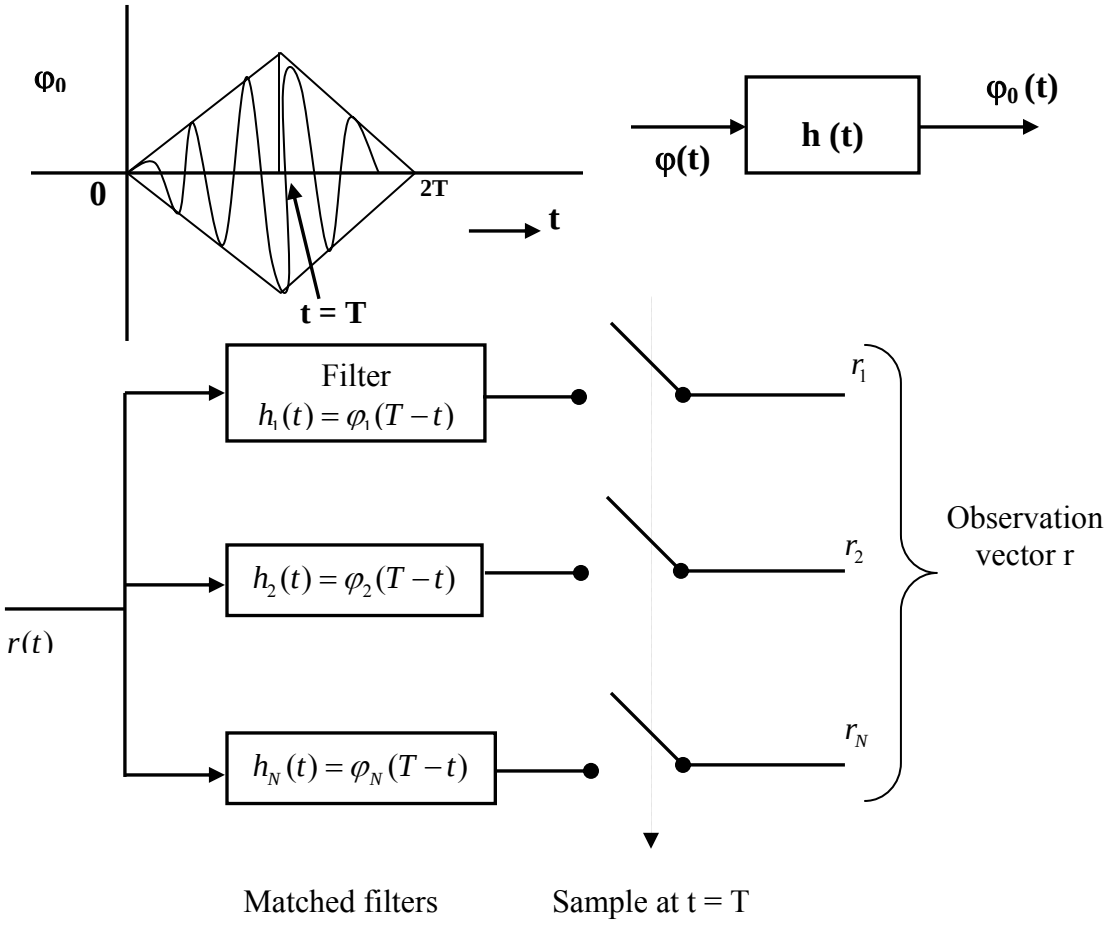


Fig. 4.20.1: *The block diagram of a matched filter bank that is equivalent to a Correlation Detector*

A physically realizable matched filter is to be causal and $h_j(t) = 0$ *for* $t < 0$. Note that if $\varphi_j(t)$ is zero outside $0 \le t \le T$, $h_j(t) = \varphi_j(T-t)$ is a causal impulse response.

Properties of a Matched Filter

We note that a filter which is matched to a known signal $\varphi(t), 0 \le t \le T$, is characterized by an impulse response h(t) which is a time reversed and delayed version of $\varphi(t)$ i.e.

$$h(t) = \varphi(T-t) \qquad 4.20.5$$

In the frequency domain, the matched filter is characterized (without mach explanation at this point), by a transfer function, which is, except for a delay factor, the complex conjugate of the F.T. of $\varphi(t)$, i.e.

$$\mathrm{H}(f) = \Phi^*(f)\exp(-j2\pi fT) \qquad 4.20.6$$

Property (1) : The spectrum of the output signal of a matched filter with the matched signal as input is, except for a time delay factor, proportional to the energy spectral density of the input signal.

Let, $\Phi_0(f)$ denote the F.T. of the filter of output $\varphi_0(t)$. Then,

$$\begin{aligned}\Phi_0(f) &= \mathrm{H}(f)\Phi(f) \\ &= \Phi^*(f)\Phi(f)\exp^{(-j2\pi fT)} \\ &= \underbrace{|\Phi(f)|^2}_{\substack{\text{Energy spectral} \\ \text{density of}\varphi(\mathrm{t})}} \exp^{(-j2\pi fT)} \qquad 4.20.7\end{aligned}$$

Property (2): The output signal of a matched filter is proportional to a shifted version of the autocorrelation function of the in the input signal to which the filter is matched.

This property follows from Property (1). As the auto-correlation function and the energy spectral density form F.T. pair, by taking IFT of (Eq.4.20.7), we may write,

$$\varphi_0(t) = R_\varphi(t-T) \qquad 4.20.8$$

Where $R_\varphi(\tau)$ is the act of $\varphi(t)$ for '$lag\ \tau$'. Note that at t = T,

$$R_\varphi(0) = \varphi_0(t) = \text{Energy of } \varphi(t). \qquad 4.20.9$$

Property (3): The output SNR of a matched filter depends only on the ratio of the signal energy to the psd of the white noise at the filter input.

Let us consider a filter matched to the input signal $\varphi(t)$.

From property (2), we see that the maximum value of $\varphi_0(t)$ at t = T is $\varphi_0(t-T) = E$.

Now, it may be shown that the average noise power at the output of the matched filter is given by, $E[n^2(t)] = \dfrac{N_0}{2}\int_{-\infty}^{\infty}|\varphi(f)|^2\,df = \dfrac{N_0}{2}.E$ 4.20.10

The maximum signal power $= |\varphi_0(T)|^2 = E^2$.

Hence, $$(SNR)_{\max} = \frac{E^2}{\frac{N_0}{2}E} = \frac{2E}{N_0} \qquad 4.20.11$$

Note that SNR in the above expression is a dimensionless quantity.

This is a very significant result as we see that the $SNR_{\max}$ depends on E and N_0 but not on the shape of $\varphi(t)$. This means a freedom to the designer to select specific pulse shape to

optimize other design requirement (the most usual requirement being the spectrum or, equivalently, the transmission bandwidth) while ensuring same SNR.

Property (4): The matched-filtering operation may be separated into two matching condition: namely, spectral phase matching that produces the desired output peak at t = T and spectral amplitude matching that gives the peak value its optimum SNR.

$$\Phi(f) = |\Phi(f)| \exp[j\theta(f)] \qquad 4.20.12$$

The filter is said to be matched to the signal $\varphi(t)$ in spectral phase if the transfer function of the filter follows:

$$H(f) = |H(f)| \exp[-j\theta(f) - j2\pi fT] \qquad 4.20.13$$

Here $|H(f)|$ is real non-negative and 'T' is a positive constant.

The output of such a filter is,

$$\varphi_0'(t) = \int_{-\infty}^{\infty} H(f).\Phi(f), \exp(j2\pi ft)df$$

$$= \int_{-\infty}^{\infty} |H(f)||\Phi(f)|.\exp[j2\pi f(t-T)df$$

Note that, $|H(t)||\varphi(t)|$ is real and non-negative. Spectral phase matching ensures that all spectral components of $\varphi_0'(t)$ add constructively at t = T and thus cause maximum value of the output:

$$\varphi_0'(T) = \int_{-\infty}^{\infty} |\Phi(f)||H(f)|df. \geq \varphi_0'(t) \qquad 4.20.14$$

For spectral amplitude matching, we choose the amplitude response $|H(f)|$ of the filter to shape the output for best SNR at t = T by using $|H(f)| = |\Phi(f)|$. The standard matched filter achieves both these features.

Maximization of output Signal –to-Noise Raito:

Let, h(t) be the impulse response of a linear filter and $x(t) = \varphi(t) + \omega(t), 0 \leq t \leq T$: is the input to the filter where $\varphi(t)$ is a known signal and $\omega(t)$ is an additive white noise sample function with zero mean and psd of $(N_0/2)$ Watt/Hz. Let, $\varphi(t)$ be one of the orthonormal basis functions. As the filter is linear, its output can be expressed as, $y(t) = \varphi_0(t) + n(t)$, where $\varphi_0(t)$ is the output due to the signal component $\varphi(t)$ and n(t) is the output due to the noise component $\omega(t)$.[**Fig. 4.20.2**].

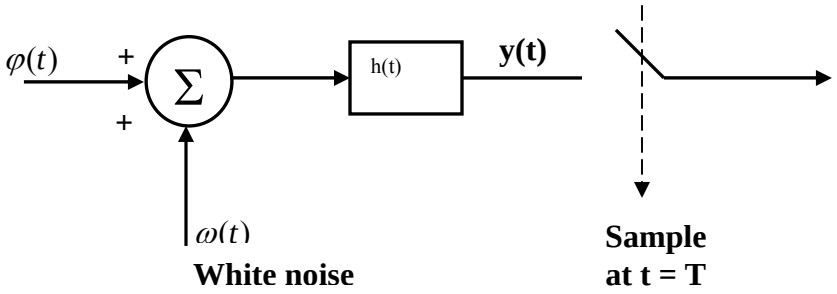


Fig. 4.20.2: *A matched filter is fed with a noisy basis function to which it is matched*

We can now re-frame the requirement of minimum probability of error (or maximum likelihood detection) as: The filter should make power of $\varphi_0(t)$ considerably greater (in fact, as large as possible) compared to the power of n(t) at t = T. That is, the filter should maximize the output signal-to-noise power ratio $[(SNR)_0]$

$$\triangleq \frac{|\varphi_0(T)|^2}{E[n^2(t)]}\Bigg]_{\max}$$

The following discussion shows that the SNR is indeed maximized when h(t) is matched to the known input signal $\varphi(t)$.

Let, $\Phi(f)$: F.T. of known signal $\varphi(t)$

$H(f)$: Transfer function of the linear filter.

$$\therefore \Phi_0(f) = H(f)\Phi(f)$$

and $$\Phi_0(t) = \int_{-\infty}^{\infty} H(f)\Phi(f)\exp(j2\pi ft)df \qquad 4.20.15$$

The filter output is sampled at t = T. Now,

$$|\varphi_0(T)|^2 = \left|\int_{-\infty}^{\infty} H(f)\Phi(f)\exp(j2\pi fT)df\right|^2 \qquad 4.20.16$$

Let, $S_N(f)$: Power spectral density of noise at the output of the linear filter. So,

$$S_N(f) = \frac{N_0}{2}.|H(f)|^2 \qquad 4.20.17$$

Now, the average noise power at the output of the filter

$$= E[n^2(t)] = \int_{-\infty}^{\infty} S_N(f)df$$

$$= \frac{N_0}{2} \int_{-\infty}^{\infty} |H(f)|^2 \, df \qquad 4.20.18$$

Form Eq. 4.20.16 and 4.20.18, we can write an expression of the output SNR as:

$$(SNR)_0 = \frac{|\varphi^2(T)|^2}{E[n^2(t)]} = \frac{\left| \int_{-\infty}^{\infty} H(f).\varphi(f) \exp(j2\pi fT) df \right|^2}{\frac{N_0}{2} \int_{-\infty}^{\infty} |H(f)|^2 \, df} \qquad 4.20.19$$

Our aim now is to find a suitable form of H(f) such that $(SNR)_0$ is maximized. We use Schwarz's inequality for the purpose.

Schwarz's Inequality

Let $\bar{x}(t)$ and $\bar{y}(t)$ denote any pair of complex-valued signals with finite energy, i.e. $\int_{-\infty}^{\infty} |\bar{x}(t)|^2 \, dt < \infty \quad \& \quad \int_{-\infty}^{\infty} |\bar{y}(t)|^2 \, dt < \infty$. Schwarz's Inequality states that,

$$\left| \int_{-\infty}^{\infty} \bar{x}(t)\bar{y}(t)dt \right|^2 \le \int_{-\infty}^{\infty} |\bar{x}(t)|^2 \, dt . \int_{-\infty}^{\infty} |\bar{y}(t)|^2 \, dt. \qquad 4.20.20$$

The equality holds if and only if $\bar{y}(t) = k.\bar{x}^*(t)$, where 'k' is a scalar constant. This implies, $\bar{y}(t)\bar{x}(t) = k.\bar{x}(t)\overline{x^*}(t) \rightarrow$ a real quantity.

Now, applying Schwarz's inequality on the numerator of (Eq.4.20.19), we may write,

$$\left| \int_{-\infty}^{\infty} H(f)\Phi(f) \exp(j2\pi fT) df \right|^2 \le \int_{-\infty}^{\infty} |H(f)|^2 \, df \int_{-\infty}^{\infty} |\Phi(f)|^2 \, df \qquad 4.20.21$$

Using inequality (4.20.21), equation (4.20.19) may be expressed as,

$$(SNR)_0 \le \frac{2}{N_0} \int_{-\infty}^{\infty} |\varphi(f)|^2 \, df \qquad 4.20.22$$

Now, from Schwarz's inequality, the SNR is maximum i.e. the equality holds, when

$H_{opt}(f) = \Phi^*(f).\exp(-j2\pi fT).$ [Assuming k = 1, a scalar)

We see, $h_{opt}(t) = \int_{-\infty}^{\infty} \Phi^*(f) \exp[-j2\pi(T-t)] df \qquad 4.20.23$

Now, $\varphi(t)$ is a real valued signal and hence,

$$\Phi^*(f) = \Phi(-f) \qquad 4.20.24$$

Using Eq. 4.20.24 we see,

$$h_{opt}(t) = \int_{-\infty}^{\infty} \Phi(-f)\exp[-j2\pi)(T-t)]df = \varphi(T-t)$$

$$\therefore h_{opt}(t) = \Phi(T-t)$$

This relation is the same as we obtained previously for a matched filter receiver. So, we can infer that, *SNR maximization is an operation, which is equivalent to minimization of average symbol error (P_e) for an AWGN Channel.*

Example #4.20.1: Let us consider a sinusoid, defined below as the basis function:

$$\varphi(t) = \begin{cases} \sqrt{\frac{2}{T}}\cos w_c t, 0 \le t \le T \\ 0, \quad elsewhere. \end{cases}$$

$$h_{opt.}(t) = \varphi(T-t) = \varphi(t). \qquad h(t) = \varphi(T-t) = \varphi(t)$$

$$\varphi_0(t) = \begin{cases} \frac{t}{T}\cos w_c t. & 0 \le t \le T \\ \left(2 - \frac{t}{T}\right)\cos w_c t. & T \le t \le 2T \\ 0 & \text{else.} \end{cases}$$

Problems

Q4.20.1) Under what conditions matched filter may be considered equivalent to an optimum correlation receiver?

Q4.20.2) Is a matched filter equivalent to an optimum correlation receiver if sampling is not possible at the right instants of time?

Q4.20.3) Explain the significance of the fact that a matched filter ensures maximum output signal-to-noise ratio.

Module 4

Signal Representation and Baseband Processing

Lesson 21

Nyquist Filtering and Inter Symbol Interference

After reading this lesson, you will learn about:

- ***Power spectrum of a random binary sequence;***
- ***Inter symbol interference (ISI);***
- ***Nyquist filter for avoiding ISI;***
- ***Practical improvisation of ideal Nyquist filter;***
- ***Raised Cosine (RC) filter and Root–Raised Cosine (RRC) filtering;***

Nyquist's sampling theorem plays a significant role in the design of pulse-shaping filters, which enable us to restrict the bandwidth of information-bearing pulses. In this lesson, we start with a short discussion on the spectrum of a random sequence and then focus on the concepts of Nyquist Filtering. Specifically, we develop an idea about the narrowest (possible) frequency band that will be needed for transmission of information at a given symbol rate.

Consider a random binary sequence shown in **Fig.4.21.1 (a)** following a common style (NRZ: Non-Return-to-Zero Pulses) for its representation. Note that a binary random sequence may be represented in several ways. For example, consider **Fig.4.21.1 (b).** The impulse sequence of **Fig.4.21.1 (b)** is an instantaneously sampled version of the NRZ sequence in **Fig.4.21.1 (a)** with a sampling rate of one sample/pulse. The information embedded in the random binary sequence of **Fig.4.21.1 (a)** is fully preserved in the impulse sequence of **Fig.4.21.1 (b)**. Verify that you can easily read out the logical sequence only by looking at the impulses. So, the two waveforms are equivalent so far as their information content is concerned. The obvious difference, from practical standpoint, is the energy carried by a pulse.

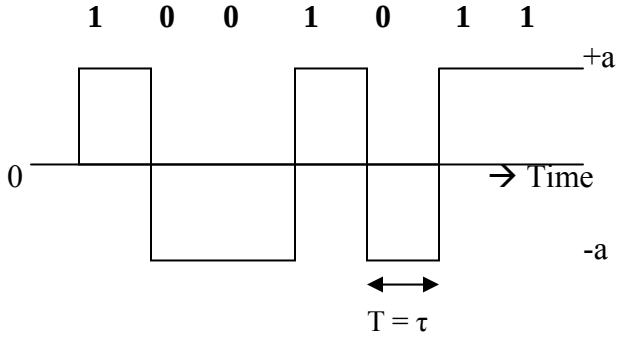


Fig.4.21.1(a) *Sketch of a NRZ (Non-Return-to-Zero) waveform for a random binary sequence*

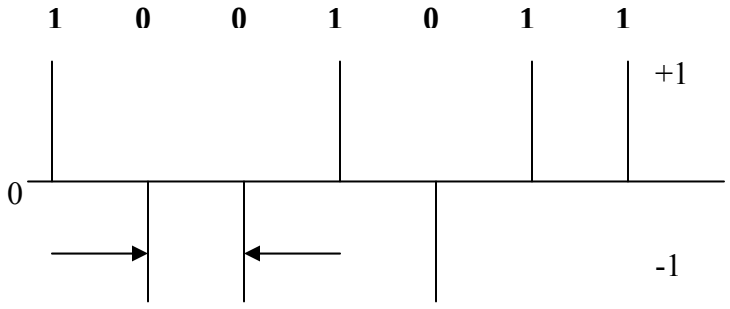


Fig.4.21.1 (b): *Instantaneously sampled version of the NRZ sequence in* **Fig.4.21.1 (a)**

Power Spectrum of Random Binary Sequence

Let us consider the NRZ representation of **Fig.4.21.2(a)** where in a pulse is of height 'a' and duration T_b. We wish to get an idea about the spectrum of such sequences. The sequence may be viewed as a sample function of a random process, say X(t). So, our approach is to find the ACF of X(t) first and then take its Fourier Transform. Now, the starting instant of observation need not synchronize with the start time of a pulse. So, we assume that the starting time of the first pulse (i.e. the initial delay)'t_d' is equally likely to lie anywhere between 0 and T_b, i.e.,

$$p_n(t_\theta) = \begin{cases} \frac{1}{T_b}, 0 \le t_d \le T_b \\ 0, elsewhere \end{cases} \qquad 4.21.1$$

Next, the '0'-s and '1'-s are equally likely. So, we can readily note that, E[X(t)] = 0.

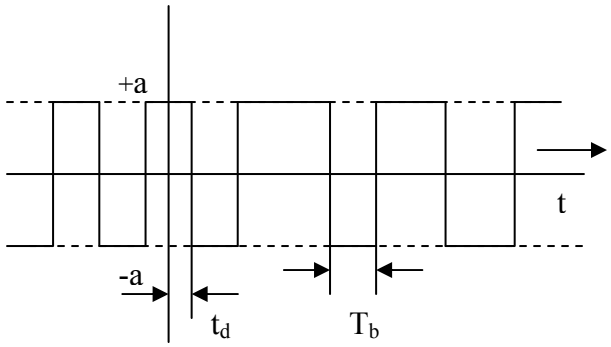


Fig. 4.21.2(a): *A random NRZ sequence representing an information sequence*

ACF of X(t)

Let $R_x(t_k,t_i)$ denote the ACF of X(t).

$$\therefore R_x(t_k,t_i) = E[X(t_k) \cdot X(t_i)] \qquad 4.21.2$$

Two distinct cases are to be considered: a) when the shift, i.e. $|t_k - t_i|$ is greater than the bit duration T_b and b) when $|t_k - t_i| \le T_b$.

Case-I: Let, $|t_k - t_i| > T_b$.

In this case, X (t_k) and X (t_i) occur in different bit intervals and hence they are independent of each other. This implies,

$$E[X(t_k) \cdot X(t_i)] = E[X(t_i)] \cdot E[X(t_k)] = 0; \qquad 4.21.3$$

Case-II: $|t_k - t_i| < T_b$. For simplicity, let us set $t_k = 0$ and $t_i < t_k = 0$.

In this case, the random variables $X(t_k)$ and $X(t_i)$ occur in the same pulse interval iff $t_d < T_b - |t_k - t_i|$. Further, both $X(t_k)$ and $X(t_i)$ are of same magnitude 'a' and same polarity.

Thus, we get a conditional expectation:

$$E\left[X(t_k)\cdot X(t_i)\Big|_{t_d}\right]=\begin{cases}a^2, t_d < T_b-|t_k-t_i| \\ 0, elsewhere\end{cases} \qquad 4.21.4$$

Averaging this result over all possible values of t_d, we get,

$$E\left[X(t_k)\cdot X(t_i)\right]=\int_0^{T_b-|t_k-t_i|} a^2 p(t_d)\,dt_d$$

$$=\int_0^{T_b-|t_k-t_i|}\frac{a^2}{T_b}dt_d=a^2\left(1-\frac{|t_k-t_i|}{T_b}\right),\quad |t_k-t_i|\le T_b \qquad 4.21.5$$

By similar argument, it can be shown that for other values of t_k , the ACF of a binary waveform is a function of the time shift $\tau = t_k - t_i$. So, the autocorrelation function $R_x(\tau)$ can be expressed as [**Fig.4.21.2(b)**]:

$$R_x(\tau)=\begin{cases}a^2\left(1-\frac{|\tau|}{T_b}\right), |\tau|<T_b \\ 0, |\tau|\ge T_b\end{cases} \qquad 4.21.6$$

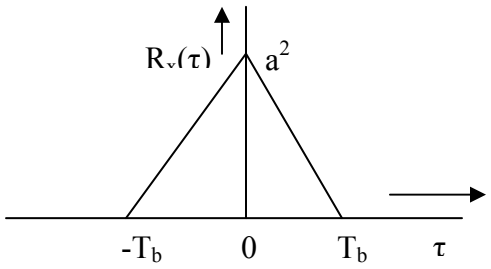


Fig. 4.21.2(b): *Auto Correlation Function $R_x(\tau)$ for a random binary waveform*

Now the power spectral density of the random process X(t) can be obtained by taking the Fourier Transform of $R_x(\tau)$:

$$S_x(f)=\int_{-T_b}^{T_b} a^2\left(1-\frac{|\tau|}{T_b}\right)\exp(-j2\pi f\tau)\,d\tau$$

$$=a^2T_b\sin c^2(fT_b) \qquad 4.21.7$$

A rough sketch of $S_x(f)$ is shown in **Fig. 4.21.2(c)**. Note that the spectrum has a peak value of '$a^2.T_b$'. The spectrum stretches towards $\pm\infty$ and it has nulls at $\pm 1/nT_b$. A normalized version of the spectrum is shown in **Fig. 4.21.2(d)** where the amplitude is normalized with respect to the peak amplitude and the frequency axis is expressed in terms of 'fT_b'.

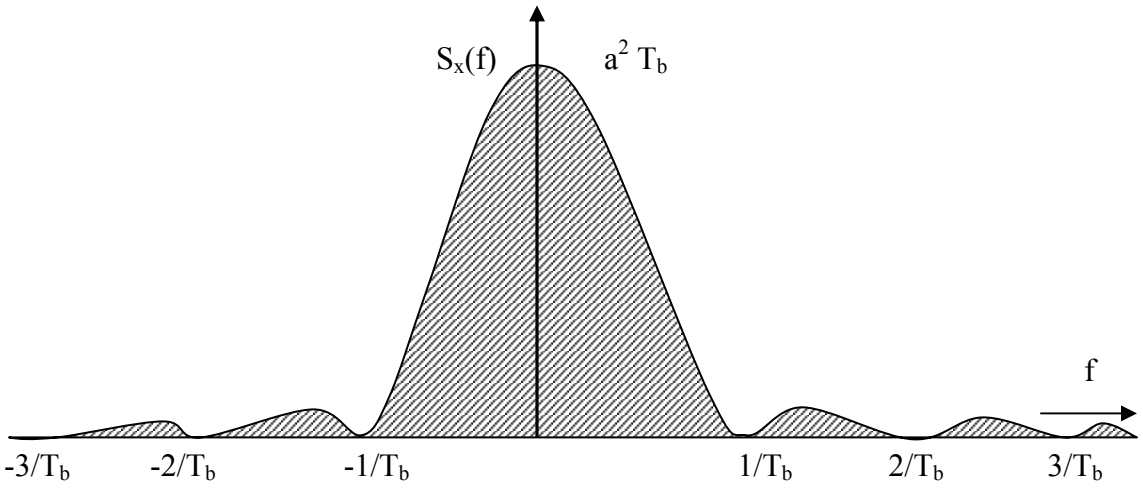


Fig. 4.21.2(c) : *A sketch of the power spectral density, $S_x(f)$ for a random binary NRZ pulse sequence*

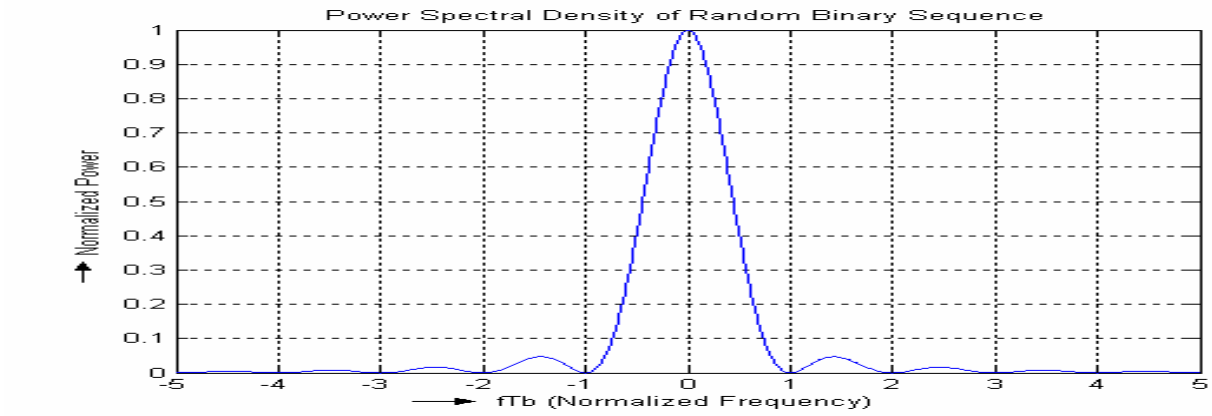


Fig. 4.21.2(d) : *Normalized power spectral density, $S_x(f)$ for a random binary NRZ pulse sequence*

The wide stretch of the spectrum is understandable as the time pulses are sharply limited within a bit duration. But then, if such a random NRZ sequence is used to modulate a carrier sinusoid, one can easily imagine that the modulated signal spectrum will also have an infinite width and hence, such a modulated signal cannot be transmitted without distortion though a wireless channel, which is band-limited. If the modulated signal is forced through a band limited channel without appropriate spectral shaping, the spectrum of the modulated signal at the receiver will be truncated due to the band pass characteristic of the channel. From the principle of time-frequency duality, one can now guess that the energy of a transmitted pulse will no more be limited within its typical duration 'T_b'.

Alternatively, the energy of one pulse will spill over the time slot of one or more subsequent pulses, causing *Inter Symbol Interference (ISI)*. So, over a specific pulse duration 'T_b', the receiver will collect energy due to one desired and multiple undesired pulses. A typically vulnerable situation is when a negative pulse appears in a string of positive pulses or vice versa. In general, the received signal becomes more vulnerable to noise and upon demodulation; the information sequence may be erroneous. The extent of degradation in the quality of received information depends on the time spread of energy of a transmitted pulse and how this effect of ISI is addresses in the receiver.

Another reason why sharp rectangular pulses, even though designed following Gramm-Scmidt orthogonalization procedure, are not good for band-limited channels is that, one is simply not allowed to use the full bandwidth that may be presented by a physical channel. Specifically, most wireless transmissions must have a priori approval from concerned regulatory authority of a country. It is mandatory that signal transmission is done precisely over the narrow portion of the allocated bandwidth so that the adjacent bands can be allocated for other transmission schemes.

Further, to conserve bandwidth for various transmission applications, the narrowest feasible frequency slot only may be allocated. So, it is necessary to address the issue of ISI in general and the issue of small transmission bandwidth by shaping pulses. There are several equalization techniques available for addressing the issue of ISI. Many of these techniques probe the physical channel and use the channel state information in devising powerful adaptive equalizers. In this course, we skip further discussion on equalization and focus only on the issue of pulse shaping for reduction in transmission bandwidth.

Nyquist Filter for avoiding ISI

Let us recollect the second part of Nyquist's sampling theorem for low pass signals which says that a signal band limited to B Hz can be recovered from a sequence of uniformly spaced and instantaneous samples of the signal taken at least at the rate of 2B samples per second. Following this theorem, we may now observe that the impulse sequence of **Fig.4.21.1 (b)**, which contains information '1001011', can be equivalently described by an analog signal, band limited to ½ x 1sample/pulse. That is, if the bit rate is 1 bit/sec and we sample it at 1 sample/sec, the minimum bandwidth necessary is ½ Hz! An ideal low-pass filter with brick-wall type frequency response and having a cutoff of 0.5 Hz will generate an equivalent analog waveform (pulse sequence) when fed with the random impulse sequence.

Recollect that the impulse response of an ideal low-pass filter is a sinc function [y = sinc(x), **Fig. 4. 21.3**]. We note that a) sinc(0) = 1.0 and the impulse response has nulls at x = ±1, ±2, …. sec, b) null-to-null time-width of the prominent pulse is 2 sec, c) the pulse is symmetric around t = 0 and d) the peak amplitude of the pulse is of the same polarity as that of the input impulse. As the filter is a linear network, the output analog waveform is simply superposition of sinc pulses, where the peaks of adjacent sinc pulses are separated in time by 1 sec, which is the sampling interval.

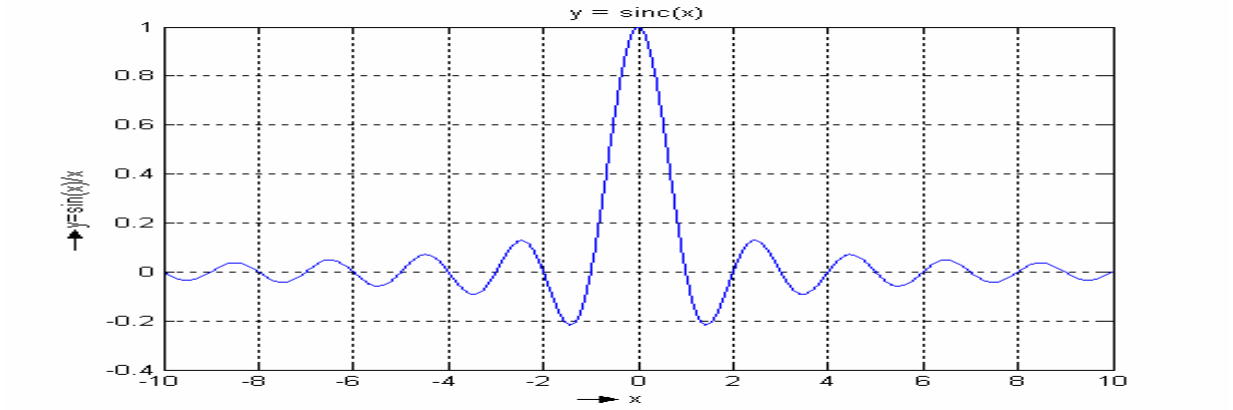


Fig. 4. 21.3(a) *Plot of y = sinc(x)*

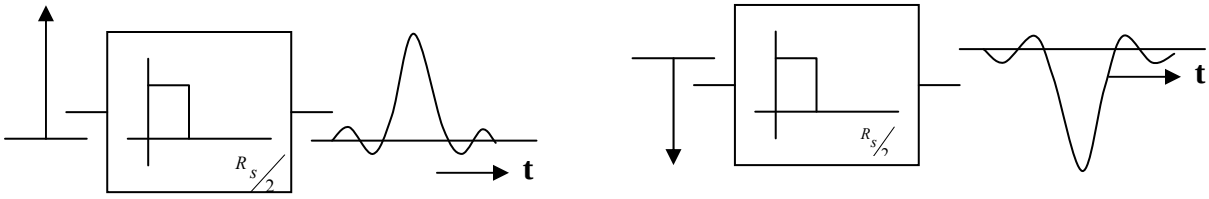


Fig. 4. 21.3(b) *Sketch of output of Nyquist filter for positive and negative impulses*

In general, if the information symbol rate is R_s symbols/sec, i.e. the symbol interval is $T_s = 1/R_s$ second, the single-sided bandwidth of the low-pass filter, known popularly as the equivalent Nyquist Bandwidth, is $B_N = R_s/2$ Hz.

A simple extension of the above observations implies that for instantaneous samples of random information-bearing pulse (or symbol) sequence (@ 1 sample per symbol) will exhibit nulls at $\pm nT_s$ seconds at the filter output. Now, assuming an ideal noise less baseband channel of bandwidth B_N and zero (or fixed but known) delay, we can comment that the same filter output waveform will appear at the input of the receiver. Though the waveform will appear much different from the initial symbol sequence, the information (embedded in the polarity and magnitude) can be retrieved without any effect of the other pulses by sampling the received baseband signal exactly at peak position of each shaped pulse as at those time instants all other pulses have nulls. This ideal brick-wall type filter is known as the Nyquist Filter for zero-ISI condition. **Fig. 4.21.4** highlights some important features of Nyquist Filter.

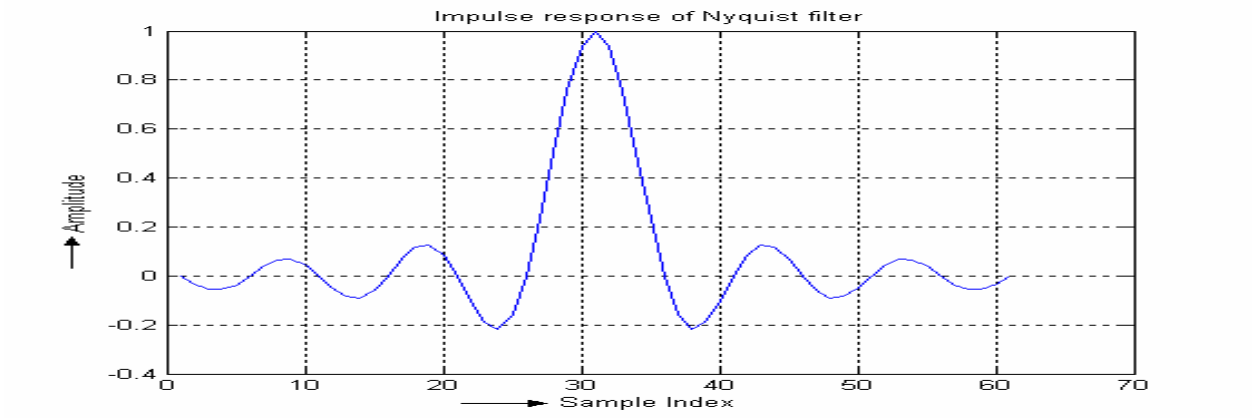


Fig. 4.21.4 *Typical impulse response of a Nyquist filter [5 samples /symbol duration have been used to obtain the responses using a digital computer. ±32 sample duration around the normalized peak value of 1.0 is shown]*

Mathematical Explanation

Let us consider an ideal lowpass filter with single-sided bandwidth W. It is usually convenient to deal with the normalized impulse response h(t) of the filter where in,

$$h(t)=\text{sinc}\,(2Wt) = \sin(2\pi Wt)\big/2\pi Wt \qquad 4.21.8$$

Now if an impulse of strength 'A' is applied at the input of the filter at $t = t_1$, the filter o/p may be expressed as,

$$y(t) = Ah(t- t_1)=A.\text{sinc}\ 2W\ (t- t_1) \qquad 4.21.9$$

An information-carrying symbol sequence may be represented as

$$\sum_{i=0}^{\infty} A_i.\delta(t-t_i), \text{ where } A_i = \pm A \text{ and } t_i = i.T_s \qquad 4.21.10$$

The response of the low-pass filter to this sequence is,

$$y(t) = \sum_{i=0}^{\infty} A_i.\sin c\{2W(t-t_i)\} \qquad 4.21.11$$

Now, if we set $W = R_s/2 = 1/2.T_s$ and sample the output of the filter at $t = m.T_s$, it is easy to see that,

$$y(t = m.T_s) = \sum_{i=0}^{m} A_i.\text{sinc}2W(m-i)T_s = \sum_{i=0}^{m} A_i.\text{sinc}(m-i) = A_m \qquad 4.21.12$$

So, we obtain the peak of the m-th pulse clean and devoid of any interference from previous pulses.

Practical improvisations:

The above discussion on Nyquist bandwidth is ideal because neither a low-pass filter (LPF) with brick-wall type response can be realized physically nor can an ideal impulse sequence be generated to represent a discrete information sequence. Further, note that if there is any error in the sampling instant (which, for a practical system is very likely to occur from time to time due to the effects of thermal noise and other disturbances), contribution from the other adjacent pulses will creep into the sample value and will cause Inter Symbol Interference. For example, the second lobe peak is only about 13 dB lower compared to the main lobe peak and the decay of the side-peaks of a sin x/x function is not very rapid with increase in x. So, the contribution of these peaks from adjacent symbols may be significant. Fortunately, the desired features of pulse epoch and zero crossings are not unique to a 'sinc' pulse. Other pulse shapes are possible to design with similar features, though the bandwidth requirement for transmitting a discrete information sequence will be more compared to the corresponding Nyquist bandwidth (B_N). Two such relevant constructs are known as a) Raised Cosine (RC) filter and b) Root-Raised Cosine (RRC) filter.

Raised Cosine Filter

Let us denote a normalized pulse shape which avoids ISI as x(t). then,

$$x(t)\big|_{t = \pm nT_s} = 0, \text{n} \neq 0 \text{ and } \text{x}(0) = 1 \qquad 4.21.13$$

Some of the practical requirements on x(t) are the following:

(a) Energy in the main pulse is as much as possible compared to the total energy distributed beyond the first nulls around the main peak. This ensures better immunity against noise at the receiver for a given signal transmission power. So, it is desired that the magnitde of the local maxima of the i-th pulse of x(t) between $iT \leq t < (i+1)T$ decreases monotonically and rapidly with time.

(b) The pulse shape x(t) should be so chosen that some error in instants of sampling at the receiver does not result in appreciable ISI. These two requirements are usually depicted in the form of a mask in technical standards.

Out of several mathematical possibilities, the following amplitude mask is very useful for application on the ideal Nyquist filter impulse response h(t):

$$m(t) = \frac{\cos\left(\beta\pi \, t/T_s\right)}{1-(4\beta^2 t^2 / T_s^{\,2})} \qquad 4.21.14$$

The resulting pulse shape is known as a Raised Cosine pulse with a *roll-off* of 'β'($0 \angle \beta \leq 1$). The Raised Cosine pulse is described as:

$$p_{RC}(\text{t}) = \text{sinc}(\text{t}/T_s).\ \frac{\cos\left(\beta\pi \, t/T_s\right)}{1-(4\beta^2 t^2 / T_s^{\,2})}$$

4.21.15

The normalized spectrum of Raised Cosine pulse is:

$$H(f) = 1, \text{ for } |f| \le \frac{(1-\beta)}{2T_s},$$

$$= \cos^2 \frac{\pi T_s}{2\beta}\left(|f| - \frac{(1-\beta)}{2T_s}\right), \quad \text{for } \frac{(1-\beta)}{2T_s} \le |f| \le \frac{(1+\beta)}{2T_s}$$

$$= 0, \quad \text{for } |f| > \frac{(1+\beta)}{2T_s} \qquad 4.21.16$$

Figs. 4.21.5 (a)-(c) highlight features of a Raised Cosine (RC) filter. The roll off factor 'β' is used to find a trade off between the absolute bandwidth that is to be used and the difficulty (in terms of the order of the filter and the associated delay and inaccuracy) in implementing the filter. The minimum usable bandwidth is obviously the Nyquist bandwidth, B_N (for β =0) for which the filter is unrealizable in practice while a maximum absolute bandwidth of $2B_N$ (for β = 1.0) makes it much easier to design the filter. β lies between 0.2 and 0.5 for most of the practical systems where transmission bandwidth is not a luxury. Use of analog components was dominant earlier though the modern trend is to use digital filters. While digital FIR structure usually ensures better accuracy and performance, IIR structure is also used to reduce the necessary hardware.

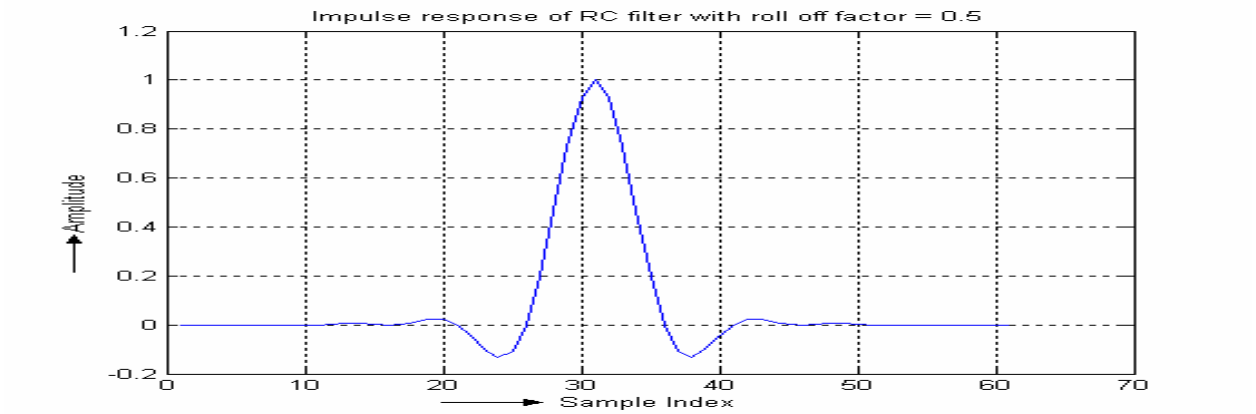


Fig. 4.21.5 (a) *Typical impulse response of a Raised Cosine filter with a roll off factor β = 0.5[5 samples /symbol duration have been used to obtain the responses using a digital computer. ±32 sample duration around the normalized peak value of 1.0 is shown]*

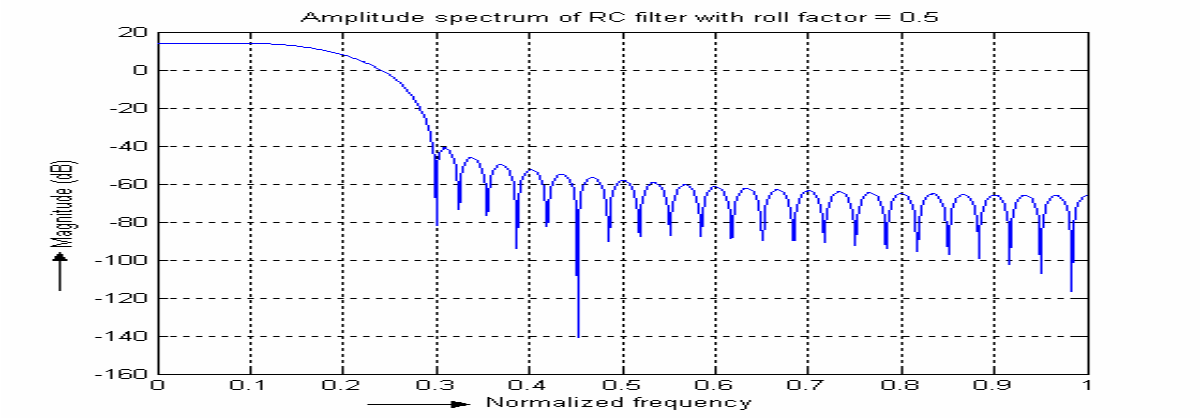


Fig. 4.21.5 (b) *Typical amplitude spectrum of a Raised Cosine filter with a roll off factor β= 0.5. The magnitude is not normalized. Multiply the normalized frequency values shown above by a factor of 5 to read the frequency normalized to symbol rate. For example, i) 0.1(from the above figure)x5 = 0.5 = (1 -β).R_s, where β=0.5 and R_s = 1. The transition band of the filter starts here and ii) 0.3(from the above figure)x5 = 1.5 = (1 +β).R_s. The stop band of the filter starts here.*

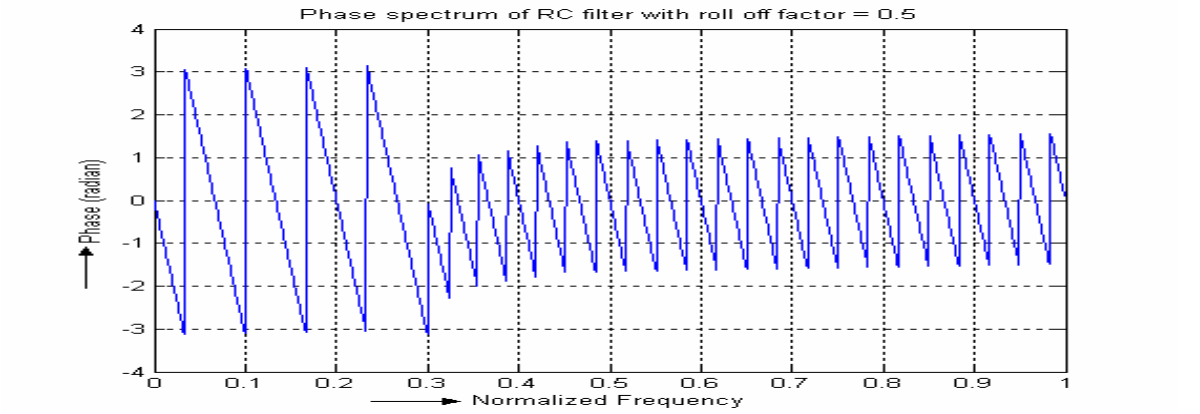


Fig. 4.21.5 (c) *Typical phase spectrum of a Raised Cosine filter with a roll off factor β = 0.5. Multiply the normalized frequency values shown above by a factor of 5 to read the frequency normalized to symbol rate.*

The side lobe peaks in the impulse response of a Raised Cosine filter decreases faster with time and hence results in less ISI compared to the ideal Nyquist filter in case of sampling error in the receiver.

There is another interesting issue in the design of pulse shaping filters when it comes to applying the concepts in a practical communication transceiver. From our discussion so far, it may be apparent that the pulse-shaping filter is for use in the transmitter only, before the signal is modulated by a carrier or launched in the physical channel. However, there is always a need for an equivalent lowpass filter in the receiver to eliminate out-of-band noise before demodulation and decision operations are carried

out. This purpose is accomplished in a practical receiver by splitting a Raised Cosine filter in two parts. Each part is known as a Root Raised Cosine (RRC) filter. One RRC filter is placed in the transmitter while the other part is placed in the receiver. The transmit RRC filter does the job of pulse shaping and bandwidth-restriction fully while not ensuring the zero-ISI condition completely. In case of a linear time invariant physical channel, the receiver RRC filter, in tandem with the transmit RRC filter, fully ensures zero-ISI condition. Additionally, it filters out undesired out-of-band thermal noise. On the whole, this approach ensures zero-ISI condition in the demodulator where it is necessary and it also effectively ensures that the equivalent noise-bandwidth of the received signal is equal to the Nyquist bandwidth B_N. The overall complexity of the transceiver is reduced without any degradation in performance compared to a system employing a RC filter in the transmitter and a different out-of-band noise eliminating filter in the receiver. **Figs. 4.21.6 (a)-(c)** summarize some features of a Root-Raised Cosine filter.

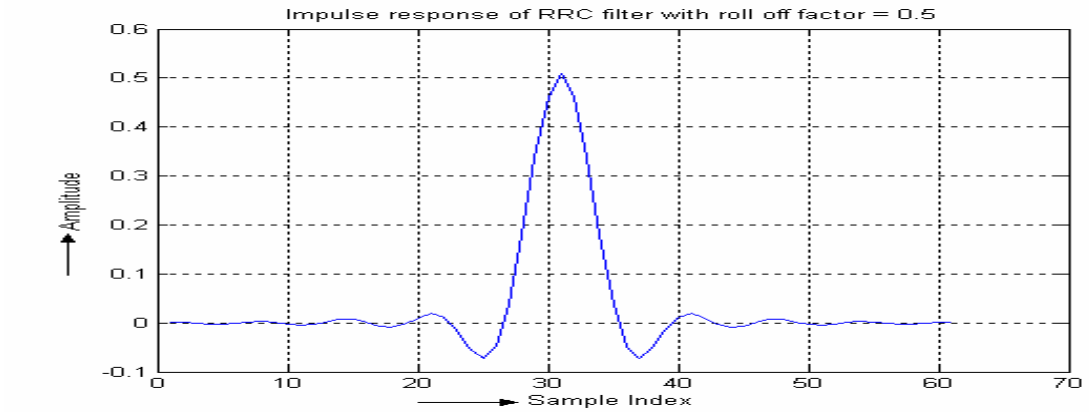


Fig. 4.21.6 (a) *Typical impulse response of a Root Raised Cosine filter with a roll off factor β = 0.5[5 samples /symbol duration have been used to obtain the responses using a digital computer. ±32 sample duration around the normalized peak value of 0.5 is shown]*

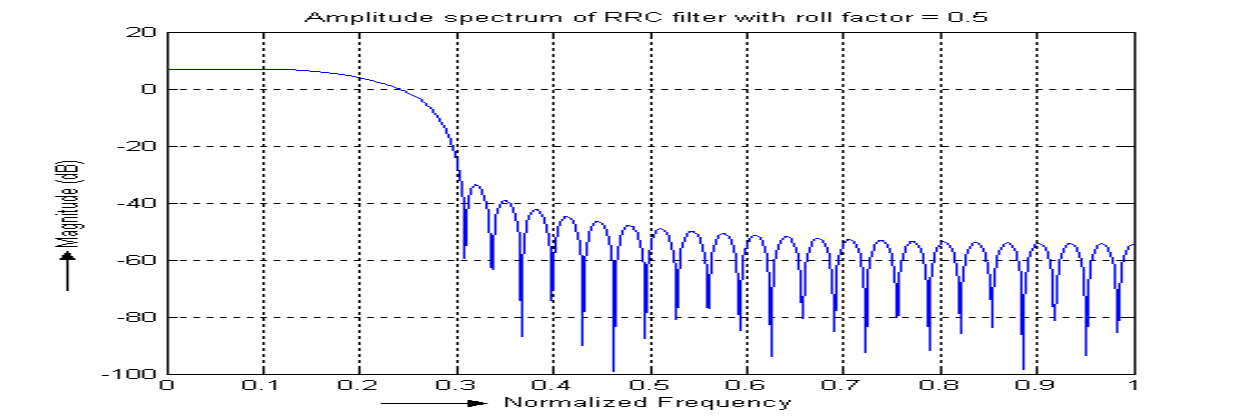


Fig. 4.21.6 (b) *Typical amplitude spectrum of a Raised Cosine filter with a roll off factor β = 0.5. Multiply the normalized frequency values shown above by a factor of 5 to read the frequency normalized to symbol rate.*

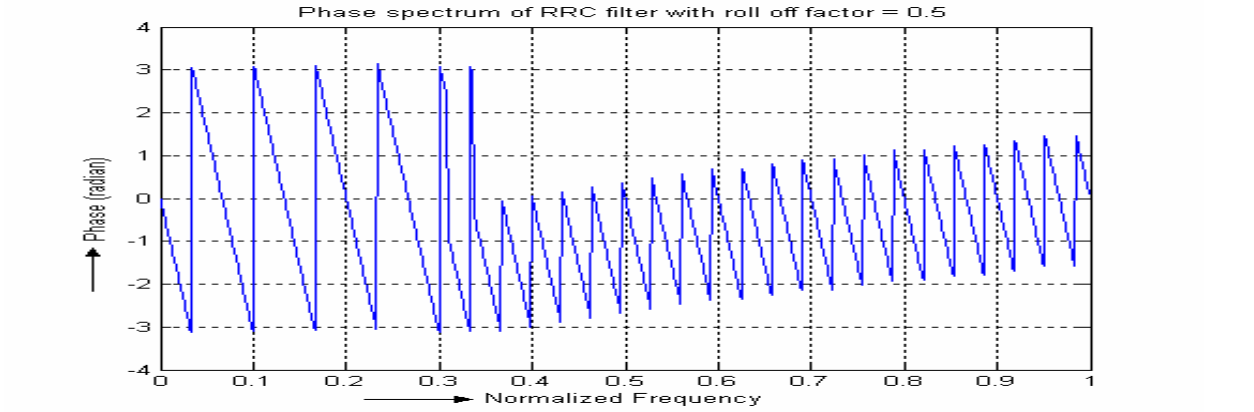


Fig. 4.21.6 (c) *Typical phase spectrum of a Raised Cosine filter with a roll off factor β = 0.5. Multiply the normalized frequency values shown above by a factor of 5 to read the frequency normalized to symbol rate.*

We will mention another practical issue related to the design and implementation of pulse-shaping filters. Usually, an information sequence is presented at the input of a pulse shaping filter in the form of pulses (.e.g. bipolar NRZ) and the width of such a pulse is not negligibly small (as in that case the energy per pulse would be near zero and hence some pulses may even go unrecognized). This finite non-zero width of the pulses causes a distortion in the RC or RRC pulse shape. The situation is somewhat analogous to what happens when we go for flat-top sampling of a band-limited analog signal instead of instantaneous sampling. The finite pulse-width results in amplitude fluctuation and again introduces ISI. It needs a relatively simple pulse magnitude correction which is commonly referred as '$\frac{x}{\sin x}$ amplitude compensation' to restore the quality of shaped

pulses. The transfer function of the RRC shaping filter in the transmitter is scaled appropriately to incorporate the amplitude correction.

Problems

Q4.21.1) Mention possible causes of Inter Symbol Interference (ISI) in a digital communication receiver.

Q4.21.2) Sketch the power spectrum of a long random binary sequence when the two logic levels are represented by +5V and 0V.

Q4.21.3) Sketch the frequency response characteristics of an ideal Nyquist low pass filter.

Q4.21.4) What are the practical difficulties in implementing an ideal Nyquist low pass filter?

Module 5

Carrier Modulation

Lesson 22

Introduction to Carrier Modulation

After reading this lesson, you will learn about

- ***Basic concepts of Narrowband Modulation;***
- ***BER and SER;***
- ***CNR and*** $\frac{E_b}{N_0}$***;***
- ***Performance Requirements;***
- ***Coherent and Non-Coherent Demodulation;***

In the previous module, we learnt about representing information symbols in suitable signal forms. We used the concepts of orthonormal basis functions to represent information-carrying energy-signals. However, we did not discuss how to prepare the signals further, so that we can transmit information over a large distance with minimal transmission power and very importantly, within a specified frequency band. You may know that well-chosen carriers are used for signal transmission over free space. It is also a common knowledge that sinusoids are used popularly as carriers of message or information.

In fact, the sinusoids can be generated easily and the orthogonality between a sine and a cosine carriers of the same frequency can be exploited to prepare or ‘ modulate ‘ information bearing signals so that the information can be received reliably at a distant receiver. In this module, we will discuss about a few basic yet interesting and popular digital modulation schemes, using sinusoids as carriers. The concepts of Gramm-Schmidt Orthogonalization (GSO) are likely to help us, gaining insight into these modulation schemes. The issues of carrier synchronization, which is important for implementing a correlation receiver structure, will also be discussed in this module.

The present lesson will discuss about a few ways to classify digital modulation techniques. We will also introduce some general issues, relevant for appreciating the concepts of digital modulations.

Narrowband Modulation

A modulation scheme is normally categorized as either a narrowband modulation or a wideband modulation. For a linear, time invariant channel model with additive Gaussian noise, if the transmission bandwidth of the carrier-modulated signal is small (typically less than 10%) compared to the carrier frequency, the modulation technique is called a narrowband one. It is easier to describe a narrowband modulation scheme and its performance compared to a wideband modulation scheme, where the bandwidth of the modulated signal may be of the order of the carrier frequency. We will mostly discuss about narrowband digital modulation schemes. Our earlier discussion (Module #4) on equivalent lowpass representation of narrow band pass signals will be useful in this context.

Bandwidth efficient and power efficient modulation schemes

Modulation schemes for digital transmission systems are also categorized as either a) bandwidth efficient or b) power efficient. Bandwidth efficiency means that a modulation scheme (e.g. 8-PSK) is able to accommodate more information (measured in bits/sec) per unit (Hz) transmission bandwidth. Bandwidth efficient modulation schemes are preferred more in digital terrestrial microwave radios, satellite communications and cellular telephony. Power efficiency means the ability of a modulation scheme to reliably send information at low energy per information bit. Some cellular telephony systems and some frequency-hopping spread spectrum communication systems (spread-spectrum systems are wideband type) operate on power-efficient modulation schemes. **Table 5.22.1** names a few digital modulation schemes and some applications. **Table 5.22.2** shows the bandwidth efficiency limits for the modulation techniques.

Some Digital Modulation Schemes	**Representative Applications**
Binary Phase Shift Keying (BPSK)	Telemetry and telecommand
Quaternary Phase Shift Keying (QPSK)	Satellite, Cellular telephony, Digital Video Broadcasting
Octal Phase Shift Keying (8-PSK)	Satellite communications
16-point Quadrature Amplitude Modulation (16-QAM) / 32 QAM	Digital Video Broadcasting, Microwave digital radio links
64-point Quadrature Amplitude Modulation (64-QAM)	Digital Video Broadcasting, MMDS, Set Top Boxes
Frequency Shift Keying (FSK)	Cordless telephony, Paging services
Minimum Shift Keying (MSK)	Cellular Telephony

Table 5.22.1: *Typical applications of some digital modulation schemes*

Modulation Scheme	**Bandwidth Efficiency (in bits/second/Hz)**
Binary Phase Shift Keying (BPSK)	1
Quaternary Phase Shift Keying (QPSK)	2
Octal Phase Shift Keying (8-PSK)	3
16-point Quadrature Amplitude Modulation (16-QAM)	4
32-point Quadrature Amplitude Modulation (32- QAM)	5
256-point Quadrature Amplitude Modulation (256-QAM)	8
Minimum Shift Keying (MSK)	1

Table 5.22.2: *Bandwidth efficiency for a few digital modulation schemes*

BER and SER

The acronym '**BER**' stands for Bit Error Rate. It is a widely used measure to indicate the quality of information, delivered to the receiving end-user. It is defined as the ratio of total number of bits received in error and the total number of bits received over a fairly large session of information transmission. It is an average figure. It is commonly assumed that the same number of bits is received as has been transmitted from the source. BER is a system-level performance. It is an indication of how good a digital communication system has been designed to perform. It also indicates the quality of service the users of a communication system should expect. As we have noted earlier in Module #2, no practical digital communication system ensures zero BER. Interestingly, it is usually sufficient if a system can ensure a BER below an 'acceptable' level. For example, the accepted BER for toll-quality telephone grade speech signal over land-line telephone system is 10^{-05}, while for second generation cellular telephone systems, the BER is usually less than 10^{-3}only. It is a costlier proposition to expect a BER similar to that enjoyed in landline telephone system because the wireless links in a typical mobile telephone system suffers from signal fading. The acceptable BER values are dependent on the type of information. For example, a BER of $10^{-05}$is acceptable for speech signal but is too bad and unacceptable for data service. The BER should be less than 10^{-07}.

'**SER**' stands for Symbol Error Rate. It is also an average figure used to describe the performance of a digital transmission scheme. It is defined as the ratio of total number of symbols detected erroneously in the demodulator and the total number of symbols received by the receiver over a fairly large session of information transmission. Again it is assumed that the same number of symbols is received as has been transmitted by the modulator. Note that this parameter is not necessarily of ultimate interest to the end-users but is important for a communication system designer.

CNR and $\frac{E_b}{N_0}$

Let,
Rate of arrival of information symbols at the input of a modulator = R_s symbols/sec.
Number of different symbols = $M = 2^m$,
Equivalent number of bits grouped to define an information symbol = m
Duration of one symbol = T_s second

Corresponding duration of one bit = $T_b = \frac{T_s}{m}$ second

Double-sided noise power spectral density at the input of an ideal noiseless receiver = $\frac{N_0}{2}$ Watt/Hz

Transmission bandwidth (in the positive frequency region) = B_T Hz

So, the total in-band noise power = N Watt $= \frac{N_0}{2} . (2 \times B_T) = N_o.B_T$ Watt.

We assume that the transmission bandwidth is decided primarily by the modulation scheme. Let us also assume that total signal power received at the receiver input = C Watt. For many digital modulation schemes, this power is distributed over the transmission bandwidth of the modulated signal. The carrier frequency may not always show up in the modulated signal. However, when no modulating data is present, the unmodulated carrier shows up at the receiver with the same power of 'C Watt'.

The ratio of 'C' and the in-band noise power 'N', i.e. $\frac{C}{N}$ is known as the 'carrier to noise power ratio' (CNR). This is a dimensionless quantity. It is often expressed in decibel as:

$$CNR\Big|_{dB} = 10.\log_{10}\left(\frac{C}{N}\right) \qquad 5.22.1$$

Let, $\overline{E_s}$ = Energy received on an average per symbol (Joule).

So, the received power can be expressed as,

$$C = \overline{E_s}.R_s = \frac{\overline{E_s}}{T_s} \qquad 5.22.2$$

If E_b represents the energy (Joule) received per information bit, $\overline{E_s}$ = m. E_b.

Let us now consider another important performance parameter E_b/N_0, defined as:

$$\frac{E_b}{N_0} = \frac{\text{Energy received per information bit}}{\text{one sided power spectral density of in - band noise}} \qquad 5.22.3$$

This is a dimensionless parameter and is often expressed in dB:

$$(E_b/N_0)\Big|_{dB} = 10.\log_{10}\left(\frac{E_b}{N_0}\right) \qquad 5.22.4$$

It should be easy to guess that the CNR and the E_b/N_0 are related.

$$\frac{C}{N} = \frac{C}{N_o B_T} = \frac{mE_b/T_s}{N_o B_T} = \left(\frac{E_b}{N_0}\right).\left(\frac{m}{B_T.T_s}\right) \qquad 5.22.5$$

If the concept of Nyquist filtering with zero-ISI is followed,

$$B_T = \frac{R_s}{2} = \frac{1}{2T_s}$$

and hence, $\frac{C}{N} = (2.m).\left(\frac{E_b}{N_0}\right)$

In logarithmic scale,

$$\left.\mathrm{CNR}\right|_{\mathrm{dB}} = \left.\left(\frac{E_b}{N_0}\right)\right|_{\mathrm{dB}} + 10.\log_{10}(2m) \qquad 5.22.6$$

Performance Requirements

Selection of a modulation scheme for an application is not always straightforward. Following are some preferable requirements from a digital transmission system:

a) Very high data rate should be supported between the end users
b) Signal transmission should take place over least transmission bandwidth
c) BER should be well within the specified limit
d) Minimum transmission power should be used
e) A digital transmission system should not cause interference beyond specified limit and in turn should also be immune to interference from other potential transmission systems
f) The system should be cost-competitive

Some of the above requirements are conflicting in nature and a communication system designer attempt a good compromise of the specified requirements by trading off available design parameters.

I/Q Modulation format

A narrowband digital modulation scheme is often expressed in terms of in-phase signal component (I-signal) and quadrature signal component (Q-signal). This approach is especially suitable for describing 2-dimensional modulation schemes and also for their digital implementation. The signal space describing the modulation format is often referred as I/Q diagrams. Often the two independent signals in I and Q exhibit similar statistical properties and can be generated and processed with similar circuitry.

Coherent and non-coherent Demodulation

The approach of correlation receiver calls for ‘coherent’ demodulation scheme where the carrier references are recovered precisely from the received signal and then used for correlation detection of symbols. This approach ensures best (near-optimal) performance, though may be costly for some modulation schemes. We will primarily discuss about coherent demodulation schemes. A non-coherent demodulation scheme does not require precise carrier reference in the receiver and hence is usually of lower-complexity. Performance is expectedly poorer compared to the corresponding coherent demodulation strategy. We will discuss later about non-coherent demodulation strategy for binary phase shift keying (BPSK) modulation.

Power Spectra of N-B modulated signal

A narrowband carrier modulated signal can be expressed in general as:

$$s(t) = u_I(t)\cos w_c t - u_Q(t)\sin w_c t$$
$$= R_e\left[\tilde{u}(t)e^{jw_c t}\right], \qquad 5.22.7$$

Where $\tilde{u}(t) = u_I(t) + ju_Q(t)$. is the lowpass complex equivalent of the real band pass signal s(t).

Let, $U_B(f)$ denote the power spectrum of the complex low pass equivalent signal. For example, we have earlier seen (also see **Fig. 5.22.1**) that for a random binary sequence, the power spectral density is of the form: $U_B(f) = 2E_b.\sin c^2(T_b f)$

Now, the power spectrum S(f) of the modulated signal is expressed in terms of $U_B(f)$ as:

$$S(f) = \frac{1}{4}\left[U_B(f - f_c) + U_B(f + f_c)\right] \qquad 5.22.8$$

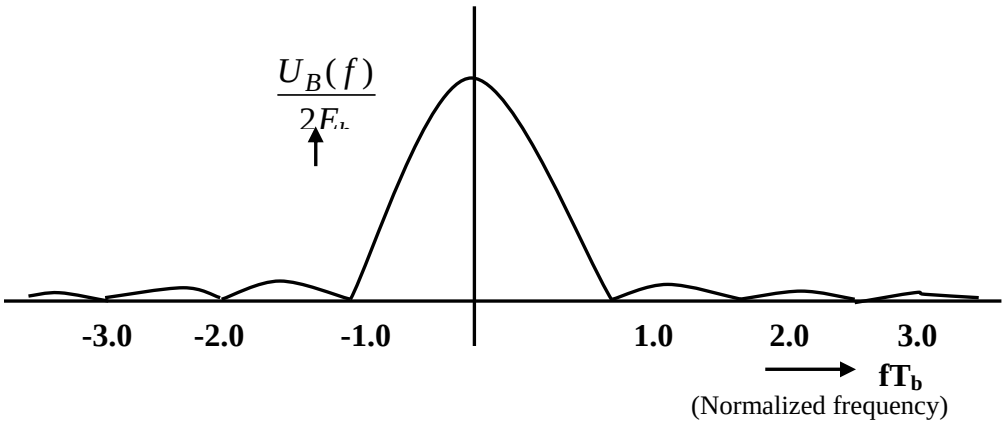


Fig. 5.22.1 *Sketch of power spectrum of a random binary sequence*

It is a good practice and usually simpler to determine $U_B(f)$ and then obtain the desired form of S(f). Towards a general approach, let us introduce a pulse or symbol shaping function g(t), also known as 'weighting function' which, on application to a rectangular time pulse, generates the $u_I(t)$ and $u_Q(t)$. Then we make use of our knowledge of the spectrum of a random sequence of rectangular pulses. As earlier, we assume that all symbols are equally likely at the input to the modulator. The procedure for determining the power spectrum S(f) is summarized below:

a) Consider a narrowband modulation scheme
b) Identify the shaping pulse g(t) , $0 \le t \le T_s$

c) Determine the energy spectral density of g(t)
d) Determine the power spectra of U_I (t) and U_Q(t)
e) Construct U_B(f) and S(f)

Problems

Q5.22.1) What is an acceptable BER for speech signal?

Q5.22.2) If a modulation scheme has 30 different symbols & if the $\frac{E_b}{N_o}$ is 8 dB at the demodulator, determine the CNR at the receiver.

Q5.22.3) Mention four performance metrics for a good digital modulation scheme.

Q5.22.4) Does the narrow band power spectrum of a real band pass signal describe the signal completely?

Q5.22.5) Mention two applications of QAM scheme.

Q5.22.6) Mention two bandwidth efficient modulation schemes.

Module 5

Carrier Modulation

Lesson 23

Amplitude Shift Keying (ASK) and Frequency Shift Keying (FSK) Modulations

After reading this lesson, you will learn about

- ***Amplitude Shift Keying (ASK) Modulation;***
- ***On-off keying;***
- ***Frequency Shift Keying (FSK) Modulation;***
- ***Power spectra of BFSK;***

Amplitude Shift Keying (ASK) Modulation**:**

Amplitude shift keying (ASK) is a simple and elementary form of digital modulation in which the amplitude of a carrier sinusoid is modified in a discrete manner depending on the value of a modulating symbol. Let a group of 'm' bits make one symbol. Hence one can design $M = 2^m$ different baseband signals, $d_m(t)$, $0 \le m \le M$ and $0 \le t \le T$. When one of these symbols modulates the carrier, say, $c(t) = \cos\omega_c t$, the modulated waveform is:

$$s_m(t) = d_m(t).\cos\omega_c t \qquad 5.23.1$$

This is a narrowband modulation scheme and we assume that a large number of carrier cycles are sent within a symbol interval, i.e. $\frac{T}{\left(2\pi/\varpi_c\right)}$ is a large integer. It is obvious that the information is embedded only in the peak amplitude of the modulated signal. So, this is a kind of digital amplitude modulation technique. From another angle, one can describe this scheme of modulation as a one-dimensional modulation scheme where one basis function $\varphi_1(t) = \sqrt{\frac{2}{T}}.\cos\varpi_c t$, defined over $0 \le t \le T$ and having unit energy is used and all the baseband signals are linearly dependent.

Ex. #5.23.1 Let $m = 2$ and $d_0 = 0$, $d_1 = 1$, $d_2 = 2$ and $d_3 = 3$. It is simple to generate such distinct and fixed levels in practice. Further, let us arbitrarily assume the following information to signal mapping: $d_0 \equiv (1,1)$, $d_1 \equiv (1,0)$, $d_2 \equiv (0,1)$ and $d_3 \equiv (0,0)$. So, we have four symbols and the modulated waveforms are:

$$s_0(t) = d_0(t).\sqrt{\frac{2}{T}}.\cos\varpi_c t = 0,\ s_1(t) = d_1(t).\sqrt{\frac{2}{T}}.\cos\varpi_c t = \sqrt{\frac{2}{T}}.\cos\varpi_c t,\ s_2(t) = d_2(t).\sqrt{\frac{2}{T}}.\cos\varpi_c t = 2.\sqrt{\frac{2}{T}}.\cos\varpi_c t \text{ and } s_3(t) = d_3(t).\sqrt{\frac{2}{T}}.\cos\varpi_c t = 3.\sqrt{\frac{2}{T}}.\cos\varpi_c t$$

The signal constellation consists of four points on a straight line. The distances of the points from the origin (signifying zero energy) are 0, 1, 2 and 3 respectively. Note that in this example, no-transmission indicates that 'd_0', i.e. the symbol (1,1) is 'transmitted'. This is not surprising and it also should not give an impression that we are able to transmit 'information' without spending any energy. In fact, it is a bad practice to assign zero energy to a symbol for any good quality carrier modulation scheme because,

in this case, it becomes difficult to recover the basis carrier accurately for coherent demodulation at the receiving end and that ultimately leads to poor SER and BER. Another interesting feature to note is that the modulated symbols have different energy levels, viz. 0, 1, 4 and 9 units. This feature does not make the highest energy symbol d_3 more immune to thermal noise.

On the contrary, the large range of energy level, namely, from '0' to '9' implies that the power amplifier in the transmitter has to have a large linear range of operation – sometime a costly proposition. If the power amplifier goes into its non-linear range while amplifying $s_3(t)$, harmonics of the carrier sinusoid will be generated which will rob some power from $s_3(t)$ away and may interfere with other wireless transmissions in frequency bands adjacent to $\pm 2\omega_c$, $\pm 3\omega_c$, etc. The point to note is that, the Euclidean distance of $s_3(t)$ from the nearest point $s_2(t)$ in the receiver signal space decreases because of amplifier nonlinearity and it means that the receiver will confuse more between $s_3(t)$ and $s_2(t)$ while trying to detect the symbols in presence of noise.

Assuming that all the symbols are equally likely to appear at the input of the modulator, we see that the average energy per symbol ($\overline{E_s}$) is 14/4 = 3.5 unit. This is an important parameter for transmission of digital signals because it is ultimately proportional to the average transmission power. A system designer would always try to ensure low transmission power to save cost and to enhance reliability of the system. So, we see the simple example of ASK modulation of four symbols could be cited in such a way that the signal points were better placed in the constellation diagram such that $\overline{E_s}$ is minimum. ♦

Now, ASK being a form of amplitude modulation, we can say that the bandwidth of the modulated signal will be the same as the bandwidth of the baseband signal. The baseband signal is a long and random sequence of pulses with discrete values. Hence, ASK modulation is not bandwidth efficient. It is implemented in practice when simplicity and low cost are principal requirements.

On-off keying

On-Off Keying (OOK) is a particularly simple form of ASK that represents binary data as the presence or absence of a sinusoid carrier. For example, the presence of a carrier over a bit duration T_b may represent a binary '1' while its absence over a bit duration T_b may represent a binary '0'. This form of digital transmission (OOK) has been commonly used to transmit Morse Codes over a designated radio frequency for telegraph services. As mentioned earlier, OOK is not a spectrally efficient form of digital carrier modulation scheme as the amplitude of the carrier changes abruptly when the data bit changes. So, this mode of transmission is suitable for low or moderate data rate. When the information rate is high, other bandwidth efficient phase modulation schemes are preferable.

Frequency Shift Keying Modulation

Frequency Shift Keying (FSK) modulation is a popular form of digital modulation used in low-cost applications for transmitting data at moderate or low rate over wired as well as wireless channels. In general, an M-ary FSK modulation scheme is a power efficient modulation scheme and several forms of M-ary FSK modulation are becoming popular for spread spectrum communications and other wireless applications. In this lesson, our discussion will be limited to binary frequency shift keying (BFSK).

Two carrier frequencies are used for binary frequency shift keying modulation. One frequency is called the 'mark' frequency (f_2) and the other as the space frequency (f_1). By convention, the 'mark' frequency indicates the higher of the two carriers used. If T_b indicates the duration of one information bit, the two time-limited signals can be expressed as :

$$s_i(t) = \begin{cases} \sqrt{\frac{2E_b}{T_b}} \cos 2\pi f_i t, & 0 \le t \le T_b, i = 1,2 \\ 0, \quad \text{elsewhere.} \end{cases} \qquad 5.23.2$$

The binary scheme uses two carriers and for special relationship between the two frequencies one can also define two orthonormal basis functions as shown below.

$$\varphi_i(t) = \sqrt{\frac{2}{T_b}} \cos 2\pi f_i t \quad ; \quad 0 \le t \le T_b \text{ and } j = 1,2 \qquad 5.23.3$$

If $T_1 = 1/f_1$ and $T_2 = 1/f_2$ denote the time periods of the carriers and if we choose $m.T_1 = n.T_2 = T_b$, where 'm' and 'n' are positive integers, the two carriers are orthogonal over the bit duration T_b. If $R_b = 1/T_b$ denotes the data rate in bits/second, the orthogonal condition implies, $f_1 = m.R_b$ and $f_2 = n.R_b$. Let us assume that $n > m$, i.e. f_2 is the 'mark' frequency. Let the separation between the two carriers be, $\Delta f = f_2 - f_1 = (n-m).R_b$.
Now, the scalar coefficients corresponding to **Eq. (5.23.1)** and **(5.23.3)** are:

$$s_{ij} = \int_0^{T_b} s_i(t)\varphi_j(t)\,dt = \begin{cases} \sqrt{E_b}, i = j \\ 0, \quad i \ne j \end{cases} \quad ; \quad i = 1,2 \text{ and } j = 1,2 \qquad 5.23.4$$

$$\text{i.e.} \quad \left.\begin{aligned} s_{11} = s_{22} = \sqrt{E_b} \\ s_{12} = s_{21} = 0 \end{aligned}\right\} \begin{aligned} i = 1,2 \\ j = 1,2 \end{aligned} \qquad 5.23.5$$

So, the two signal victors can be expressed as:

$$\underline{s}_1 = \begin{bmatrix} \sqrt{E_b} \\ 0 \end{bmatrix} \quad \text{and} \quad \underline{s}_2 = \begin{bmatrix} 0 \\ \sqrt{E_b} \end{bmatrix} \qquad 5.23.6$$

Please note that one can generate an FSK signal without following the above concept of orthogonal carriers and that is often easy in practice. **Fig. 5.23.1** shows a

possible FSK modulated waveform. Notice the waveform carefully and verify if the two carriers are orthogonal. An obvious feature of an FSK modulated signal, analogous to analog FM signal is that envelop of the modulated signal is constant. All modulation schemes which exhibit constant envelope, are preferable for high power digital transmission because, operation of the power amplifier in a non-linear region may not produce considerable harmonics.

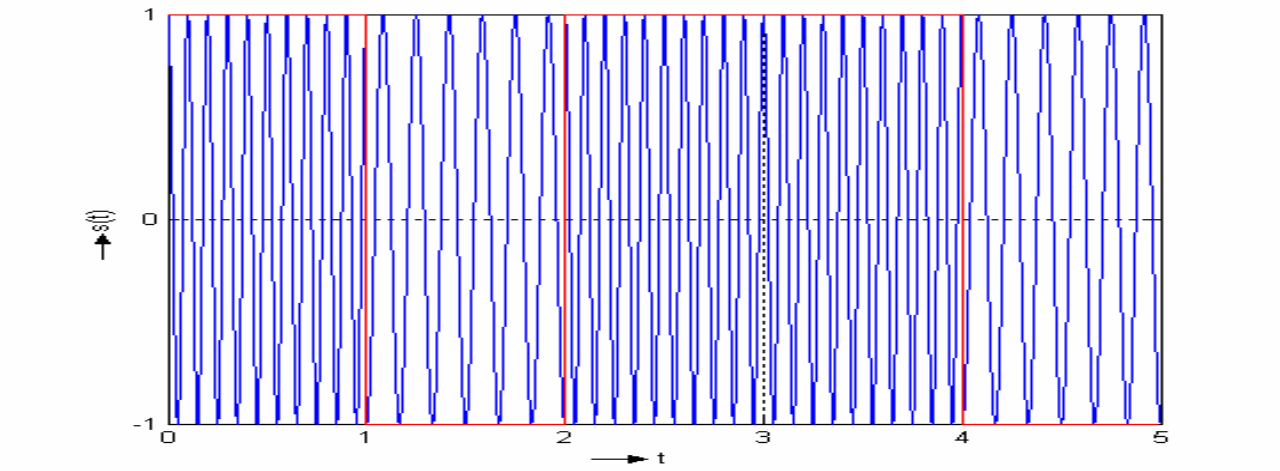


Fig. 5.23.1 *Binary FSK waveform*

Fig. 5.23.2 shows the constellation diagram for binary FSK. **Fig. 5.23.3** shows a conceptual diagram for generating binary FSK modulated signal. Note that the input random binary sequence is represented by '1' and '0' where '0' represents no voltage at the input of the multipliers. A '0' input to the inverter results in a '1' at its output. That is, the inverter, along with the two multipliers and the summing unit, may be thought to behave as a 'switch' which selects output of one of the two oscillators.

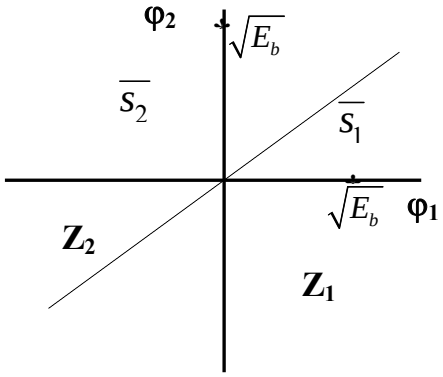


Fig. 5.23.2 *Signal constellation for binary FSK. The diagram also shows the two decision zones, Z_1 and Z_2.*

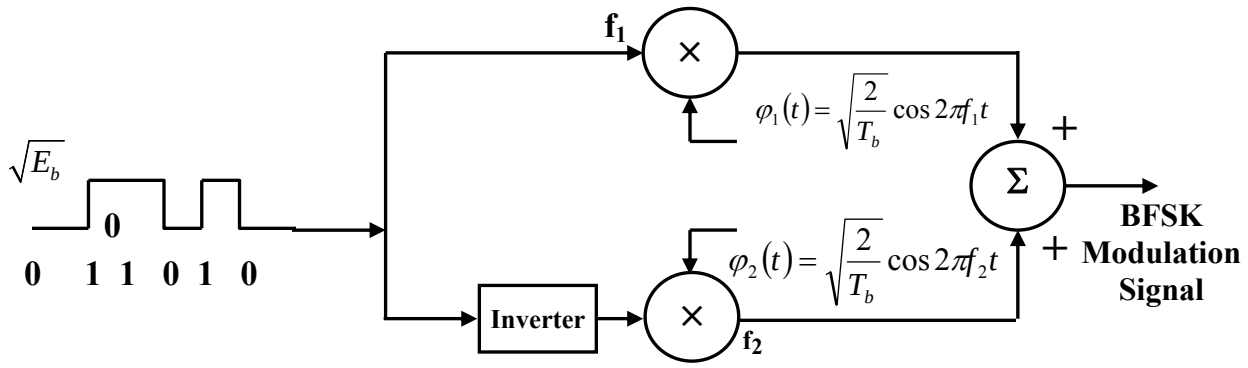


Fig. 5.23.3 *A schematic diagram for BFSK modulation*

In practice, however, this scheme will not work reliably because, the two oscillators being independent, it will be difficult to maintain the orthogonal relationship between the two carrier frequencies. Any relative phase shift among the two oscillators, which may even occur due to thermal drift, will result in deviation from the orthogonality condition. Another disadvantage of the possible relative phase shift is random discontinuity in the phase of the modulated signal during transition of information bits. A better proposition for physical implementation is to use a voltage controlled oscillator (VCO) instead of two independent oscillators and drive the VCO with an appropriate baseband modulating signal, derived from the serial bit stream. The VCO free-running frequency (f_{free}) should be chosen as:

$$f_{free} = \frac{f_1 + f_2}{2} = \frac{m+n}{2}.R_b \qquad 5.23.6$$

Fig. 5.23.4 (a) shows the form of a coherent FSK demodulator, based on the concepts of correlation receiver as outlined in Module #4. The portion on the LHS of the dotted line shows the correlation detector while the RHS shows that the vector receiver reduces to a subtraction unit. Output of the subtraction unit is compared against a threshold of zero to decide about the corresponding transmitted bit.

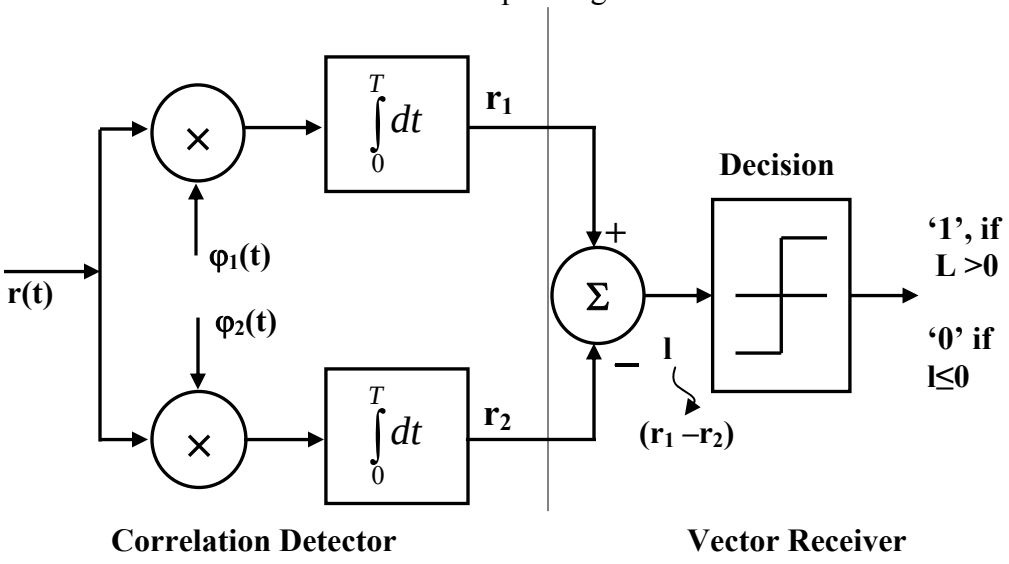


Fig. 5.23.4(a) *Schematic diagram of a coherent BFSK demodulator*

Fig. 5.23.4(b) gives a scheme for non-coherent demodulation of BFSK signal using matched filters. It is often easier to follow this approach than the coherent demodulation scheme without sacrificing error performance.

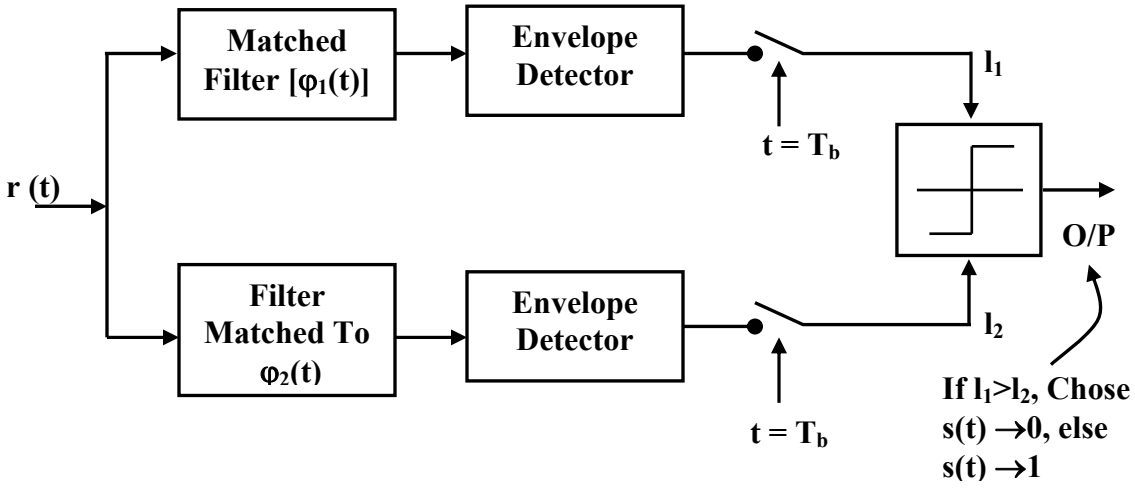


Fig. 5.23.4(b) *General scheme for non-coherent demodulation of BFSK signal using matched filters*

When the issues of performance and bandwidth are not critical and the operating frequencies are low or moderate, a low complexity realization of the demodulator is also possible. Two bandpass filters, one centered at f_1 and the other centered at f_2 may replace the matched filters in **Fig. 5.23.4(b).**

Power Spectra of BFSK

Power spectrum of an FSK modulated signal depends on the choice of f_1 and f_2, i.e. on 'm' and 'n'. When (n-m) is large, we may visualize BFSK as the sum of two OOK signals (see **Fig.5.23.3**) with carriers f_1 and f_2. However, such choice of (n-m) does not a result in bandwidth efficiency.

In the following, we consider n = (m+1), i.e. $f_1 = m.R_b = m/T_b$, $f_2 = f_1 + \frac{1}{T_b} = n.R_b = (m+1).R_b = (m+1)/T_b$ and $f_c = \frac{f_2 + f_1}{2} = f_1 + \frac{1}{2T_b} = \varpi_c / 2\pi = f_{free}$.

Considering the spectrum of a random binary sequence, as we have presented in Module #4, it is easy to see that, ISI can be avoided in detecting the signals for the above choice of f_1 and f_2.

Now, the BFSK modulated signal can be expressed as:

$$s(t) = \sqrt{\frac{2E_b}{T_b}}.\cos\left[w_c t \pm \frac{\pi t}{T_b}\right] = \underbrace{\sqrt{\frac{2E_b}{T_b}}.\cos\left(\frac{\pi t}{T_b}\right)}_{u_I(t)}.\cos w_c t \mp \sqrt{\frac{2E_b}{T_b}}.\sin\left(\frac{\pi t}{T_b}\right)\sin w_c t$$

5.23.8

The '±' sign in the above expression contains information about the information sequence, d(t). It is interesting to note that $u_I(t)$, the real part of the lowpass complex equivalent of the modulated signal s(t) is independent of the information sequence d(t) This portion of s(t) gives rise to a set of two delta functions, each of strength $\frac{E_b}{2T_b}$ and located at $f = +\frac{1}{2T_b} = \frac{f_b}{2}$ and $f = -\frac{1}{2T_b} = -\frac{f_b}{2}$, where $f_b = R_b$.

'$u_Q(t)$', the imaginary part of the lowpass complex equivalent of the modulated signal s(t) can be expressed in terms of a shaping function $g_Q(t)$ as,

$$u_Q(t) = \mp\sqrt{\frac{2E_b}{T_b}}.\sin\left(\frac{\pi t}{T_b}\right) = \mp g_Q(t), \qquad 5.23.9$$

where $g_Q(t) = \sqrt{\frac{2E_b}{T_b}}.\sin\left(\frac{\pi t}{T_b}\right)$, $\quad 0 \le t \le T_b$ 5.23.10

Now, the energy spectral density (esd) of the shaping function $g_Q(t)$ is,

$$\Psi g_Q(f) = \frac{8\,E_b T_b.\cos^2(\pi T_b f)}{\pi^2\left(4T_b{}^2 f^2 - 1\right)^2} \qquad 5.23.11$$

From the above expression, we define the psd of $u_Q(t)$ as:

$$= \frac{esd\ of\ g(t)}{T_b} = \frac{8\,E_b \cos^2(\pi T_b f)}{\pi^2\left(4T_b{}^2 f^2 - 1\right)^2} \qquad 5.23.12$$

As $u_I(t)$ and $u_Q(t)$ are statistically independent of each other, we can now construct the baseband spectrum $U_B(f)$ of the BFSK modulated signal s(t) as:

$$U_B(f) = \frac{E_b}{2T_b}\left[\delta\left(f - \frac{1}{2T_b}\right) + \delta\left(f + \frac{1}{2T_b}\right)\right] + \frac{8E_b\cos^2(\pi T_b f)}{\pi^2\left(4T_b{}^2 f^2 - 1\right)^2}$$

Fig. 5.23.5 shows a sketch (approximate) of the power spectrum of binary FSK signal. In a subsequent lesson (**Lesson #29**), we will discuss about another form of FSK, known as Minimum Shift Keying (MSK), which is operates with minimum possible separation between two frequencies f_1 and f_2.

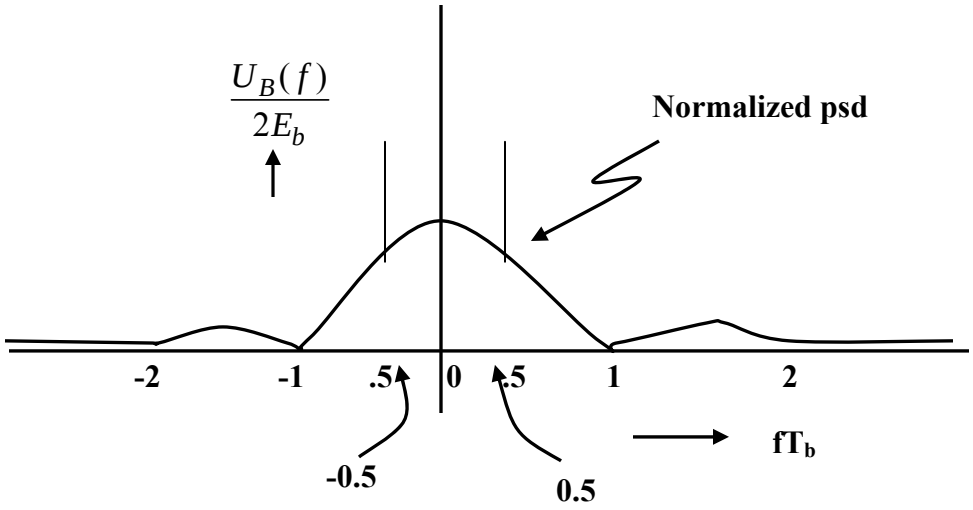


Fig. 5.23.5 *Sketch of the power spectrum of binary FSK signal vs. 'fT_b' when (f_2-f_1) = $1/T_b$.*

Problems

Q5.23.1) Draw the signal constellation of an ASK modulation scheme.

Q5.23.2) How is binary FSK modulation scheme different from binary PSK?

Q5.23.3) Comment if it is a good practice to generate a binary FSK signal by switching an oscillation?

Q5.23.4) Explain in Fig. 5.23.5 why two spikes appear in the spectrum of binary FSK signal for $fT_b = \pm 0.5$?

Module 5

Carrier Modulation

Lesson 24

Binary Phase Shift Keying (BPSK) Modulation

After reading this lesson, you will learn about

- ***Binary Phase Shift Keying (BPSK);***
- ***Power Spectrum for BPSK Modulated Signal;***

Binary Phase Shift Keying (BPSK)

BPSK is a simple but significant carrier modulation scheme. The two time-limited energy signals $s_1(t)$ and $s_2(t)$ are defined based on a single basis function $\varphi_1(t)$ as:

$$s_1(t) = \sqrt{\frac{2E_b}{T_b}}.\cos 2\pi f_c t \quad \text{and} \quad s_2(t) = \sqrt{\frac{2E_b}{T_b}}.\cos\left[2\pi f_c t + \pi\right] = -\sqrt{\frac{2E_b}{T_b}}.\cos 2\pi f_c t \qquad 5.24.1$$

The basis function, evidently, is, $\varphi_1(t) = \sqrt{\frac{2}{T_b}}.\cos 2\pi f_c t$; $0 \le t < T_b$. So, BPSK may be described as a one-dimensional digital carrier modulation scheme. Note that the general form of the basis function is, $\varphi_1(t) = \sqrt{\frac{2}{T_b}}.\cos(2\pi f_c t + \phi)$, where ' Φ ' indicates an arbitrary but fixed initial phase offset. For convenience, let us set $\Phi = 0$.

As we know, for narrowband transmission, $f_c >> \frac{1}{T_b}$. That is, there will be multiple cycles of the carrier sinusoid within one bit duration (T_b). For convenience in description, let us set, $f_c = n \times \frac{1}{T_b}$ (though this is not a condition to be satisfied theoretically).

Now, we see,

$$s_1(t) = \sqrt{E_b}.\varphi_1(t) \quad \text{and} \quad s_2(t) = -\sqrt{E_b}.\varphi_1(t), \qquad 5.24.2$$

The two associated scalars are:

$$s_{11}(t) = \int_0^{T_b} s_1(t).\varphi_1(t)dt = +\sqrt{E_b} \quad \text{and} \quad s_{21} = \int_0^{T_b} s_2(t).\varphi_2(t)dt = -\sqrt{E_b} \qquad 5.24.3$$

Fig. 5.24.1 (a) presents a sketch of the basis function $\varphi_1(t)$ and **Fig. 5.24.1 (b)** shows the BPSK modulated waveform for a binary sequence. Note the abrupt phase transitions in the modulated waveform when there is change in the modulating sequence. On every occasion the phase has changed by 180^0. Also note that, in the diagram, we have chosen to set $\sqrt{\frac{2E_b}{T_b}} = 1$, i.e. $\frac{E_b}{T_b} = \frac{1}{2} = 0.5$, which is the power associated with an unmodulated carrier sinusoid of unit peak amplitude.

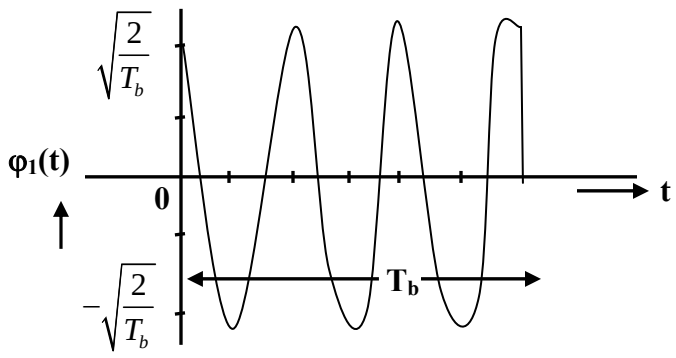


Fig. 5.24.1: (a) *Sketch of the basis function $\varphi_1(t)$ for BPSK modulation*

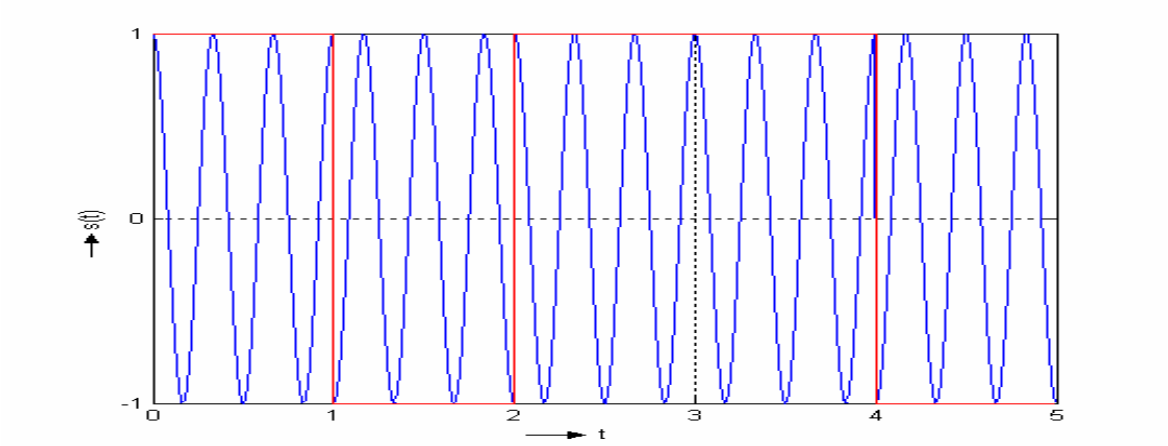


Fig. 5.24.1: (b) *BPSK modulated waveform for the binary sequence 10110. Note that the amplitude has been normalized to ±1, as is a common practice.*

Fig. 5.24.1: (c) shows the signal constellation for binary PSK modulation. The two pints are equidistant from the origin, signifying that the two signals carry same energy.

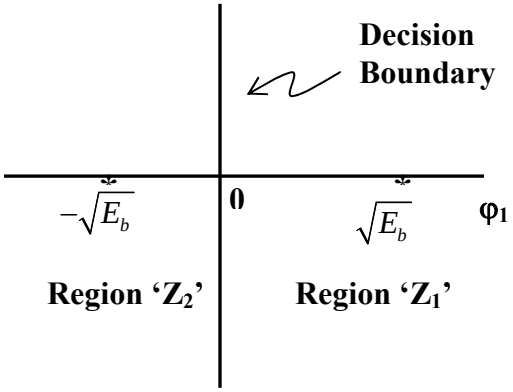


Fig. 5.24.1: (c) *Signal constellation for binary PSK modulation. The diagram also shows the optimum decision boundary followed by a correlation receiver*

Fig. 5.24.2 shows a simple scheme for generating BPSK modulated signal without pulse shaping. A commonly available balanced modulator (such as IC 1496) may be used as the product modulator to actually generate the modulated signal. The basis function $\varphi_1(t)$, shown as the second input to the product modulator, can be generated by an oscillator. Note that the oscillator may work independent of the data clock in general.

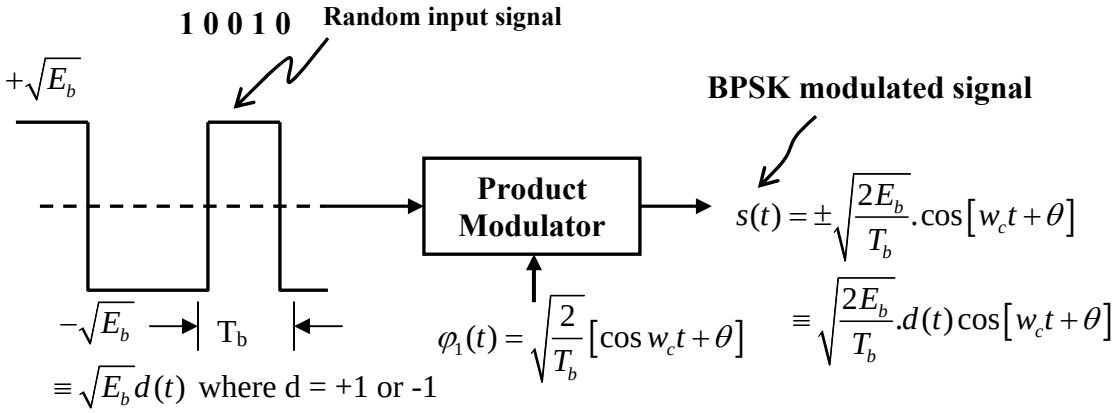


Fig. 5.24.2 *A simple scheme for generating BPSK modulated signal. No pulse-shaping filter has been used.*

Fig. 5.24.3 presents a scheme for coherent demodulation of BPSK modulated signal following the concept of optimum correlation receiver. The input signal r(t) to the demodulator is assumed to be centered at an intermediate frequency (IF). This real narrowband signal consists of the desired modulated signal s(t) and narrowband Gaussian noise w(t). As is obvious, the correlation detector consists of the product modulator, shown as an encircled multiplier, and the integrator. The vector receiver is a simple binary decision device, such as a comparator. For simplicity, we assumed that the basis function phase reference is perfectly known at the demodulator and hence the $\varphi_1(t)$, shown as an input to the product demodulator, is phase-synchronized to that of the modulator. Now it is straightforward to note that the signal at (A) in **Fig. 5.24.3** is:

$$r_A(t) = \left[s(t) + w(t)\right].\sqrt{\frac{2}{T_b}}.\cos\left(w_c t + \theta\right) \qquad 5.24.4$$

The signal at (B) is:

$$r_1 = \sqrt{\frac{2}{T_b}}\int_0^{T_b}\left[d(t).\sqrt{\frac{2E_b}{T_b}}.\cos(w_c t + \theta) + w(t)\right]\cos(w_c t + \theta)dt \qquad 5.24.5$$

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

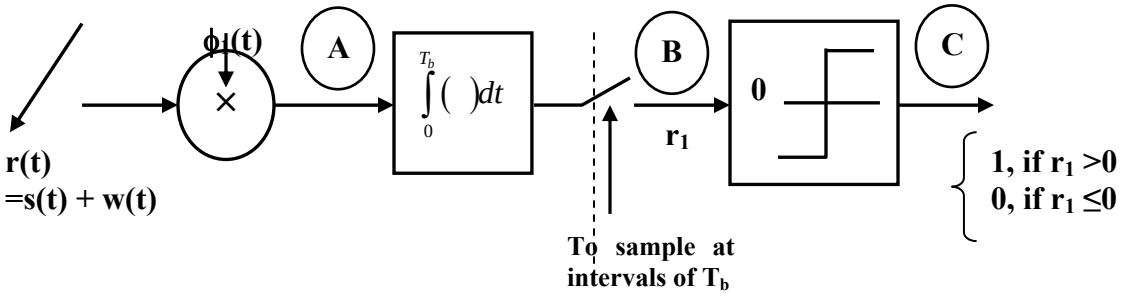


Fig. 5.24.3 *A scheme for coherent demodulation of BPSK modulated signal following the concept of optimum correlation receiver*

Note that the first term in the above expression is the desired term while the second term represents the effect of additive noise. We have discussed about similar noise component earlier in Module #4 and we know that this term is a Gaussian distributed random variable with zero mean. Its variance is proportional to the noise power spectral density. It should be easy to follow that, if d(t) = +1 and the second term in **Eq. 5.24.5** (i.e. the noise sample voltage) is not less than -1.0, the threshold detector will properly decide the received signal as a logic '1'. Similarly, if d(t) = -1 and the noise sample voltage is not greater than +1.0, the comparator will properly decide the received signal as a logic '0'. These observations are based on 'positive binary logic'.

Power Spectrum for BPSK Modulated Signal

Continuing with our simplifying assumption of zero initial phase of the carrier and with no pulse shaping filtering, we can express a BPSK modulated signal as:

$$s(t) = \sqrt{\frac{E_b.2}{T_b}}.d(t)\cos w_c t \text{ , where } \quad d(t) = \pm 1 \qquad 5.24.6$$

The baseband equivalent of s(t) is,

$$\tilde{u}(t) = u_I(t) = \sqrt{\frac{2E_b}{T_b}}.d(t) = \pm g(t), \qquad 5.24.7$$

where $g(t) = \sqrt{\frac{2E_b}{T_b}}$ and $u_Q(t) = 0$.

Now, $u_I(t)$ is a random sequence of $+\sqrt{\frac{2E_b}{T_b}}$ and $-\sqrt{\frac{2E_b}{T_b}}$ which are equi-probable. So, the power spectrum of the base band signal is:

$$\rightarrow U_B(f) = \frac{2E_b.\sin^2(\pi T_b f)}{(\pi T_b f)^2} \qquad = 2.E_b.\sin c^2(T_b f) \qquad 5.24.8$$

Now, the power spectrum S(f) of the modulated signal can be expressed in terms of $U_B(f)$ as:

$$S(f)=\frac{1}{4}\left[U_B(f-f_c)+U_B(f+f_c)\right] \qquad 5.24.9$$

*'**Fig.5.24.4** shows the normalized base band power spectrum of BPSK modulated signal. The spectrum remains the same for arbitrary non-zero initial phase of carrier oscillator.'*

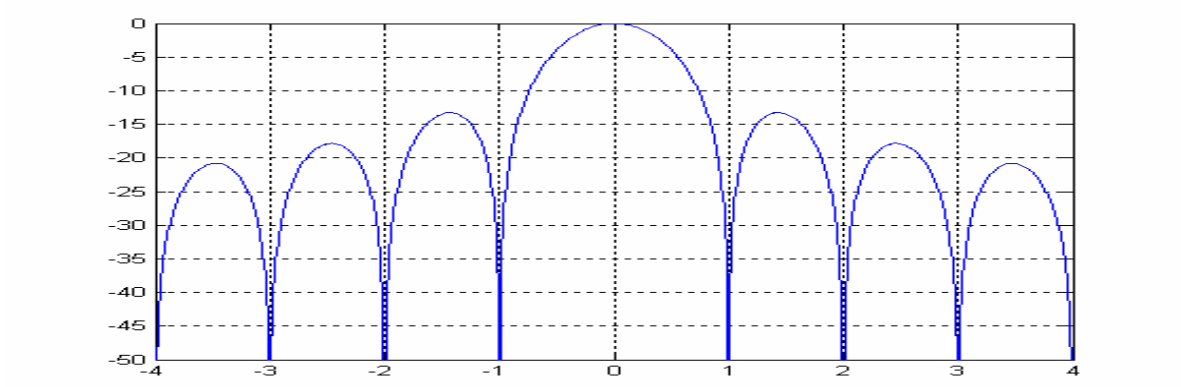


***Fig.5.24.4:** Normalized base band power spectrum of BPSK modulated signal*

Problems

Q5.24.1) Dose BPSK modulated signal have constant envelope?

Q5.24.2) Why coherent demodulation is preferred for BPSK modulation?

Q5.24.3) Do you think the knowledge of an optimum correlation receiver is useful for understanding the demodulation of BPSK signal?

Q5.24.4) Sketch the spectrum of the signal at the output of a BPSK modulator when the modulating sequence is 1, 1, 1, ……

Module 5

Carrier Modulation

Lesson 25

Quaternary Phase Shift Keying (QPSK) Modulation

After reading this lesson, you will learn about

- ***Quaternary Phase Shift Keying (QPSK);***
- ***Generation of QPSK signal;***
- ***Spectrum of QPSK signal;***
- ***Offset QPSK (OQPSK);***
- ***M-ary PSK;***

Quaternary Phase Shift Keying (QPSK)

This modulation scheme is very important for developing concepts of two-dimensional I-Q modulations as well as for its practical relevance. In a sense, QPSK is an expanded version from binary PSK where in a symbol consists of two bits and two orthonormal basis functions are used. A group of two bits is often called a 'dibit'. So, four dibits are possible. Each symbol carries same energy.

Let, E: Energy per Symbol and T: Symbol Duration = 2. T_b, where T_b: duration of 1 bit. Then, a general expression for QPSK modulated signal, without any pulse shaping, is:

$$s_i(t) = \sqrt{\frac{2E}{T}} \cos\left[2\pi f_c t + (2i-1).\frac{\pi}{4}\right] + 0\,; \quad 0 \le t \le T;\ i = 1,2,3,4 \qquad 5.25.1$$

where, $f_c = n.\frac{1}{T} = n.\frac{1}{2T_b}$ is the carrier (IF) frequency.

On simple trigonometric expansion, the modulated signal $s_i(t)$ can also be expressed as:

$$s_i(t) = \sqrt{\frac{2E}{T}}.\cos\left[(2i-1)\frac{\pi}{4}\right].\cos 2\pi f_c t - \sqrt{\frac{2E}{T}}.\sin\left[(2i-1)\frac{\pi}{4}\right].\sin 2\pi f_c t\,;\ 0 \le t \le T \qquad 5.25.2$$

The two basis functions are:

$$\varphi_1(t) = \sqrt{\frac{2}{T}}.\cos 2\pi f_c t\ ;\ 0 \le t \le T \quad \text{and} \quad \varphi_2(t) = \sqrt{\frac{2}{T}}.\sin 2\pi f_c t\ ;\ 0 \le t \le T \qquad 5.25.3$$

The four signal points, expressed as vectors, are:

$$\overline{s_i} = \left\{\sqrt{E}\cos\left[(2i-1)\pi/4\right] - \sqrt{E}\sin\left[(2i-1)\pi/4\right]\right\} = \begin{bmatrix} s_{i1} \\ s_{i2} \end{bmatrix} \quad ; \quad i = 1,2,3,4 \qquad 5.25.4$$

Fig.5.25.1 shows the signal constellation for QPSK modulation. Note that all the four points are equidistant from the origin and hence lying on a circle. In this plain version of QPSK, a symbol transition can occur only after at least $T = 2T_b$ sec. That is, the symbol rate $R_s = 0.5R_b$. This is an important observation because one can guess that for a given binary data rate, the transmission bandwidth for QPSK is half of that needed by BPSK modulation scheme. We discuss about it more later.

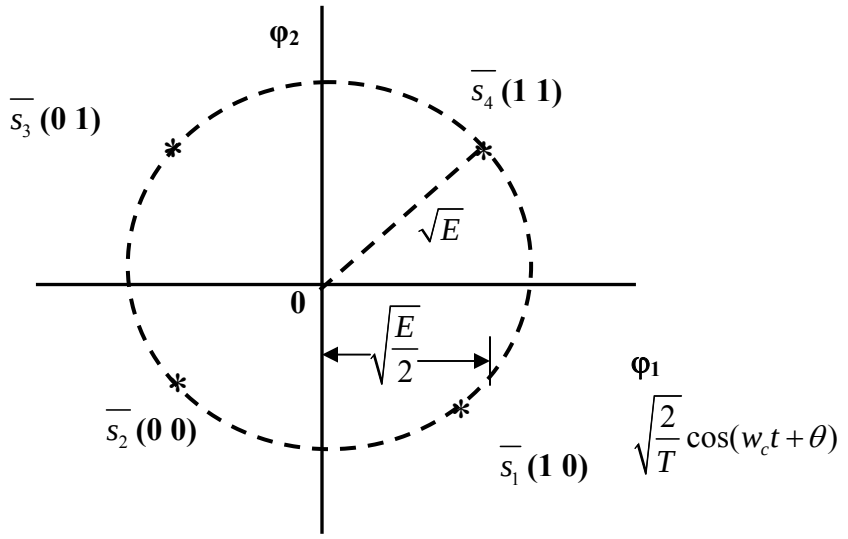


Fig.5.25.1 *Signal constellation for QPSK. Note that in the above diagram θ has been considered to be zero. Any fixed non-zero initial phase of the basis functions is permissible in general.*

Now, let us consider a random binary data sequence: 10111011000110… Let us designate the bits as 'odd' (b_o) and 'even' (b_e) so that one modulation symbol consists of one odd bit and the adjacent even bit. The above sequence can be split into an odd bit sequence (1111001…) and an even bit sequence (0101010…). In practice, it can be achieved by a 1-to-2 DEMUX. Now, the modulating symbol sequence can be constructed by taking one bit each from the odd and even sequences at a time as {(10), (11), (10), (11), (00), (01), (10), …}. We started with the odd sequence. Now we can recognize the binary bit stream as a sequence of signal points which are to be transmitted: {$\overline{s_1}$, $\overline{s_4}$, $\overline{s_1}$, $\overline{s_4}$, $\overline{s_2}$, $\overline{s_3}$, $\overline{s_1}$, …}.

With reference to **Fig.5.25.1**, let us note that when the modulating symbol changes from $\overline{s_1}$ to $\overline{s_4}$, it ultimately causes a phase shift of $\pi^c/2$ in the pass band modulated signal [from $-\pi^c/4$ to $+\pi^c/4$ in the diagram]. However, when the modulating symbol changes from $\overline{s_4}$ to $\overline{s_2}$, it causes a phase shift of π^c in the pass band modulated signal [from $+\pi^c/4$ to $+5\pi^c/4$ in the diagram]. So, a phase change of 0^c or $\pi^c/2$ or π^c occurs in the modulated signal every $2T_b$ sec. It is interesting to note that as no pulse shaping has been used, the phase changes occur almost instantaneously. Sharp phase transitions give rise to significant side lobes in the spectrum of the modulated signal.

Table 5.25.1 summarizes the features of QPSK signal constellation.

Input	Dibit		Phase of QPSK	Coordinates of signal points		
	(b_0)	(b_e)		s_{i1}	s_{i2}	i
$\overline{s_1}$	1	0	$\pi/4$	$+\sqrt{E/2}$	$-\sqrt{E/2}$	1
$\overline{s_2}$	0	0	$3\pi/4$	$-\sqrt{E/2}$	$-\sqrt{E/2}$	2
$\overline{s_3}$	0	1	$5\pi/4$	$-\sqrt{E/2}$	$+\sqrt{E/2}$	3
$\overline{s_4}$	1	1	$7\pi/4$	$+\sqrt{E/2}$	$+\sqrt{E/2}$	4

Table 5.25.1 *Feature summary of QPSK signal constellation*

Fig.5.25.2 shows the QPSK modulated waveform for a data sequence 101110110001. For better illustration, only three carrier cycles have been shown per symbol duration.

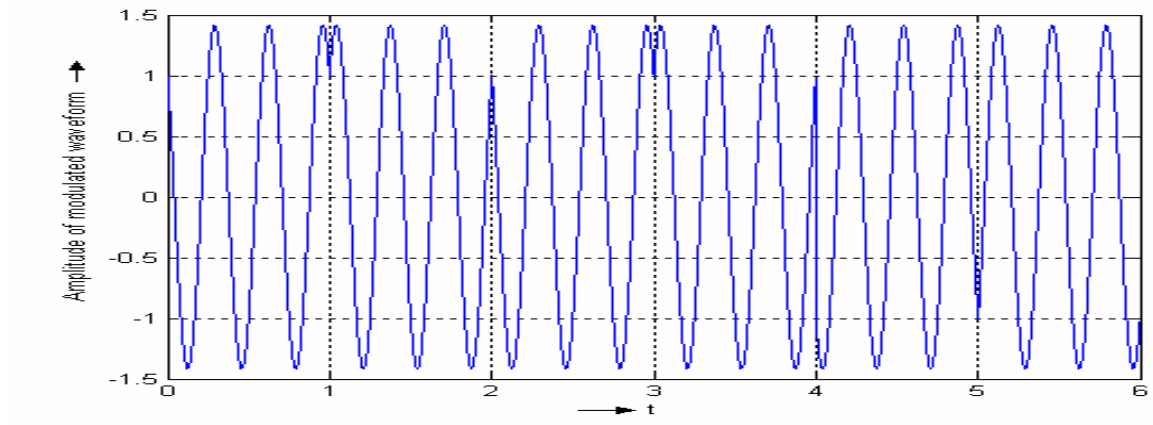


Fig.5.25.2 *QPSK modulated waveform*

Generation of QPSK modulated signal

Let us recall that the time-limited energy signals for QPSK modulation can be expressed as,

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

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

$$= s_{i1}\varphi_1(t) + s_{i2}\varphi_2(t) \qquad \text{i = 1,2,3.4} \qquad 5.25.5$$

The QPSK modulated wave can be expressed in several ways such as:

$$s(t) = \sqrt{E}.d_{odd}(t).\sqrt{\frac{2}{T}}\cos w_c t + \sqrt{E}.d_{even}(t).\sqrt{\frac{2}{T}}\sin w_c t$$

$$= \sqrt{\frac{2E}{T}}.d_{odd}(t)\cos w_c t + \sqrt{\frac{2E}{T}}.d_{even}(t)\sin w_c t$$

$$= \left\{d_{odd}(t).\sqrt{\frac{2E}{T}}\right\}\cos w_c t + \left\{d_{even}(t).\sqrt{\frac{2E}{T}}\right\}\sin w_c t \qquad 5.25.6$$

For narrowband transmission, we can further express s(t) as:

$$\text{s(t)} \equiv u_I(t).\cos w_c t - u_Q(t).\sin w_c t$$

where $\tilde{u}(t) = u_I(t) + ju_Q(t)$ is the complex low-pass equivalent representation of s(t).

One can readily observe that, for rectangular bipolar representation of information bits and without any further pulse shaping,

$$u_I(t) = \sqrt{\frac{2E}{T}}.d_{odd}(t) \text{ and } u_Q(t) = \sqrt{\frac{2E}{T}}.d_{even}(t) \qquad 5.25.7$$

Note that while expressing Eq. 5.25.6, we have absorbed the '-' sign, associated with the quadrature carrier 'sin ω_ct' in $d_{even}(t)$. We have also assumed that $d_{odd}(t) = +1.0$ for 'b_o' $\equiv 1$ while $d_{even}(t) = -1.0$ when $b_e \equiv 1$. This is not a major issue in concept building as its equivalent effect can be implemented by inverting the quadrature carrier.

Fig. 5.25.3(a) shows a schematic diagram of a QPSK modulator following **Eq. 5.25.6**. Note that the first block, accepting the binary sequence, does the job of generation of odd and even sequences as well as the job of scaling (representing) each bit appropriately so that its outputs are s_{i1} and s_{i2} (**Eq.5.25.5**). **Fig. 5.25.3(b)** is largely similar to **Fig. 5.25.3(a)** but is better suited for simple implementation. Close observation will reveal that both the schemes are equivalent while the second scheme allows adjustment of power of the modulated signal by adjusting the carrier amplitudes. Incidentally, both the in-phase carrier and the quadrature phase carriers are obtained from a single continuous-wave oscillator in practice.

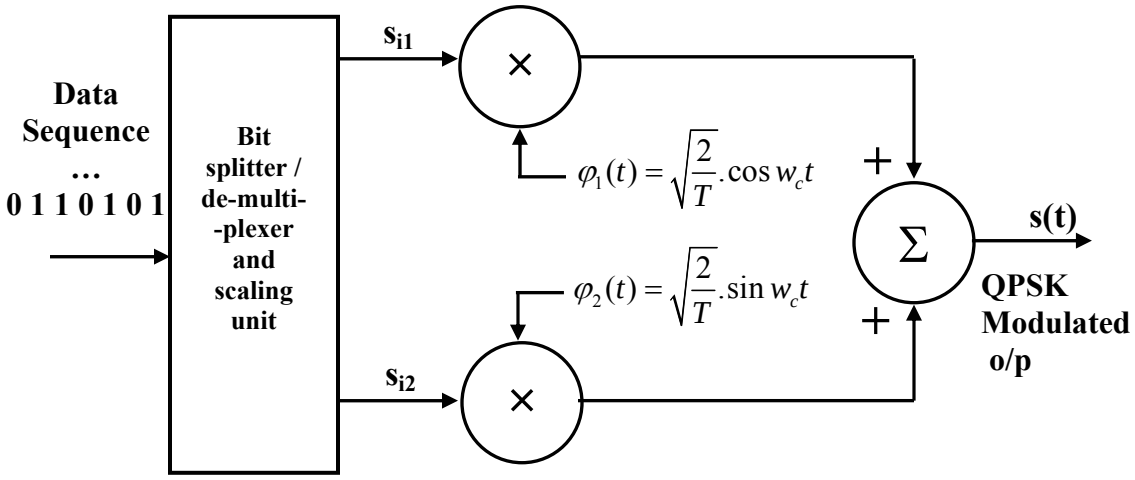


Fig.5.25.3 (a) *Block schematic diagram of a QPSK modulator*

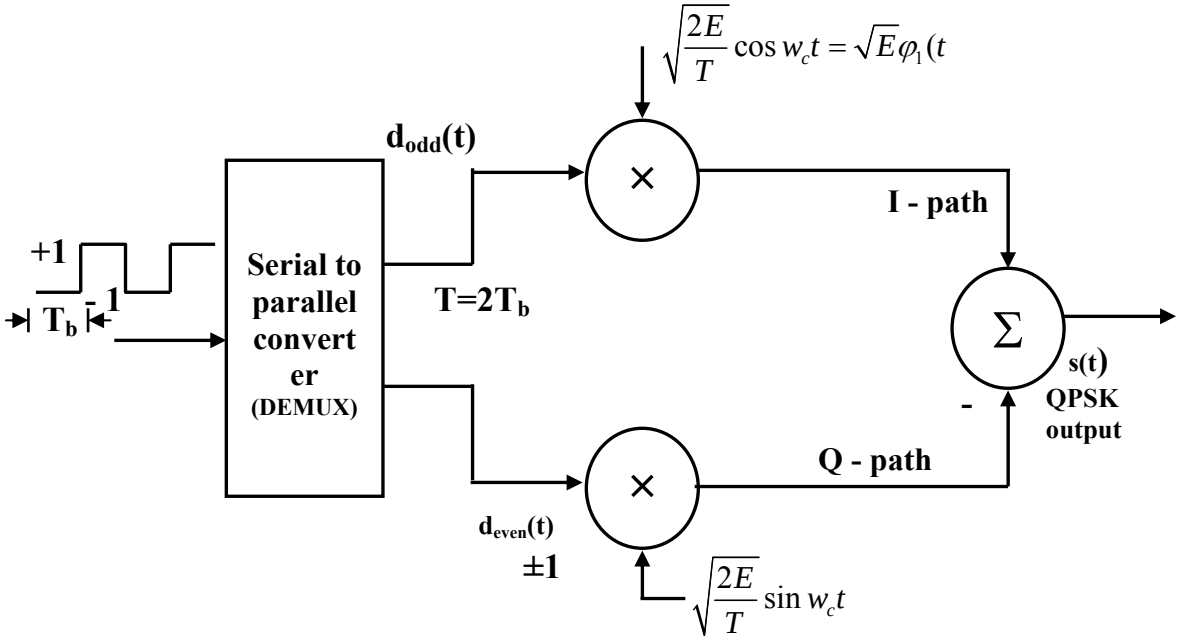


Fig.5.25.3 (b) *Another schematic diagram of a QPSK modulator, equivalent to* ***Fig. 5.23.3(a)*** *but more suitable in practice*

The QPSK modulators shown in **Fig.5.25.3** follow a popular and general structure known as I/Q (In-phase / Quadrature-phase) structure. One may recognize that the output of the multiplier in the I-path is similar to a BPSK modulated signal where the modulating sequence has been derived from the odd sequence. Similarly, the output of the multiplier in the Q-path is a BPSK modulated signal where the modulating sequence is derived from the even sequence and the carrier is a sine wave. If the even and odd bits are independent of each other while occurring randomly at the input to the modulator, the

QPSK modulated signal can indeed be viewed as consisting of two independent BPSK modulated signals with orthogonal carriers.

The structure of a QPSK demodulator, following the concept of correlation receiver, is shown in **Fig. 5.25.4**. The received signal r(t) is an IF band pass signal, consisting of a desired modulated signal s(t) and in-band thermal noise. One can identify the I- and Q- path correlators, followed by two sampling units. The sampling units work in tandem and sample the outputs of respective integrator output every $T = 2T_b$ second, where 'T_b' is the duration of an information bit in second. From our understanding of correlation receiver, we know that the sampler outputs, i.e. r_1 and r_2 are independent random variables with Gaussian probability distribution. Their variance is same and decided by the noise variance while their means are $\pm\sqrt{E/2}$, following our style of representation. Note that the polarity of the sampler output indicates best estimate of the corresponding information bit. This task is accomplished by the vector receiver, which consists of two identical binary comparators as indicated in **Fig.5.25.4.** The output of the comparators are interpreted and multiplexed to generate the demodulated information sequence ($\hat{d}(t)$ in the figure).

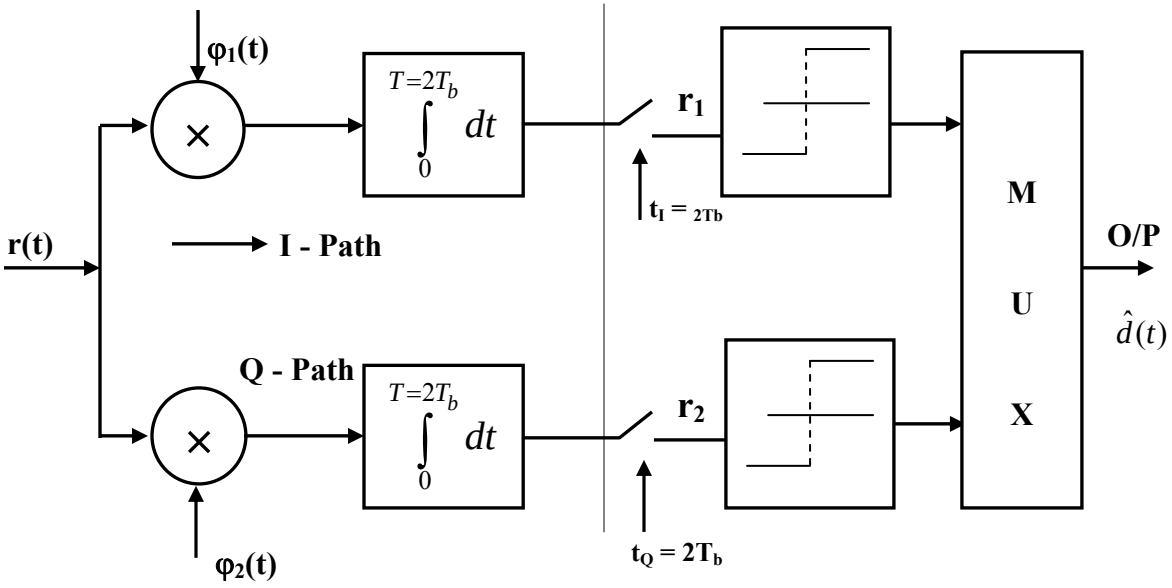


Fig. 5.25.4 *Correlation receiver structure of QPSK demodulator*

We had several ideal assumptions in the above descriptions such as a) ideal regeneration of carrier phase and frequency at the receiver, b) complete knowledge of symbol transition instants, to which the sampling clock should be synchronized, c) linear modulation channel between the modulator output and our demodulator input and so

forth. These issues must be addressed satisfactorily while designing an efficient QPSK modem.

Spectrum of QPSK modulated signal

To determine the spectrum of QPSK modulated signal, we follow an approach similar to the one we followed for BPSK modulation in the previous lesson. We assume a long sequence of random independent bits as our information sequence. Without Nyquist filtering, the shaping function in this case can be written as:

$$g(t) = \sqrt{\frac{E}{T}}; \qquad 0 \le t \le T = 2\ T_b \qquad 5.25.8$$

After some straight forward manipulation, the single-sided spectrum of the equivalent complex baseband signal $\tilde{u}(t)$ can be expressed as:

$$U_B(f) = 2E.\sin c^2(Tf) \qquad 5.25.9$$

Here 'E' is the energy per symbol and 'T' is the symbol duration. The above expression can also be put in terms of the corresponding parameters associated with one information bit:

$$U_B(f) = 4.E_b.\sin c^2(2T_b f) \qquad 5.25.10$$

Fig. 5.25.5 shows a sketch of single-sided baseband spectrum of QPSK modulated signal vs. the normalized frequency (fT_b). Note that the main lobe has a null at $fT_b = 0.5.f.T = 0.5$ because no Nyquist pulse shaping was adopted. The width of the main lobe is half of that necessary for BPSK modulation. So, for a given data rate, QPSK is more bandwidth efficient. Further, the peak of the first sidelobe is not negligibly small compared to the main lobe peak. The side lobe peak is about 12 dB below the main lobe peak. The peaks of the subsequent lobes monotonically decrease. So, theoretically the spectrum stretches towards infinity. As discussed in Module #4, the spectrum is restricted in a practical system by resorting to pulse shaping. The single-sided equivalent Nyquist bandwidth for QPSK = (1/2) symbol rate (Hz) = $1/2T$ (Hz) = $1/4T_b$ (Hz). So, the normalized single-sided equivalent Nyquist bandwidth = ¼ = 0.25. The Nyquist transmission bandwidth of the real pass band modulated signal s(t) = 2 x single-sided Nyquist bandwidth = $1/2T_b$ (Hz) = $1/T$ (Hz) ≡ The symbol rate.

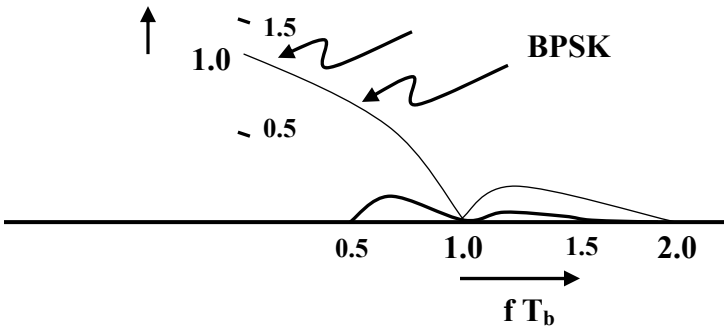


Fig. 5.25.5 *Normalized base band bandwidth of QPSK and BPSK modulated signals*

The actual transmission bandwidth that is necessary = Nyquist transmission bandwidth) x (1 + α) Hz = (1 + α). $1/T$ Hz = (1 + α).R_s Hz, where 'R_s' is the symbol rate in symbols/sec.

Offset QPSK (OQPSK)

As we have noted earlier, forming symbols with two bits at a time leads to change in phase of QPSK modulated signal by as much as 180°. Such large phase transition over a small symbol interval causes momentary but large amplitude change in the signal. This leads to relatively higher sidelobe peaks in the spectrum and it is avoidable to a considerable extent by adopting a simple trick. Offsetting the timing of the odd and even bits by one bit period ensures that the in-phase and quadrature components do not change at the same time instant and as a result, the maximum phase transition will be limited to $\pi^c/2$ at a time, though the frequency of phase changes over a large period of observation will be more. The resultant effect is that the sidelobe levels decrease to a good extent and the demodulator performs relatively better even if the modulation channel is slightly non-linear in behaviour. The equivalent Nuquist bandwidth is not altered by this method. This simple and practical variation of QPSK is known as Offset QPSK.

M–ary PSK

This is a family of two-dimensional phase shift keying modulation schemes. Several bandwidth efficient schemes of this family are important for practical wireless applications.

As a generalization of the concept of PSK modulation, let us decide to form a modulating symbol by grouping 'm' consecutive binary bits together. So, the number of possible modulating symbols is, $M = 2^m$ and the symbol duration $T = m. T_b$. **Fig. 5.25.6** shows the signal constellation for m = 3. This modulation scheme is called as '8-PSK' or 'Octal Phase Shift Keying'. The signal points, indicated by '*', are equally spaced on a circle. This implies that all modulation symbols $s_i(t)$, $0 \leq i \leq (M-1)$, are of same energy 'E'. The dashed straight lines are used to denote the decision zones for the symbols for optimum decision-making at the receiver.

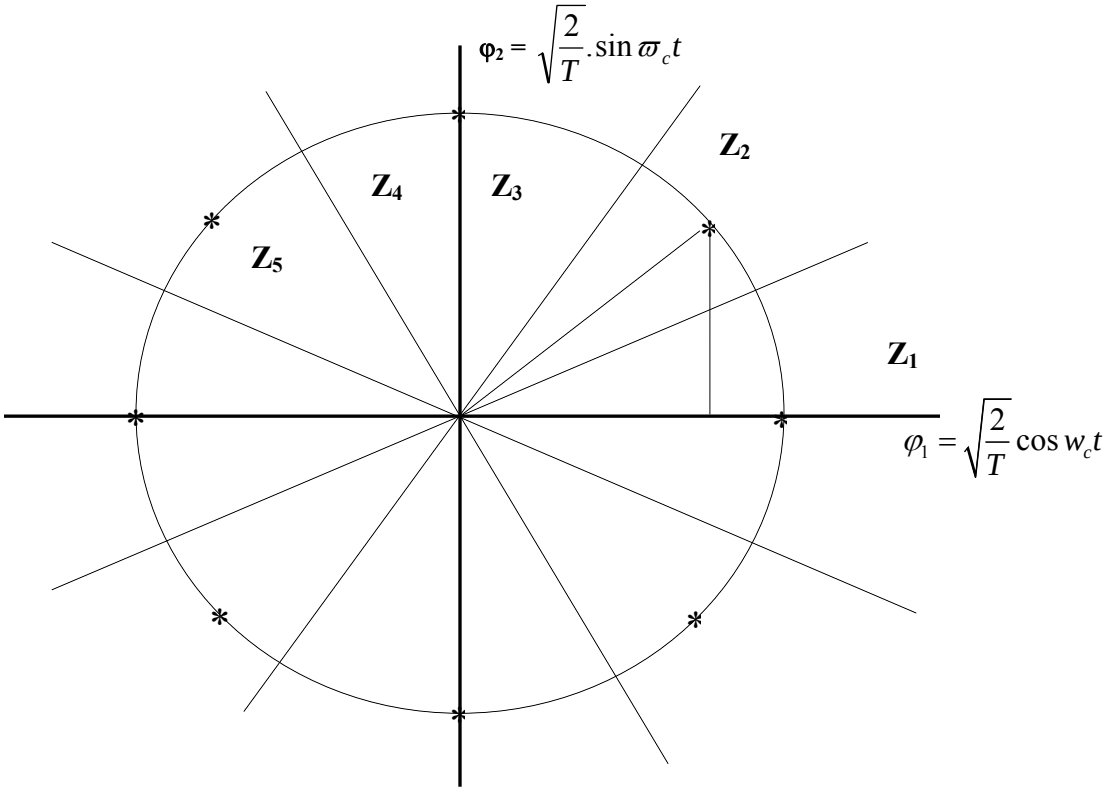


Fig. 5.25.6 *Signal space for 8-PSK modulation*

The two basis functions are similar to what we considered for QPSK, viz.,

$$\varphi_1(t) = \sqrt{\frac{2}{T}} \cos 2\pi f_c t \text{ and } \quad \varphi_2(t) = \sqrt{\frac{2}{T}} \sin 2\pi f_c t \quad ; 0 \le t \le T \qquad 5.25.11$$

The signal points can be distinguished by their angular location:

$$\theta_i = \frac{2\pi_i}{M}; \; i = 0, 1 \ldots\ldots M - 1 \qquad 5.25.12$$

The time-limited energy signals $s_i(t)$ for modulation can be expressed in general as

$$s_i(t) = \sqrt{\frac{2E}{T}} . \cos\left(2\pi f_c t + \frac{2\pi i}{M}\right) \qquad 5.25.13$$

Considering M-ary PSK modulation schemes are narrowband-type, the general form of the modulated signal is

$$s(t) = u_I(t) \cos w_c t - u_Q(t) \sin w_c t \qquad 5.25.14$$

Fig. 5.25.7 shows a block schematic for an M-ary PSK modulator. The baseband processing unit receives information bit stream serially (or in parallel), forms information symbols from groups of 'm' consecutive bits and generates the two scalars s_{i1} and s_{i2} appropriately. Note that these scalars assume discrete values and can be realized in

practice in multiple ways. As a specific example, the normalized discrete values that are to be generated for 8-PSK are given below in **Table 5.25.2**.

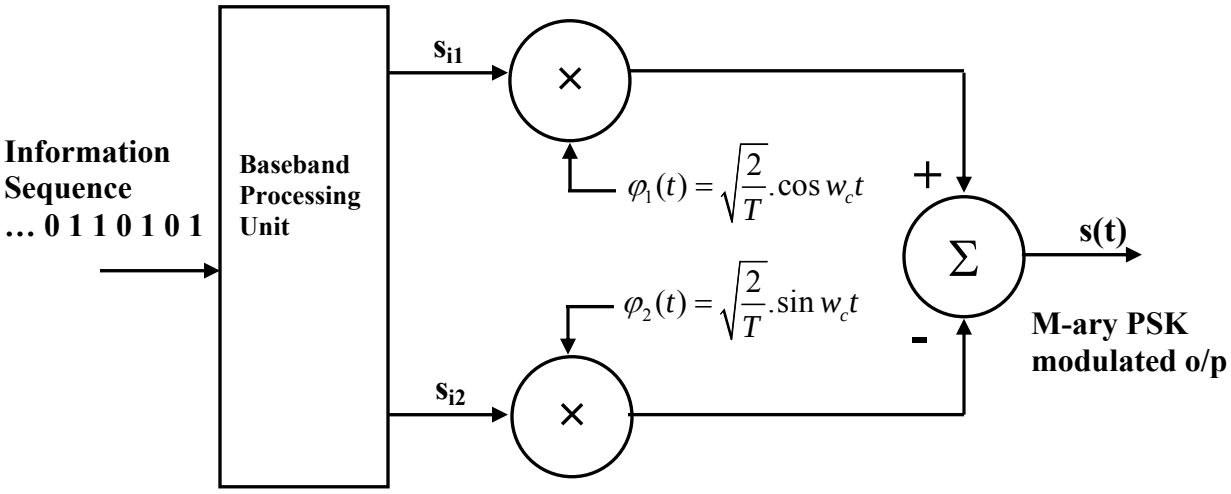


Fig. 5.25.7 *Block schematic diagram of M-ary PSK modulator*

i	0	1	2	3	4	5	6	7
s_{i1}	1	$+\sqrt{1/2}$	0	$-\sqrt{1/2}$	-1	$-\sqrt{1/2}$	0	$+\sqrt{1/2}$
s_{i2}	0	$+\sqrt{1/2}$	1	$+\sqrt{1/2}$	0	$-\sqrt{1/2}$	-1	$-\sqrt{1/2}$

Table 5.25.2 *Normalized scalars for 8-PSK modulation.*

Without any pulse shaping, the $u_I(t)$ and $u_Q(t)$ of **Eq. 5.25.14** are proportional to s_{i1} and s_{i2} respectively. Beside this baseband processing unit, the M-ary PSK modulator follows the general structure of an I/Q modulator.

Fig. 5.25.8 shows a scheme for demodulating M-ary PSK signal following the principle of correlation receiver. The in-phase and quadrature-phase correlator outputs are:

$$r_I = \sqrt{E}\cos\left(\frac{2\pi i}{M}\right) + W_I \quad , \text{i} = 0, 1\ldots \text{M} - 1$$

$$r_Q = -\sqrt{E}\sin\left(\frac{2\pi i}{M}\right) + W_Q \, , \text{i} = 0, 1\ldots \text{M} - 1 \qquad 5.25.15$$

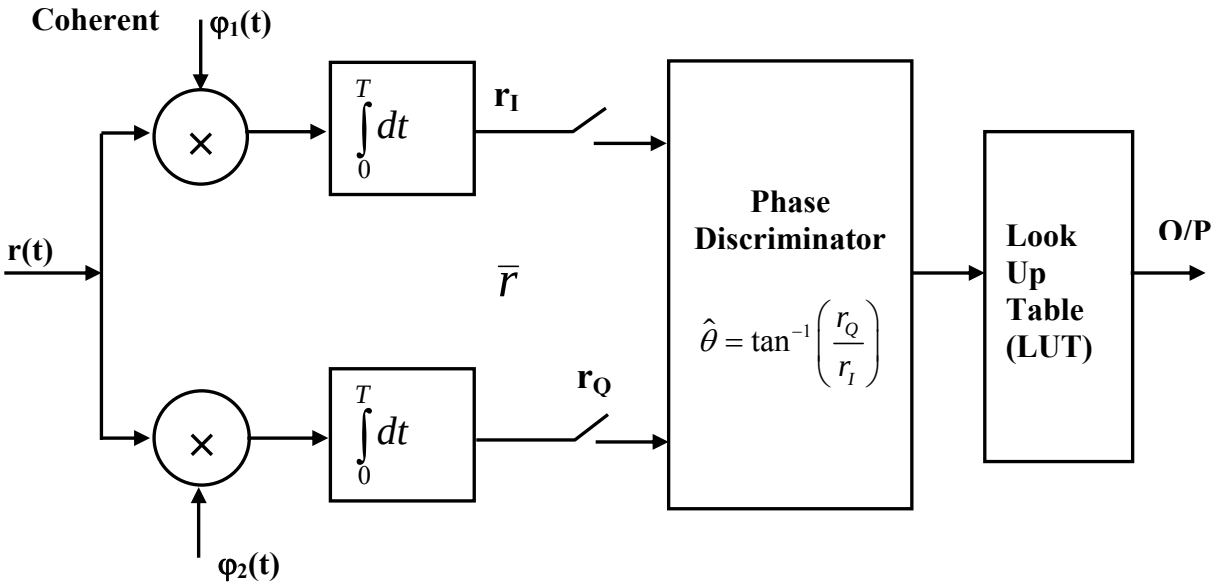


Fig. 5.25.8 *Structure of M-ary PSK demodulator*

W_I represents the in-phase noise sample and W_Q represents the Q-phase noise sample. The samples are taken once every $m.T_b$ sec.

A notable difference with the correlation receiver of a QPSK demodulator is in the design of the vector receiver. Essentially it is a phase discriminator, followed by a map or look-up table (LUT). Complexity in the design of an M-ary PSK modem increases with 'm'.

Problems

Q5.25.1) Write the expression of a QPSK modulated signal & explain all the symbols you have used.

Q5.25.2) What happens to a QPSK modulated signal if the two basis functions are the same that is $\varphi_1(t) = \varphi_2(t)$.

Q5.25.3) Suggest how a phase discriminator can be implemented for an 8-PSK signal?

Module 5

Carrier Modulation

Lesson 26

Differential Encoding and Decoding

After reading this lesson, you will learn about

- ***Differential Encoding of BPSK Modulation (DEBPSK);***
- ***Differential Coding for QPSK;***

Let us consider the processes of quadrature carrier modulation and demodulation and express the output of a quadrature modulator as,

$$s(t) = \text{Re}\{A.\tilde{u}(t).e^{jw_c t}\} = \text{Re}\{[u_i(t) + j u_q(t)].A.e^{jw_c t}\}$$
$$= A.\{u_i(t)\cos w_c t - u_q(t)\sin w_c t\} \quad 5.26.1$$

'A' is a scalar quantity. The two orthogonal carriers which act on $u_i(t)$ and $u_q(t)$ are 'A $\cos w_c t$' and '-A$\sin w_c t$'. The appropriate carriers in the receiver are as shown in **Fig. 5.25.3** which result in correct estimates in absence of noise $\hat{u}_i(t) = u_i(t)$ and $\hat{u}_q(t) = u_q(t)$. Now think what may happen if, say, '-A$\sin w_c t$' multiplies r(t) in the I-arm and 'A $\cos w_c t$' multiplies r(t) in the Q-arm? One can easily see that, the quadrature demodulation structure produces $\hat{u}_i(t) = u_q(t)$ and $\hat{u}_q(t) = u_i(t)$, i.e. the expected outputs have swapped their places. Do we lose information if it happens? No, provided (i) either we are able to recognize that swapping of I-path signal with Q-path signal has occurred (so that we relocate the signals appropriately before delivering to the next stage) or (ii) we devise a scheme which will extract proper signal even when such anomalies occur.

In a practical coherent demodulation scheme, the phase of the carrier is assessed almost continuously against background noise. This issue of phase synchronization is treated separately at some length in **Lesson #31**.

Summarily, precise phase synchronization is a complex process and it increases the cost of a receiver. In any case, sudden change in phase in the transmit oscillator by multiples of 90^0 is never completely ruled out. In view of several such reasons, the approach of differential encoding is followed in practice. Differential encoding and decoding also aid the process of differential demodulation. In the following, we briefly discuss the issue of differential encoding and differential decoding for PSK modulations.

Differential Encoding of BPSK Modulation (DEBPSK)

Let us assume that for an ordinary BPSK modulation scheme, the carrier phase is 0^c when the message bit, 'm_k' is logic '1' and it is π^c if the message bit m_k' is logic '0'.

When we apply differential encoding, the encoded binary '1' will be transmitted by adding 0^c to the current phase of the carrier and an encoded binary '0' will be transmitted by adding π^c to the current phase of the carrier. Thus, relation of the current message bit to the absolute phase of the received signal is modified by differential encoding. The current carrier phase is dependent on the previous phase and the current message bit. For BPSK modulation format, the differential encoder generates an encoded

binary logic sequence $\{d_k\}$ such that, $d_k = 1$ if d_{k-1} and m_k are similar and $d_k = 0$ if d_{k-1} and m_k are not similar.

For completeness, let us assume that the first encoded bit, say, d_0 is '1' while the index 'k' takes values 1, 2, ….**Fig. 5.26.1(a)** shows a block schematic diagram for differential encoding and BPSK modulation. For clarity, we will refer the modulated signal as 'Differentially Encoded and BPSK modulated (DEBPSK)' signal. The level shifter converts the logic symbols to binary antipodal signals of ±1. Note that the encoding logic is simple to implement:

$$d_k = d_{k-1}m_k + \overline{d_{k-1}}.\overline{m_k} \qquad 5.26.2$$

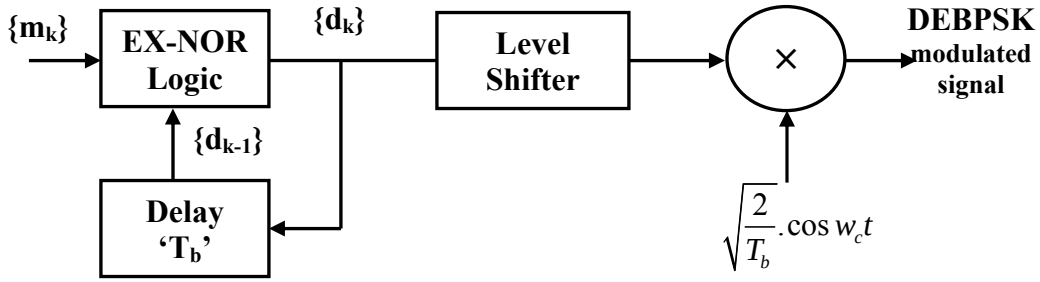
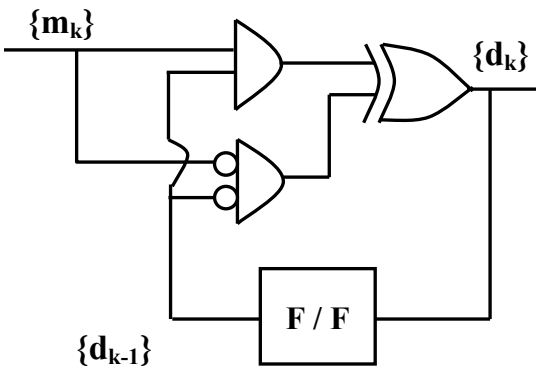


Fig. 5.26.1(a) *Block schematic diagram showing differential encoding for BPSK modulation*

Fig. 5.26.1(b) shows a possible realization of the differential encoder. It also explains the encoding operation for a sample message sequence {1,0,1,1,0,1,0,0,…} highlighting the phase of the modulated carrier.



$\{m_k\}$	:	1 0 1 1 0 1 0 0
$\{d_k\}$	:	1 1 0 0 0 1 1 0 1
Tx.Phase	:	0 0 π π π 0 0 π 0
Demod. Data	:	1 0 1 1 0 1 0 0

Fig. 5.26.1(b) *A realization of the differential encoder for DEBPSK showing the encoding operation for a sample message sequence*

Now, demodulation of a DEBPSK modulated signal can be carried out following the concept of correlation receiver as we have explained earlier in **Lesson #24** (**Fig. 5.24.3**), followed by a differential decoding operation. This ensures optimum (i.e., best achievable) error performance while not requiring a very precise carrier phase recovery scheme. We will refer this combination of correlation receiver with differential encoding-decoding also as the DEBPSK modulation-demodulation scheme.

This is to avoid confusion with another possible scheme of demodulation, which uses a concept of direct differential demodulation. **Fig.5.26.2** explains the differential demodulation scheme for BPSK when differential encoding has been used for BPSK modulation. We refer this demodulator as 'Differential Binary PSK (DBPSK) demodulator'. This is an example of 'non-coherent' demodulation scheme, as it does not require the regenerated carrier for demodulation. So, it is simpler to implement. With reference to the diagram, note that the output x(t) of the multiplier can be expressed without considering the noise component as:

$$x(t) = r(t) \times r(t - T_b) = A_c^2 \left\{ \cos\left[\varpi_c t + \theta + \bar{d}_k . \pi\right] \times \cos\left[\varpi_c \left(t - T_b\right) + \theta + \bar{d}_{k-1} . \pi\right] \right\}$$

$$= \frac{A_c^2}{2} \left\{ \cos\left[\left(\bar{d}_k - \bar{d}_{k-1}\right)\pi\right] + \cos\left[2.\varpi_c t + 2\theta + \left(\bar{d}_k + \bar{d}_{k-1}\right)\pi\right] \right\} \qquad 5.26.3$$

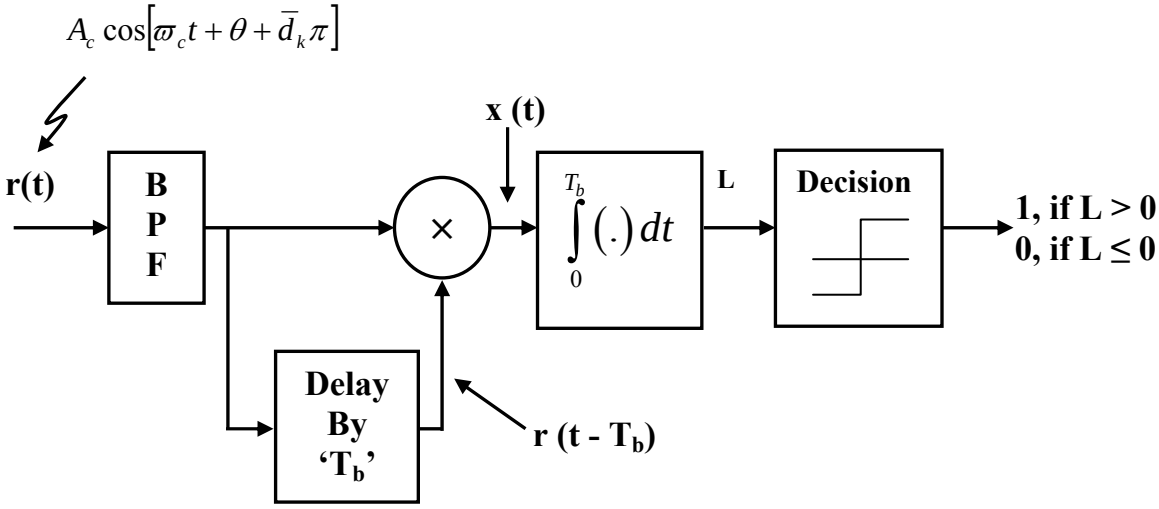


Fig.5.26.2 *Differential demodulation of differentially encoded BPSK modulated signal*

Here, the received signal r(t) is:

$$r(t) = A_c \cos\left[\varpi_c t + \theta + \bar{d}_k \pi\right] \qquad 5.26.4$$

The integrator, acting as a low pass filter, removes the second term of x(t), which is centered around $2\omega_c$ and as a result, the output 'L' of the integrator is $\pm A_c^2 / 2$ which

is used by the threshold detector to determine estimates of the message bit 'm_k' directly. Unlike the DEBPSK demodulation scheme, no separate differential decoding operation is necessary. However, the DBPSK demodulator scheme requires that the IF modulated signal (or equivalently, its time samples) is delayed precisely by 'T_b', the duration of one message bit, and fed to the multiplier. Error performance of the DBPSK demodulation scheme is somewhat inferior to that of ordinary BPSK (or DEBPSK) as one decision error in the demodulator may cause two bits to be in error in quick succession. However, the penalty in error performance is not huge for many applications where lower cost or complexity is preferred more. DBPSK scheme needs about 0.94 dB of additional E_b/N_0 to ensure a BER of 10^{-05}, compared to the optimum and coherent BPSK demodulation scheme.

Differential Coding for QPSK

The four possibilities that are to be considered for designing differential encoder and decoder for QPSK are shown in **Table 5.26.1**, assuming that the I-path carrier in the modulator is A cos w_c t and the Q-path carrier is -Asin w_c t. In any of the four possibilities listed in **Table 5.26.1**, we wish to extract $u_i(t)$ in the I-arm and $u_q(t)$ in the Q-arm using differential encoding and decoding

I-Path Regenerated Carrier	**Q-Path Regenerated carrier**	$\hat{u}_i(t)$	$\hat{u}_q(t)$	**Remarks**
A cos w_c t	-Asin w_c t	$u_i(t)$	$u_q(t)$	Correctly derived
-A cos w_c t	Asin w_c t	$-u_i(t)$	$-u_q(t)$	Inverted
Asin w_c t	-A cos w_c t	$-u_q(t)$	$-u_i(t)$	Swapped and inverted
-Asin w_c t	A cos w_c t	$u_q(t)$	$u_i(t)$	Swapped

Table 5.26.1 *The outputs of the I- and Q- correlators in the demodulator*

One can easily verify from the truth table (**Table 5.26.2**) that,

$$d_{ik} = \overline{u_{ik}} \cdot \overline{u_{qk}} \cdot d_{qk-1} + u_{ik} \cdot \overline{u_{qk}} \cdot d_{qk-1} + u_{ik} \cdot u_{qk} \cdot d_{ik-1} + u_{qk} \cdot u_{ik} \cdot d_{qk-1} \quad 5.26.4$$

$$d_{qk} = \overline{u_{ik}} \cdot \overline{u_{qk}} \cdot d_{qk-1} + u_{ik} \cdot \overline{u_{qk}} \cdot d_{ik-1} + u_{ik} \cdot u_{qk} \cdot \overline{d_{qk-1}} + \overline{u_{ik}} \cdot u_{qk} \cdot d_{ik-1} \quad 5.26.5$$

u_{ik}	u_{qk}	d_{ik-1}	d_{qk-1}	d_{ik}	d_{qk}
0	0	0	0	0	0
0	0	0	0	0	1
0	0	1	0	1	0
0	0	1	1	1	1
0	1	0	0	0	1
0	1	0	1	1	1
0	1	1	0	0	0
0	1	1	1	1	0
1	0	0	0	1	0
1	0	0	1	0	0
1	0	1	0	1	1
1	0	1	1	0	1
1	1	0	0	1	1
1	1	0	1	1	0
1	1	1	0	0	1
1	1	1	1	0	0

Table 5.26.2 *Truth table for differential encoder for QPSK*

A feed forward logic circuit is used in a differential decoder in a DEQPSK scheme, which considers the output from the quadrature demodulator to recover u_{ik} and u_{qk} in the correct arms.

Let us consider a situation, represented by the 11th row of the encoder truth table, i.e., $u_{qk}=0,\ u_{ik}=1,\ d_{ik-1}=1,\ d_{qk-1}=0,\ d_{ik}=0$ and $d_{qk}=1$.

Now, let us consider the four possible phase combination of the quadrature demodulator at the receiver to write the values of e_i's and e_q's (**Table 5.26.3).**

I-Path carrier	**Q-path carrier**	$d_{ik-1}\ \ d_{qk-1}$	$d_{ik}\ \ d_{qk}$	$e_{ik-1}\ e_{qk-1}$	$e_{ik}\ \ e_{qk}$	$u_{ik}\ \ u_{qk}$	**Remarks**
A cos w_c t	-A sin w_c t	1 0	1 1	1 0	1 1	1 0	Phase OK
-A cos w_c t	A sin w_c t	1 0	1 1	0 1	0 0	1 0	Phase inverted
A sin w_c t	-A cos w_c t	1 0	1 1	1 0	0 0	1 0	Data swapped and inverted
-A sin w_c t	A cos w_c t	1 0	1 1	0 1	0 1	1 0	Data swapped

Table 5.26.3 *Four phase combinations of the quadrature demodulator at the receiver*

Related to the values of e_i's and e_q's

The last three columns showing e_i's, e_q's and the desired outputs partially indicate the necessary logic for designing a differential decoder. Continuing in a similar fashion, one can construct the complete truth table of a differential decoder (**Table 5.26.4**).

e_{ik-1}	e_{qk-1}	e_{ik}	e_{qk}	$\hat{u}_{ik}$	$\hat{u}_{qk}$
0	0	0	0	0	0
		0	1	0	1
		1	0	1	0
		1	1	1	1
0	1	0	0	1	0
		0	1	0	0
		1	0	1	1
		1	1	0	1
1	0	0	0	0	1
		0	1	1	1
		1	0	0	0
		1	1	1	0
1	1	0	0	1	1
		0	1	1	0
		1	0	0	1
		1	1	0	0

Table 5.26.4 *Truth Table of the differential decoder for QPSK*

It is easy to deduce that,

$$\hat{u}_{ik} = \overline{e_{ik}} \cdot \overline{e_{qk}} \cdot e_{qk-1} + e_{ik} \cdot \overline{e_{qk}} \cdot \overline{e_{ik-1}} + e_{ik} \cdot e_{qk} \cdot \overline{e_{qk-1}} + \overline{e_{ik}} \cdot e_{qk} \cdot e_{ik-1}$$

$$\hat{u}_{qk} = \overline{e_{ik}} \cdot \overline{e_{qk}} \cdot e_{ik-1} + e_{ik} \cdot \overline{e_{qk}} \cdot e_{qk-1} + e_{ik} \cdot e_{qk} \cdot \overline{e_{ik-1}} + \overline{e_{ik}} \cdot e_{qk} \cdot \overline{e_{qk-1}}$$

Somewhat analogous to DBPSK, one can design a QPSK modulation-demodulation scheme-using differential encoding in the modulator and employing noncoherent differential demodulation at the receiver. The resultant scheme may be referred as DQPSK. The complexity of such a scheme is less compared to a coherent QPSK scheme because precise recovery of carrier phase is not necessary in the receiver. However, analysis shows that the error performance of DQPSK scheme is considerably poorer compared to the coherent DEQPSK or ordinary coherent QPSK receiver. The differential demodulation approach requires more than 2dB extra E_b/N_o to ensure a BER of 10^{-05} when compared to ordinary uncoded QPSK with correlation receiver structure.

Problems

Q5.26.1) Justify the need for differential encoding.

Q5.26.2) Re-design the circuit of Fig. 5.26.1(b). Considering negative logic for the binary digits.

Q5.26.3) Mention two merits and two demerits of QPSK modem compare to a BPSK modem.

Module 5

Carrier Modulation

Lesson 27

Performance of BPSK and QPSK in AWGN Channel

After reading this lesson, you will learn about

- ***Bit Error Rate (BER) calculation for BPSK;***
- ***Error Performance of coherent QPSK;***
- ***Approx BER for QPSK;***
- ***Performance Requirements;***

We introduced the principles of Maximum Likelihood Decision and the basic concepts of correlation receiver structure for AWGN channel in Lesson #19, Module #4. During the discussion, we also derived a general expression for the related likelihood function for use in the design of a receiver. The concept of likelihood function plays a key role in the assessment of error performance of correlation receivers. For ease of reference, we reproduce the expression (Eq.4.19.14) below with usual meaning of symbols and notations:

$$f_{\bar{r}}\left(\bar{r}\middle|m_i\right)=\left(\pi N_0\right)^{-\frac{N}{2}}.\exp\left[-\frac{1}{N_0}\sum_{j=1}^{N}\left(r_j-s_{ij}\right)^2\right] \quad \text{i} = 1,2,\ldots,\text{M.} \qquad 5.27.1$$

Bit Error Rate (BER) calculation for BPSK

We consider ordinary BPSK modulation with optimum demodulation by a correlation receiver as discussed in Lesson #24. **Fig. 5.27.1** shows the familiar signal constellation for the scheme including an arbitrarily chosen received signal point, denoted by vector $\bar{r}$.

The two time limited signals are $s_1(t)=\sqrt{E_b}.\varphi_1(t)$ and $s_2(t)=-\sqrt{E_b}.\varphi_1(t)$, while the basis function, as assumed earlier in Lesson #24 is $\varphi_1(t)=\sqrt{\frac{2}{T_b}}.\cos 2\pi f_c t$; $0 \le t < T_b$. Further, $s_{11} = \sqrt{E_b}$ and $s_{21} = -\sqrt{E_b}$. The two signals are shown in **Fig.5.27.1** as two points $\bar{s}_1$ and $\bar{s}_2$. The discontinuous vertical line dividing the signal space in two halves identifies the two decision zones Z_1 and Z_2. Further, the received vector $\bar{r}$ is the vector sum of a signal point ($\bar{s}_1$ or $\bar{s}_2$) and a noise vector (say, $\bar{w}$) as, in the time domain, r(t) = s(t) + n(t). Upon receiving $\bar{r}$, an optimum receiver makes the best decision about whether the corresponding transmitted signal was $s_1(t)$ or $s_2(t)$.

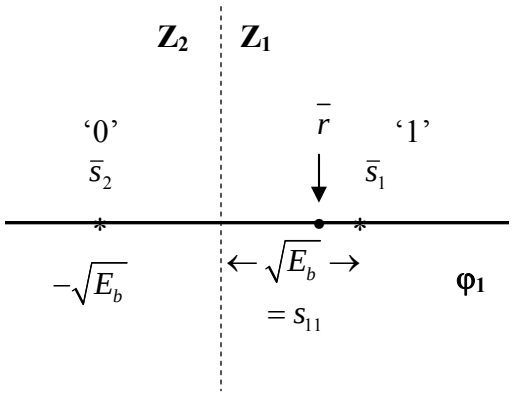


Fig.5.27.1 *Signal constellation for BPSK showing an arbitrary received vector* $\bar{r}$

Now, with reference to **Fig. 5.27.1**, we observe that, an error occurs if

a) $s_1(t)$ is transmitted while $\bar{r}$ is in Z_2 or
b) $s_2(t)$ is transmitted while $\bar{r}$ is in Z_1.

Further, if 'r' denotes the output of the correlator of the BPSK demodulator, we know the decision zone in which $\bar{r}$ lies from the following criteria:

a) $\bar{r}$ lies in Z_1 if $r = \int_0^{T_b} r(t)\varphi_1(t)dt > 0$
b) $\bar{r}$ lies in Z_2 if $r \le 0$

Now, from **Eq. 5.27.1**, we can construct an expression for a Likelihood Function:

$f_r\left(\bar{r}\middle| s_2(t)\right) = f_r(r(t)/\text{message '0' was transmitted})$

From our previous discussion,

$$f_r\left(\bar{r}\middle| s_2(t)\right) = f_r\left(\bar{r}\middle|'0'\right)$$

$$= \frac{1}{\sqrt{\pi N_0}}.\exp\left[-\frac{1}{N_0}\left(r - s_{21}\right)^2\right]$$

$$= \frac{1}{\sqrt{\pi N_0}}.\exp\left\{\frac{-\left[r-\left(-\sqrt{E_b}\right)\right]^2}{N_0}\right\}$$

$$= \frac{1}{\sqrt{\pi N_0}}.\exp\left[-\frac{1}{N_0}\left(r+\sqrt{E_b}\right)^2\right] \qquad 5.27.2$$

$\therefore$ The conditional Probability that the receiver decides in favour of '1' while '0' was transmitted $= \int_0^{\infty} f_r\left(r|0\right)dr = P_e\left(0\right)$, say.

$$\therefore P_e\left(0\right) = \frac{1}{\sqrt{\pi N_0}} \int_0^{\infty} \exp\left[-\frac{1}{N_0}\left(r+\sqrt{E_b}\right)^2\right] dr \qquad 5.27.3$$

Now, putting $\frac{1}{\sqrt{N_0}}\left(r+\sqrt{E_b}\right) = Z$, we get,

$$P_e\left(0\right) = \frac{1}{\sqrt{\pi}} \int_{\sqrt{E_b/N_0}}^{\infty} \exp\left(-Z^2\right) dz$$

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

$$= \frac{1}{2}.erfc\left(\sqrt{\frac{E_b}{N_o}}\right) = Q\left(\sqrt{\frac{2E_b}{N_o}}\right), \qquad \left[\because erf_c(u) = 2Q\left(\sqrt{2}\,u\right)\right] \qquad 5.27.4$$

Fig. 5.27.2 shows the profiles for the error function erf(x) and the complementary error function erfc(x)

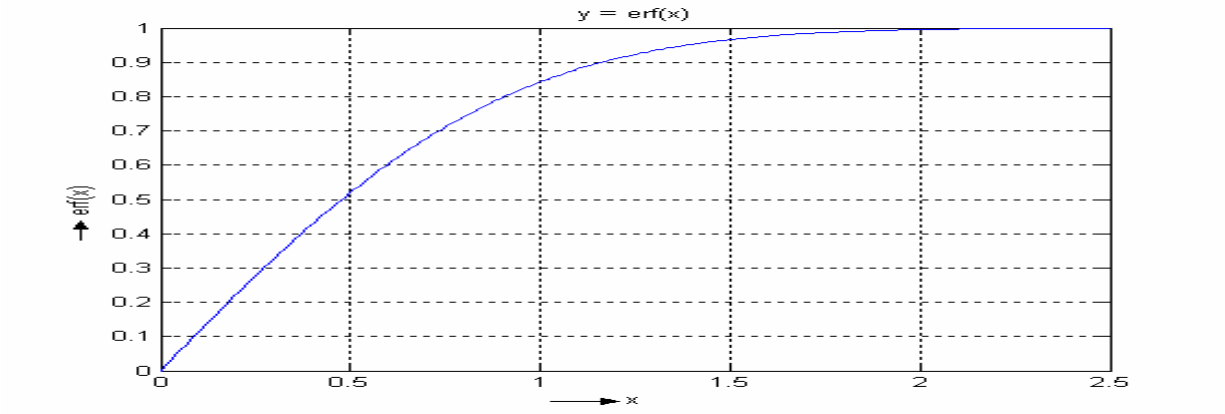


Fig.5.27.2(a) *The 'error function' y = erf(x)*

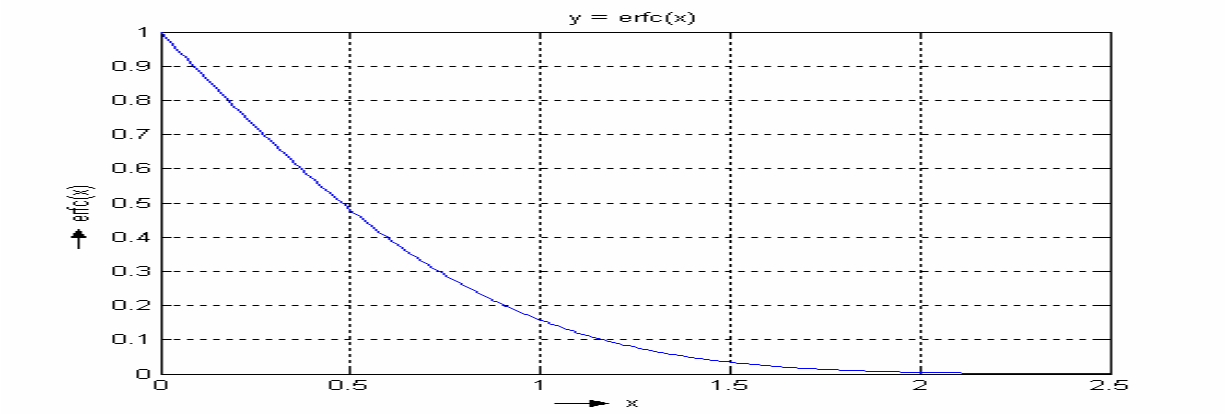


Fig.5.27.2(b) *The 'complementary error function' y = erfc(x)*

Following a similar approach as above, we can determine the probability of error when '1' is transmitted from the modulator, i.e. $P_e(1)$ as,

$$P_e(1) = \frac{1}{2}.erfc\left(\sqrt{\frac{E_b}{N_o}}\right) \qquad 5.27.5$$

Now, as we have assumed earlier, the '0'-s and '1'-s are equally likely to occur at the input of the modulator and hence, the average probability of a received bit being decided erroneously (P_e) is,

$$P_e = \frac{1}{2}.P_e(0) + \frac{1}{2}P_e(1) = \frac{1}{2}.erfc\left(\sqrt{\frac{E_b}{N_o}}\right) \qquad 5.27.6$$

We can easily recognize that P_e is the BER, or equivalently the SER(Symbol error rate) for the optimum BPSK modulator. This is the best possible error performance any BPSK modulator-demodulator can achieve in presence of AWGN. **Fig. 5.27.3** depicts the above relationship. This figure demands some attention as it is often used as a benchmark for comparing error performance of other carrier modulation schemes. Careful observation reveals that about 9.6 dB of E_b/N_o is necessary to achieve a BER of 10^{-05} while an E_b/N_o of 8.4 dB implies an achievable BER of 10^{-04}.

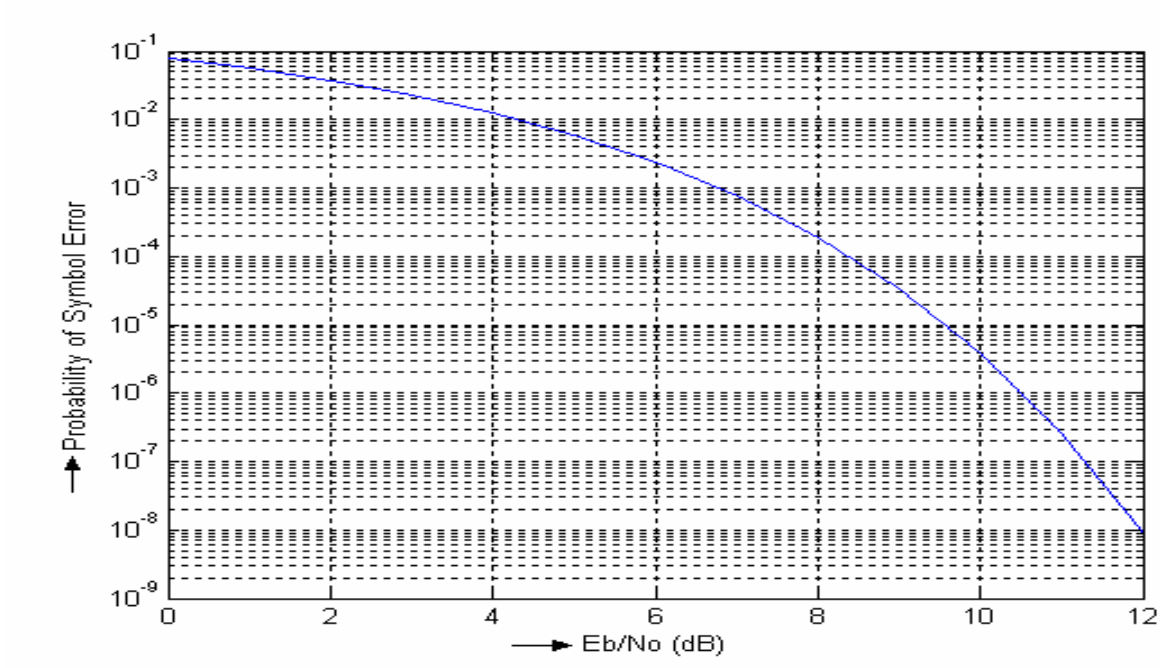


Fig.5.27.3 *Optimum error performance of a Maximum Likelihood BPSK demodulator in presence of AWGN*

Error Performance of coherent QPSK

Fig.5.27.4, drawn following an earlier **Fig.5.25.1**, shows the QPSK signal constellation along with the four decision zones. As we have noted earlier, all the four signal points are equidistant from the origin. The dotted lines divide the signal space in four segments.

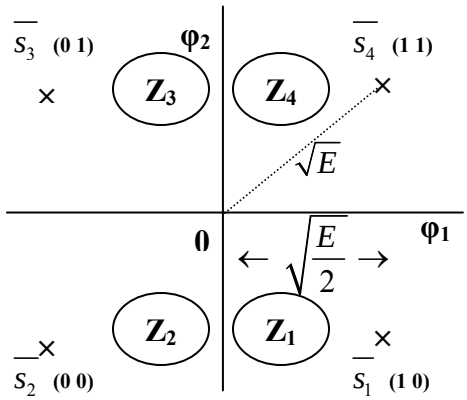


Fig.5.27.4 *QPSK signal constellation showing the four-decision zone*

To recapitulate, a time-limited QPSK modulated signal is expressed as,

$$s_i(t) = \sqrt{\frac{2E}{T}}.\cos\left[(2i-1)\pi/4\right]\cos w_c t - \sqrt{\frac{2E}{T}}.\sin\left[(2i-1)\frac{\pi}{4}\right]\sin w_c t\,,\ 1 \le i \le 4 \qquad 5.27.7$$

The corresponding signal at the input of a QPSK receiver is $r(t) = s_i(t) + w(t)$, $0 \le t \le T$, where 'w(t)' is the noise sample function and 'T' is the duration of one symbol. Following our discussion on correlation receiver, we observe that the received vector $\bar{r}$, at the output of a bank of I-path and Q-path correlators, has two components:

$$r_1 = \int_0^T r(t)\varphi_1(t)\,dt = \sqrt{E}.\cos\left[(2i-1)\frac{\pi}{4}\right] + w_1$$

and $$r_2 = \int_0^T r(t)\varphi_2(t)\,dt = -\sqrt{E}\sin\left[(2i-1)\frac{\pi}{4}\right] + w_2 \qquad 5.27.8$$

Note that if $r_1>0$, it implies that the received vector is either in decision zone Z_1 or in decision zone Z_4. Similarly, if $r_2 > 0$, it implies that the received vector is either in decision zone Z_3 or in decision zone Z_4.

We have explained earlier in Lesson #19, Module #4 that w_1 and w_2 are independent, identically distributed (iid) Gaussian random variables with zero mean and variance $= \frac{N_0}{2}$. Further, r_1 and r_2 are also sample values of independent Gaussian random variables with means $\sqrt{E}\cos\left[(2i-1)\frac{\pi}{4}\right]$ and $-\sqrt{E}\sin\left[(2i-1)\frac{\pi}{4}\right]$ respectively and with same variance $\frac{N_0}{2}$.

Let us now assume that $s_4(t)$ is transmitted and that we have received $\bar{r}$. For a change, we will first compute the probability of correct decision when a symbol is transmitted.

Let, $P_{c_{s4(t)}}$ = Probability of correct decision when $s_4(t)$ is transmitted.

From **Fig.5.27.4,** we can say that,

$P_{c_{s4(t)}}$ = Joint probability of the event that, $r_1 > 0$ and $r_2 > 0$

As $s_4(t)$ is transmitted,

Mean of $r_1 = \sqrt{E}\cos\left[7\frac{\pi}{4}\right] = \sqrt{\frac{E}{2}}$ and

Mean of $r_2 = -\sqrt{E}\sin[7\pi/4] = \sqrt{\frac{E}{2}}$

$$\therefore \quad Pc_{s_4(t)} = \int_0^\infty \frac{1}{\sqrt{\pi N_0}}.\exp\left[-\frac{\left(r_1-\sqrt{\frac{E}{2}}\right)^2}{N_0}\right]dr_1.\int_0^\infty \frac{1}{\sqrt{\pi N_0}}.\exp\left[-\frac{\left(r_2-\sqrt{\frac{E}{2}}\right)^2}{N_0}\right]dr_2 \qquad 5.27.9$$

As r_1 and r_2 are statistically independent, putting $\frac{r_j - \sqrt{\frac{E}{2}}}{\sqrt{N_0}} = Z$, j = 1, 2, we get,

$$Pc_{s_4(t)} = \left[\frac{1}{\sqrt{\pi}} . \int_{-\sqrt{\frac{E}{2N_0}}}^{\infty} \exp\left(-Z^2\right) dz \right]^2 \qquad 5.27.10$$

Now, note that, $\frac{1}{\sqrt{\pi}} \int_{-a}^{\infty} e^{-x^2} dx = 1 - \frac{1}{2} erfc(a)$. 5.27.11

$$\therefore Pc_{s_4(t)} = \left[1 - \frac{1}{2} erfc\left(\sqrt{\frac{E}{2N_0}} \right) \right]^2$$

$$= 1 - erfc\left(-\sqrt{\frac{E}{2N_0}} \right) + \frac{1}{4} . erfc^2 \left(\sqrt{\frac{E}{2N_0}} \right) \qquad 5.27.12$$

So, the probability of decision error in this case, say, $P_{e_{s4(t)}}$ is

$$= Pe_{s_4(t)} = 1 - Pc_{s_4(t)} = erfc\left(\sqrt{\frac{E}{2N_0}} \right) - \frac{1}{4} erfc^2 \left(\sqrt{\frac{E}{2N_0}} \right) \qquad 5.27.13$$

Following similar argument as above, it can be shown that $P_{e_{s1(t)}} = P_{e_{s2(t)}} = P_{e_{s3(t)}} = P_{e_{s4(t)}}$.

Now, assuming all symbols as equally likely, the average probability of symbol error

$$= Pe = 4 \times \frac{1}{4} \left[erfc\left(\sqrt{\frac{E}{2N_0}} \right) - \frac{1}{4} erfc^2 \left(\sqrt{\frac{E}{2N_0}} \right) \right] \qquad 5.27.14$$

A relook at **Fig.5.27.2(b)** reveals that the value of erfc(x) decreases fast with increase in its argument. This implies that, for moderate or large value of $E_b/N_{0,}$ the second term on the R.H.S of **Eq.5.27.14** may be neglected to obtain a shorter expression for the average probability of symbol error, P_e:

$$P_e \cong erfc\left(\sqrt{\frac{E}{2N_o}} \right) = erfc\left(\sqrt{\frac{2.E_b}{2.N_o}} \right) = erfc\left(\sqrt{\frac{E_b}{N_o}} \right) \qquad 5.27.15$$

Fig.5.27.5 shows the average probabilities for symbol error for BPSK and QPSK. Note that for a given E_b/N_o, the average symbol error probability for QPSK is somewhat more compared to that of BPSK.

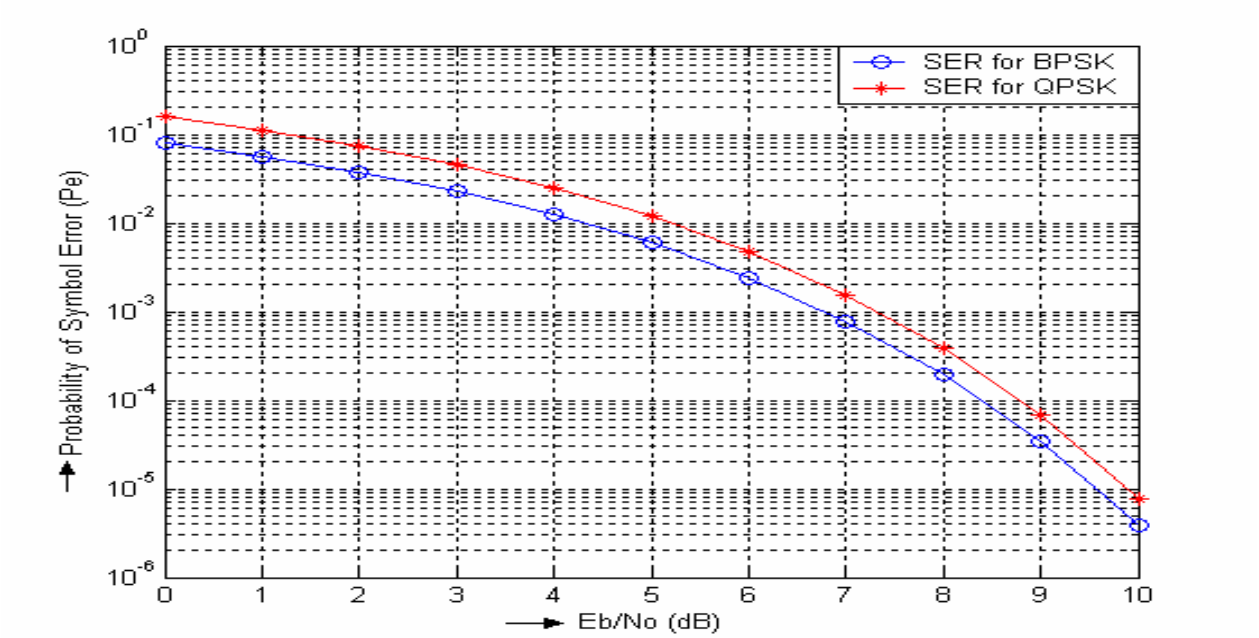


Fig.5.27.5 *SER for BPSK and QPSK*

Approx BER for QPSK:

When average bit error rate, BER, for QPSK is of interest, we often adopt the following approximate approach:

By definition, $Av.\,BER = \lim_{N_{Tot.}\to\infty} \dfrac{\text{No of erroneous bits}}{\text{Total No. of bits transmitted}\left(N_{Tot}\right)}$

Now, let us note that, one decision error about a QPSK symbol may cause an error in one bit or errors in both the bits that constitute a symbol. For example, with reference to **Fig.5.27.4**, if $s_4(t)$ is transmitted and it is wrongly detected as $s_1(t)$, information symbol (1,1) will be misinterpreted as (1,0) and this actually means that one bit is in error. On the contrary, if $s_4(t)$ is misinterpreted as $s_2(t)$, this will result in two erroneous bits. However, the probability of $s_4(t)$ being wrongly interpreted as $s_2(t)$ is much less compared to the probability that $s_4(t)$ is misinterpreted as $s_1(t)$ or $s_3(t)$. We have tacitly taken advantage of this observation while representing the information symbols in the signal space. See that two adjacent symbols differ in one bit position only. This scheme, which does not increase the complexity of the modem, is known as gray encoding. It ensures that one wrong decision on a symbol mostly results in a single erroneous bit. This observation is good especially at moderate or high E_b/N_o. Lastly, the

total number of message bits transmitted over an observation duration is twice the number of transmitted symbols.

$$\therefore Av.\,BER = \frac{1}{2} \times \lim_{Ns \to \infty} \frac{\text{no. of erroneous symbols}}{\text{Total no. of symbols (Ns)}} = \frac{1}{2} \times Pe \cong \frac{1}{2} erfc\left(\sqrt{\frac{E_b}{N_0}}\right) \qquad 5.27.16$$

That is, the BER for QPSK modulation is almost the same as the BER that can be achieved using BPSK modulation. So, to summarize, for a given data rate and a given channel condition (E_b/N_o), QPSK as as good as BPSK in terms of error performance while it requires half the transmission bandwidth needed for BPSK modulation. This is very important in wireless communications and is a major reason why QPSK is widely favoured in digital satellite communications and other terrestrial systems.

Problems

Q5.27.1) Suppose, 1 million information bits are modulated by BPSK scheme & the available $\frac{E_b}{N_o}$ is 6.0 dB in the receiver.

Q5.27.2) Determine approximately how many bits will be erroneous at the output of the demodulator.

Q5.27.3) Find the same if QPSK modulator is used instead of BPSK.

Q5.27.4) Mention three situations which can be describe suitable using error function.

Module 5

Carrier Modulation

Lesson 28

Performance of ASK and binary FSK in AWGN Channel

After reading this lesson, you will learn about

- ***Error Performance of Binary FSK;***
- ***Performance indication for M-ary PSK;***
- ***Approx BER for QPSK;***
- ***Performance Requirements;***

Error Performance of Binary FSK

As we discussed in Lesson #23, BFSK is a two-dimensional modulation scheme with two time-limited signals as reproduced below:

$$s_i(t) = \begin{cases} \sqrt{\frac{2E_b}{T_b}} \cos 2\pi f_i t, & 0 \le t \le T_b, i = 1,2 \\ 0, \quad \text{elsewhere.} \end{cases} \qquad 5.28.1$$

We assume appropriately chosen 'mark' and 'space' frequencies such that the two basis functions are orthonormal:

$$\varphi_i(t) = \sqrt{\frac{2}{T_b}} \cos 2\pi f_i t \quad ; \quad 0 \le t \le T_b \text{ and } j = 1,2 \qquad 5.28.2$$

We will consider the coherent demodulator structure [**Fig. 5.28.1(b)**] so that we can apply similar procedure as in Lesson #27 and obtain the optimum error performance for binary FSK. For ease of reference, the signal constellation for BFSK is reproduced in **Fig. 5.28.1(a)**. As we can see, the two signal vectors are $\overline{s_1} = \begin{bmatrix} \sqrt{E_b} \\ 0 \end{bmatrix}$ and $\overline{s_2} = \begin{bmatrix} 0 \\ \sqrt{E_b} \end{bmatrix}$ while the two associated scalars are $s_{11} = s_{22} = \sqrt{E_b}$. The decision zones are shown by the discontinuous line.

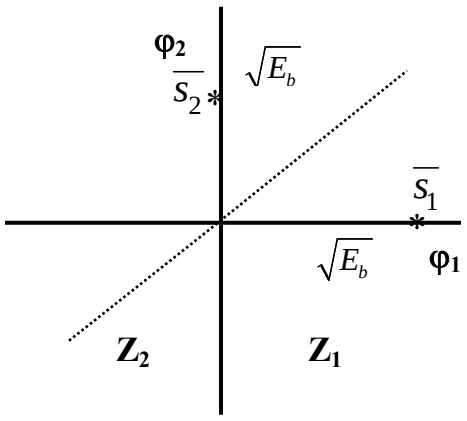


Fig. 5.28.1(a) *Signal constellation for BFSK showing the decision zones Z_1 and Z_2*

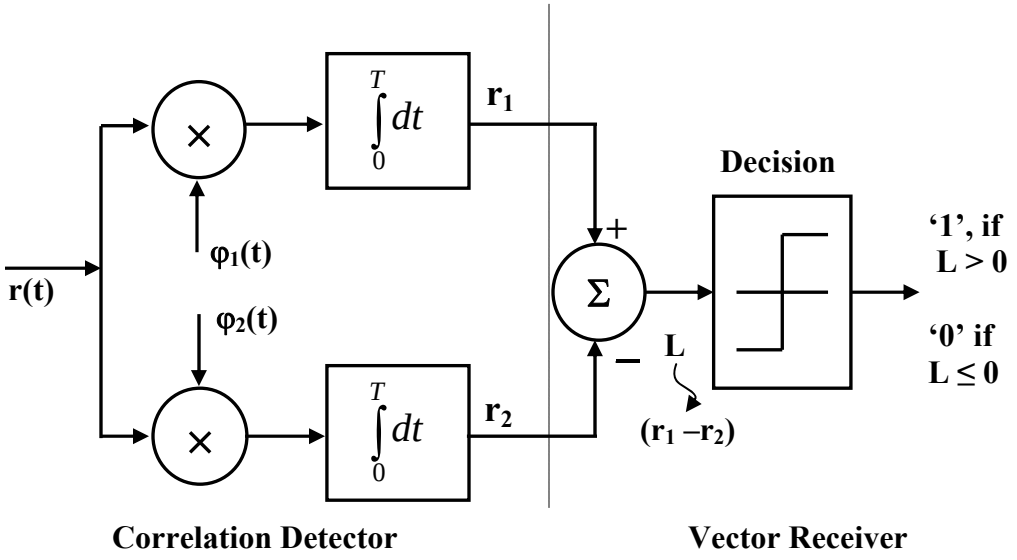


Fig. 5.28.1(b) *Coherent demodulator structure for BFSK highlighting the decision process*

Now, suppose $\bar{s}_1$ represents logic '1' and $\bar{s}_2$ represents logic '0'. If $s_1(t)$ is transmitted and if noise is absent, the output of the upper correlator in **Fig. 5.28.1(b)**, i.e. r_1 is $\sqrt{E_b}$ while the output of the lower correlator, i.e. r_2 is zero. So, we see that the intermediate parameter $L = (r_1 - r_2) > 0$. Similarly, it is easy to see that if $s_2(t)$ is transmitted, $L < 0$. Now, from **Fig. 5.28.1(a)** we see that the decision boundary is a straight line with unit slope. This implies that, if the received vector $\bar{r}$ at the output of the correlator bank is in decision zone Z_1, then $L > 0$ and otherwise it is in zone Z_2.

When we consider additive noise, 'L' represents a random variable whose mean is $+\sqrt{E_b}$ if message '1' is transmitted. For message '0', the mean of 'L' is $-\sqrt{E_b}$.
Further, as we noted in our general analysis earlier in Lesson #19 (Module #4), r_1 and r_2 are independent and identically distributed random variables with the same variance $\frac{N_o}{2}$.

So, variance of 'L' = variance of 'r_1' + variance of 'r_2' = $\frac{N_o}{2} + \frac{N_o}{2} = N_o$ 5.28.3

Now, assuming that a '0' has been transmitted, the likelihood function is:

$$f_L(l \mid 0) = \frac{1}{\sqrt{2\pi N_0}} . \exp\left[- \frac{\left\{ l - \left(-\sqrt{E_b} \right) \right\}^2}{2N_0} \right]$$

$$= \frac{1}{\sqrt{2\pi N_0}} . \exp\left[-\frac{\left(l + \sqrt{E_b}\right)^2}{2N_0} \right] \qquad 5.28.4$$

In the above expressions, '*l*' represents a sample value of the random variable 'L'.

From the above expression, we can determine the average probability of error when '0'-s are transmitted as:

$P_e(0)$ = Average probability of error when '0'– s are transmitted $= \int_0^\infty f_L(l \mid 0)\, dl$

$$= \frac{1}{\sqrt{2\pi N_0}} . \int_0^\infty \exp\left[-\frac{\left(l + \sqrt{E_b}\right)^2}{2N_0} \right] dl \qquad 5.28.5$$

Putting $\frac{l + \sqrt{E_b}}{\sqrt{2N_0}} = Z$ in the above expression, we readily get,

$$\mathrm{P}\,e(0) = \frac{1}{\sqrt{\pi}} \int_{\sqrt{\frac{E_b}{2N_0}}}^\infty \exp\left(-Z^2\right) dz$$

$$= \frac{1}{2} erfc\left(\sqrt{\frac{E_b}{2N_0}} \right) \qquad 5.28.6$$

Following similar approach, we get,

$P_e(1)$ = Average probability of error when '1'-s are transmitted

$$= \frac{1}{2} erfc\left(\sqrt{\frac{E_b}{2N_0}} \right) \qquad 5.28.7$$

Now, using the justification that '1' and '0' are equally likely to occur at the input of the modulator, the overall BER = $P_e = \frac{1}{2}.\ P_e(0) + \frac{1}{2}.\ P_e(1) = \frac{1}{2} erfc\left(\sqrt{\frac{E_b}{2N_0}} \right)$ 5.28.8

Fig. 5.28.2 shows the error performance of binary FSK for coherent demodulation. For comparison, the performance curve for BPSK is also included. Observe that the FSK modulation scheme performs significantly poorer.

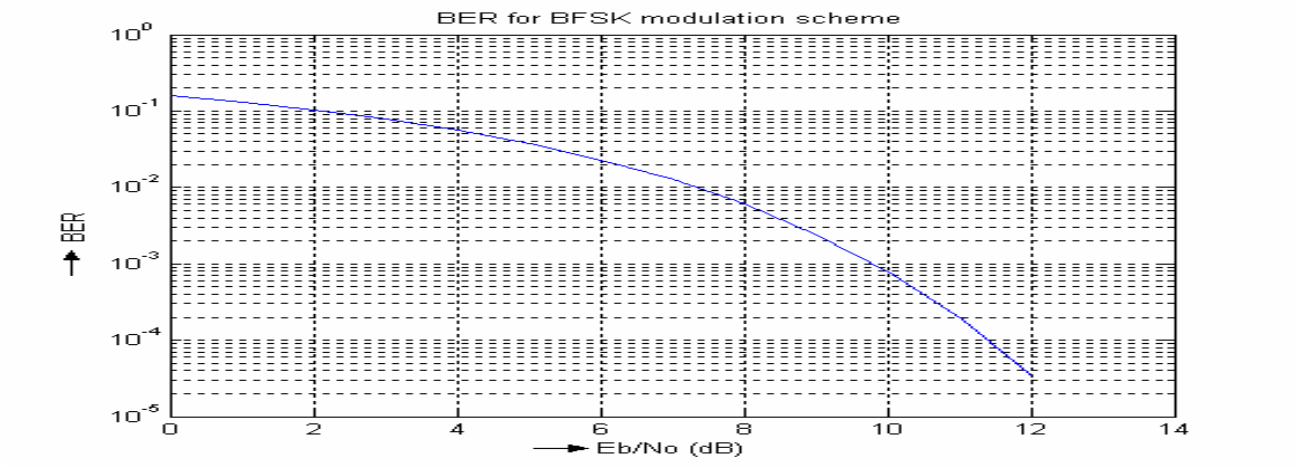


Fig. 5.28.2 *Error performance of binary FSK modulation schemes*

Performance indication for M-ary PSK

Fig. 5.28.3(a) shows the signal constellation for M-ary PSK with M = 2^3. As we have discussed earlier, an M-ary PSK modulated signal over a symbol duration 'T' can be expressed as:

$$s_i(t) = \sqrt{\frac{2E}{T}} \cos\left(w_c t + \frac{2\pi i}{M} \right), \text{ i = 0,1,….m M-1 and M = } 2^m \qquad 5.28.9$$

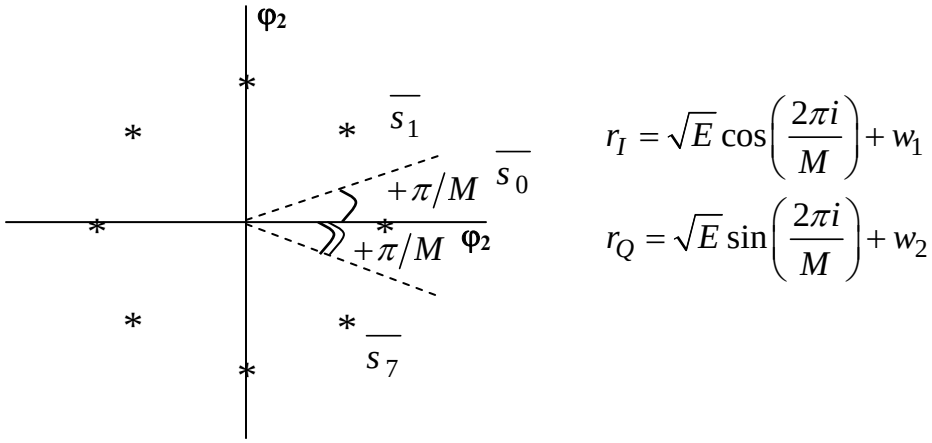


Fig. 5.28.3(a) *Signal constellation for M-ary PSK showing the decision zone Z_0*

The basis functions are :

$$\varphi_1(t) = \sqrt{\frac{2}{T}} \cos w_c t \text{ and } \varphi_2(t) = \sqrt{\frac{2}{T}} \sin w_c t\,; \qquad 0 \le t \le T \qquad 5.28.10$$

The decision zone Z_0 for the vector $\bar{s}_0$, making an angle of $\pm\pi^c/8 = \pm\pi^c/M$ at the origin, is also shown in the figure. The I-Q demodulator structure is reproduced in **Fig. 5.28.3(b)** for ease of reference. In case of PSK modulation, the Maximum Likelihood decision procedure can be equivalently framed in terms of a 'phase discrimination' procedure. This is usually followed in practice. The phase discriminator in **Fig. 5.28.3(b)** determines $\hat{\theta} = \tan^{-1}\left(\frac{r_Q}{r_I}\right)$ from r_I and r_Q as obtained from the correlation detector. The sign and magnitude of $\hat{\theta}$ produces the optimum estimate of a transmitted symbol.

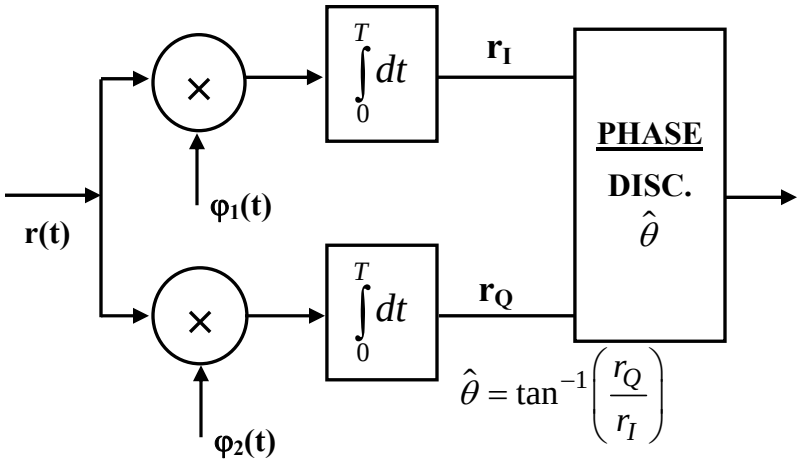


Fig. 5.28.3(b) *I-Q structure for coherent demodulation of M-ary PSK signal*

Determination of the expression for average Symbol Error Rate (SER) will be avoided here for simplicity. However, the approach is general and similar to what we have followed earlier. For example, consider the decision zone of $\overline{s_0}$ where $-\frac{\pi}{M} < \hat{\theta} < +\frac{\pi}{M}$. Now, if $f_\theta\left(\hat{\theta}\right)$ denotes the likelihood function for $\overline{s_0}$, the probability of correct decision for $\overline{s_0}$ ($P_{c_{s0}}$)is:

$$P_{c_{s0}} = \int_{-\pi/M}^{\pi/M} f_\theta\left(\hat{\theta}\right) d\hat{\theta} \qquad 5.28.11$$

So, the probability for erroneous decision when $\overline{s_0}$ is transmitted = $P_{e_{s0}} = 1 - P_{c_{s0}}$.

Fig. 5.28.4 shows the probability of symbol error of M-ary PSK modulation schemes for M = 8, 16 and 32. For comparison, the performance of QPSK modulation is also included. Note that the error performance degrades significantly with increase in the number of symbols (M). This is intuitively acceptable because, with increase in M, the decision region of a symbol narrows down and hence, smaller amount of noise may cause decision error.

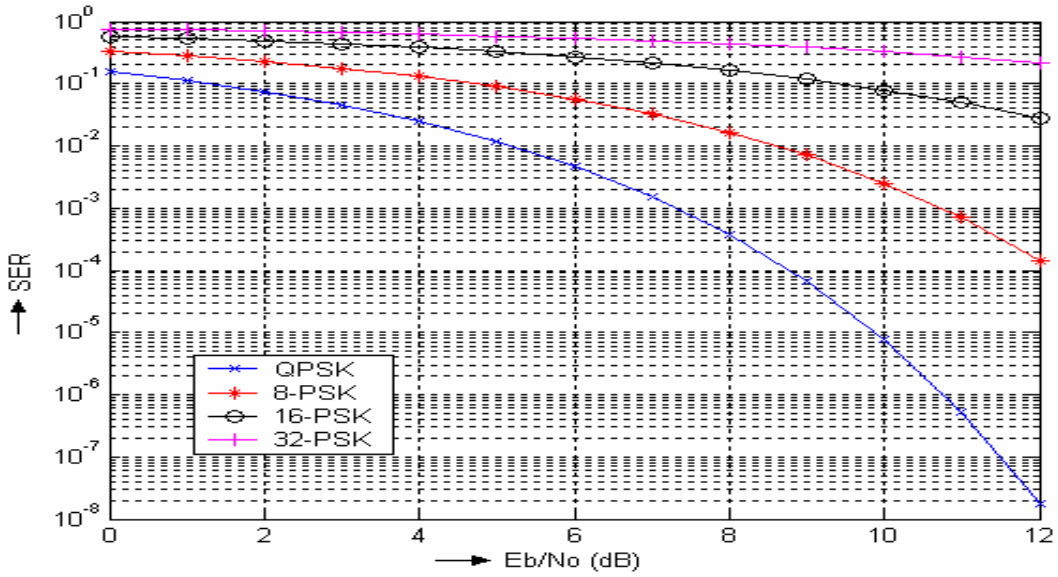


Fig. 5.28.4 *Optimum error performance of M-ary PSK demodulators for M= 4, 8, 16 and 32*

Another issue, which is of great significance in practice, is the need for precise recovery of carrier phase at the receiver so that the optimum coherent demodulation approach can be adopted. Usually, the required level of accuracy pushes up the complexity (and cost) of an M-ary PSK receiver for higher values of M. Still, M-ary modulation schemes have been popular for moderate values of 'M' such as 8 and16 because of their bandwidth efficiency. Another related family of modulation scheme, known as Quadrature Amplitude Modulation (QAM) has become attractive in terms of performance-bandwidth tradeoff especially for large number of signal points (M). We will briefly discuss about QAM in the next lesson.

Problems

Q5.28.1) Briefly mention how a binary FSK modulated signal can be demodulated non-coherently.

Q5.28.2) Compare Binary FSK & binary PSK modulation scheme.

Module 5

Carrier Modulation

Lesson 29

Minimum Shift Keying (MSK) Modulation

After reading this lesson, you will learn about

- ***CPFSK and MSK Modulation Schemes;***
- ***MSK Modulator;***
- ***Demodulation of MSK Signal;***
- ***Differential Detector;***

Linear modulation schemes without memory like QPSK, OQPSK, DPSK and FSK exhibit phase discontinuity in the modulated waveform. These phase transitions cause problems for band limited and power-efficient transmission especially in an interference limited environment. The sharp phase changes in the modulated signal cause relatively prominent side-lobe levels of the signal spectrum compared to the main lobe. In a cellular communication system, where frequency reuse is done extensively, these side-lobe levels should be as small as possible. Further in a power-limited environment, a non-linear power amplifier along with a band pass filter in the transmitter front-end results in phase distortion for the modulated signal waveform with sharp phase transitions. The abrupt phase transitions generate frequency components that have significant amplitudes. Thus the resultant power in the side-lobes causes co-channel and inter-channel interference.

Consequently, in a practical situation, it may be necessary to use either a linear power amplifier or a non-linear amplifier using extensive distortion compensation or selective pre-distortion to suppress out-of-band frequency radiation. However, high power amplifiers may have to be operated in the non-linear region in order to improve the transmission power. Continuous phase modulation schemes are preferred to counter these problems.

Continuous Phase Frequency Shift Keying (CPFSK) refers to a family of continuous phase modulation schemes that allow use of highly power-efficient non-linear power amplifiers. Minimum Shift Keying (MSK) modulation is a special subclass of CPFSK modulation and MSK modulation is free from many of the problems mentioned above. In this lesson, we briefly describe the various features of MSK modulation and demodulation through a general discussion of CPFSK modulation.

CPFSK and MSK Modulation Schemes

Extending the concepts of binary FSK modulation, as discussed earlier, one can define an M-ary FSK signal which may be generated by shifting the carrier by an amount $f_n = I_n /2\Delta f$. where $I_n = \pm 1, \pm 3, \ldots \pm (M-1)$. The switching from one frequency to another may be done by having $M = 2^k$ separate oscillators tuned to the desired frequencies and selecting one of the M frequencies according to the particular k bit symbol transmitted in a duration of $T = k/R$ seconds. But due to such abrupt switching the spectral side-lobes contain significant amount of power compared to the main lobe and hence this method requires a large frequency band for transmission of the signal.

This may be avoided if the message signal frequency modulates a single carrier continuously. The resultant FM signal is phase-continuous FSK and the phase of the carrier is constrained to be continuous. This class of continuous phase modulated signal is may be expressed as:

$$s(t) = \sqrt{(2E/T)\cos\left(2\pi f_c t + \Phi(t;I) + \Phi_0\right)} \tag{5.29.1}$$

where $\Phi(t;I)$ is the time-varying phase of the carrier defined as:

$$\Phi(t;I) = 2\pi T f_d \int_{-\infty}^{t} d(\tau)\, d\tau \tag{5.29.2}$$

f_d = peak frequency deviation ; Φ_0 = initial phase.

Let,
$$d(t) = \sum_{-\infty}^{\infty} I_n g(t - nT) \tag{5.29.3}$$

where $\{I_n\}$ is the sequence of amplitudes obtained by mapping k bit blocks of binary digits from the information sequence $\{a_n\}$ into amplitude levels ±1, ±3, …± (M-1). g(t) is a rectangular pulse of amplitude 1/2T and duration T. The signal d(t) is used to frequency modulate the carrier.

Substituting equation (5.29.3) in (5.29.2),

$$\Phi(t;I) = 2\pi T f_d \int_{-\infty}^{\infty} \left[\sum_{-\infty}^{\infty} I_n g(\tau - nT) \right] d\tau \tag{5.29.4}$$

It is evident from (5.29.4) that though d(t) contains discontinuities, the integral of d(t) is continuous. The phase of the carrier in the interval $nT \le t \le (n+1)T$ is determined by integrating (5.29.4):

$$\begin{aligned} \Phi(t;I) &= 2\pi f_d T \sum_{-\infty}^{\infty} I_k + 2\pi f_d (t - nT) I_n \\ &= \theta_n + 2\pi h I_n q(t - nT) \end{aligned} \tag{5.29.5}$$

where h, θ and q(t) are defined as

$$h = 2f_d T \tag{5.29.6}$$

$$\theta_n = \pi h \sum_{-\infty}^{\infty} I_k \tag{5.29.7}$$

and

$$\begin{aligned} q(t) &= 0 && t < 0 \\ &= 1/2T && (0 \le t \le T) \\ &= 1/2 && (t > T) \end{aligned} \tag{5.29.8}$$

As is evident from (5.29.7), θ_n represents the accumulated phase due to all previous symbols up to time (n – 1) T.

Minimum Shift Keying (MSK) is a special from of binary CPFSK where the modulation index h = 1/2. Substituting this value of h in (5.29.5) we get the phase of the carrier in the interval nT ≤ t ≤ (n + 1) T as

$$\Phi(t;I) = 1/2\pi \sum_{-\infty}^{\infty} I_k + \pi I_n q(t - nT)$$

$$= \theta_n + 1/2\pi I_n (t - nT) \; ; \quad nT \le t \le (n+1)T \qquad (5.29.9)$$

The above equation is obtained by substituting the value of q(t) by

$$q(t) = \int_{-\infty}^{t} g(\tau) d\tau$$

where g(τ) is some arbitrary pulse. Thus the carrier-modulated signal can be represented as

$$s(t) = A\cos\left[2\pi f_c t + \theta_n + 1/2\pi I_n (t - nT)/T\right]$$

$$= A\cos\left[2\pi (f_c + 1/4TI_n)t - 1/2n\pi I_n + \theta_n\right] \; ; \; nT \le t \le (n+1)T \qquad (5.29.10)$$

From the above expression it may be noted that binary CPFSK signal has one of the two possible frequencies in the interval nT ≤ t ≤ (n + 1)T as:

$$f_1 = f_c - \tfrac{1}{4}\,T \quad \text{and} \quad f_2 = f_c + \tfrac{1}{4}\,T \qquad (5.29.11)$$

Equation (5.29.10) can also be written as

$$s_i(t) = A\cos\left[2\pi f_i t + \theta_n \_ 1/2\, n\pi (-1)^{(i-1)}\right] \quad , i = 1, 2 \qquad (5.29.12)$$

The frequency separation $\Delta f = f_2 - f_1 = \frac{1}{2}T$ is the minimum necessary to ensure the orthogonality of the signals $s_1(t)$ and $s_2(t)$ over a signaling interval of length T. Hence binary CPFSK with h = ½ is called Minimum Shift Keying (MSK) modulation.

MSK Modulator

The block schematic diagram of an MSK modulator is shown in **Fig. 5.29.1**. This structure is based on Equation (5.29.1). However, this cannot be easily converted to hardware as an exact relation between the symbol rate and modulation index is required necessitating intricate control circuitry. An advantage with this structure is that it can be used for both analog and digital input signals.

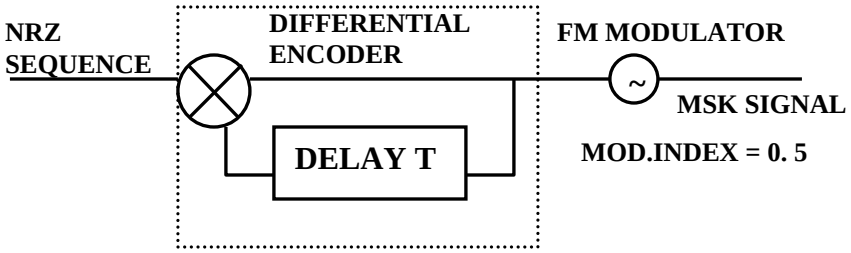


Fig. 5.29.1 An MSK modulator

I denote the sequence of amplitude obtained by mapping binary digits from the information sequence $\{a_n\}$ into amplitude level ±1. This is multiplied with 2π and the resultant product is then used to frequency modulate the carrier.

Demodulation of MSK Signal

An MSK modulated signal can be demodulated either by a coherent or a differential (non coherent) demodulation technique. The coherent demodulator scheme, as we have mentioned earlier for QPSK and other modulations, needs precise reference of carrier, phase, and frequency for optimum demodulation. This is not that easy for an MSK modulated signal. Hence a sub optimal differential modulated technique along with threshold detection is a popular choice in many applications such as in cellular telephony. In the following we briefly discuss a one bit differential demodulation scheme for MSK

Differential Detector

The block diagram for differential detection of MSK signal is shown in **Fig. 5.29.2**.

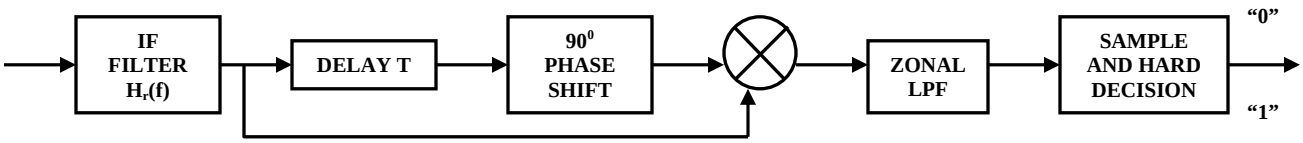


***Fig. 5.29.2** Demodulator for MSK*

A 90^0 phase shifter circuit is included in the delayed arm so that to represent the multiplier output represents the *sine* of the change in the phase of the received signal over one-symbol duration.

Now the received signal r(t) may be expressed as:

$$r(t) = \sqrt{(2P)}A(t)\cos\left[\omega_0 t + \phi(t) + n(t)\right] \qquad (5.29.13)$$

where $\sqrt{2PA(t)}$ represents envelope, $\phi(t)$ is the phase and n(t) is the noise.

The noise n(t) can be expressed using quadrature representation of noise by

$$n(t) = n_c(t)\cos\left[\omega_0 t + \phi(t)\right] - n_s(t)\sin\left[\omega_0 t + \phi(t)\right] \qquad (5.29.14)$$

Replacing n(t) of Eqn. (5.29.13) with that of Eqn. (5.29.15) we get

$$r(t) = R(t)\cos\left[\omega_0 t + \phi(t)\right] + \eta(t) \qquad (5.29.15)$$

where

$$R(t) = \sqrt{\left(\sqrt{2P}A(t) + n_c(t)\right)^2 + n_s^2(t)} \qquad (5.29.16)$$

and

$$\eta(t) = -\tan^{-1}\frac{n_s(t)}{\sqrt{2P}A(t) + n_c(t)} \qquad (5.29.17)$$

The one-bit differential detector compares the phase of the received signal r(t) with its one-bit delayed and 90^0 phase shifted version r(t – T). The resultant output of the comparator is given by

$$y(t) = 1/2\left[R(t)R(t-T)\right]\sin\left[\omega_0 T + \Delta\phi(T)\right] \qquad (5.29.18)$$

where the phase difference $\Delta\phi(t)$ is,

$$\Delta\phi(T) = \phi(t) - \phi(t-T) + \eta(t) - \eta(t-T) \qquad (5.29.19)$$

This phase difference indicates the change over symbol duration of the distorted signal phase and phase noise due to the AWGN.

Now let,

$$\omega_0 T = 2\pi K, \text{ where K an integer} \qquad (5.29.20)$$

Then eqn. 5.29.18 reduces to

$$y(t) = 1/2\left[R(t) * R(t-T)\right]\sin\left(\Delta\phi(T)\right) \qquad (5.29.21)$$

The receiver then decides that a '+1' has been sent if y(t) > 0 and a '-1' otherwise. As the envelope R(t) is always positive, actually it is sufficient to determine whether $\sin\left(\Delta\phi(T)\right)$ is ≥ 0.

Problems

Q5.29.1) Justify how MSK can also be viewed as a kind of FSK modulation scheme.

Q5.29.2) Why differential demodulation of MSK is popular?

Module 5

Carrier Modulation

Lesson 30

Orthogonal Frequency Division Multiplexing (OFDM)

After reading this lesson, you will learn about:

- ***Basic concept of OFDM;***
- ***Peak to Average Power Ratio (PAPR);***
- ***Effect of the transmission channel on OFDM;***
- ***Effect of ISI on OFDM;***
- ***OFDM applications;***

Orthogonal Frequency Division Multiplexing (OFDM)

OFDM is a relatively new spectrally efficient digital modulation scheme which employs multiple carriers that are mutually orthogonal to one another over a given time interval. Each carrier, consisting of a pair of sine wave and a cosine wave, is referred as a 'sub-carrier'.

Let us consider an OFDM scheme with N sub-carriers. The available transmission bandwidth is equally divided amongst the N sub-carriers. If 'W' Hz is the single-sided transmission bandwidth available, the bandwidth allocated to each sub-carrier is $\frac{W}{N}$ Hz. The difference between two adjacent sub carriers is called the sub carrier spacing, which is also $\frac{W}{N}$ Hz in our example. Each sub carrier, upon data modulation may often be categorized as a narrowband modulated signal but the overall OFDM signal is a wideband signal for moderate or large value of 'N'. Usually, the modulation operation is carried out at the baseband level and the baseband-modulated signal is translated in the frequency domain by frequency up-conversion to the required radio frequency band.

A data symbol consists of several bits and the symbol is used to modulate all the carriers simultaneously. The symbol rate and the sub carrier spacing are so chosen that all the carriers are orthogonal over one OFDM symbol duration. **Fig. 5.30.1** shows three sub carriers, which are orthogonal over one symbol duration. If the duration of an OFDM symbol is 'T' second, we see from the figure that the sub carrier frequencies are $f_0 = \frac{1}{T}$ Hz, $f_1 = \frac{2}{T}$ Hz and $f_2 = \frac{3}{T}$ Hz. It is interesting to note that, many other combinations of three frequencies are possible which are orthogonal over T second, such as $(\frac{1}{T}, \frac{2}{T}, \frac{4}{T})$, $(\frac{1}{T}, \frac{3}{T}, \frac{4}{T})$, $(\frac{1}{T}, \frac{5}{T}, \frac{9}{T})$ and so on. However, all combinations are not optimally bandwidth efficient.

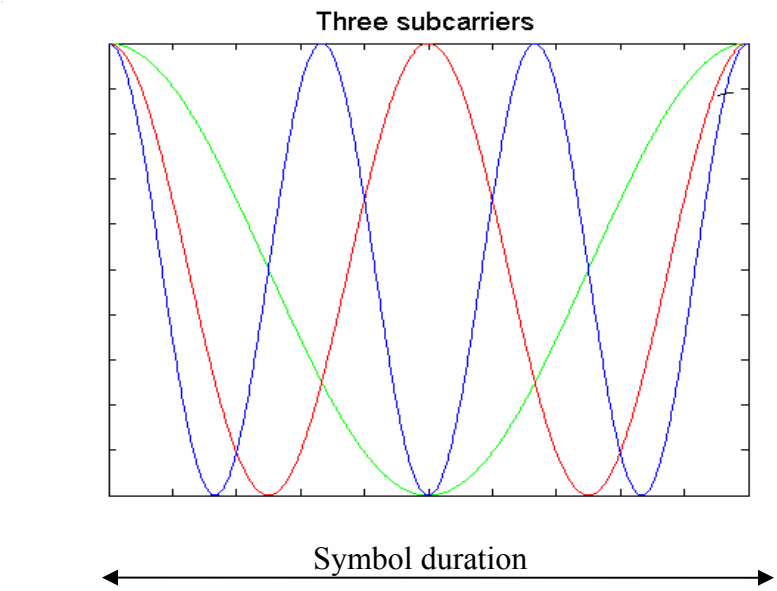


Fig. 5.30.1 *Three sub carriers, which are orthogonal over one symbol duration*

A particularly compact arrangement may be like this: We accept a group of 2N bits every T second and form N dibits (one dibit is made of two bits) from these bits. All these N dibits are then fed in parallel to the N sub carriers with the first dibit modulating the first sub carrier, second dibit modulating the second sub carrier and so forth. As noted earlier, each sub carrier consists of a pair of orthogonal sinusoids at the same frequency. So, each sub carrier receives one dibit every T second and hence its own symbol rate of arrival is 1 symbol per T second. We set the sub carrier frequencies as

$$f_0 = \frac{1}{T} \text{ Hz}, \; f_1 = \frac{2}{T} \text{Hz}, \; \ldots\ldots, \; f_{N-1} = \frac{N}{T} \text{ Hz}.$$

Fig.5.30.2 shows a conceptual diagram highlighting the orthogonal multiple carrier modulation scheme. The a_i-s in the diagram indicates the modulating signal in the I-path and the b_i-s are the modulating signals in the Q-path. The 'encoder' in a practical system performs several operations but is of no special significance at the moment. Let us refer a 'cosine' carrier as an in-phase carrier and a 'sine' carrier as a quadrature-phase carrier. All the in-phase carrier modulated signals are added algebraically and similarly are the quadrature-phase modulated signals. The overall I-phase and Q-phase signals together form the complex baseband OFDM signal. At this point, one may interpret the scheme of **Fig.5.30.2** as consisting of a bank of N parallel QPSK modulators driven by N orthogonal sub carriers.

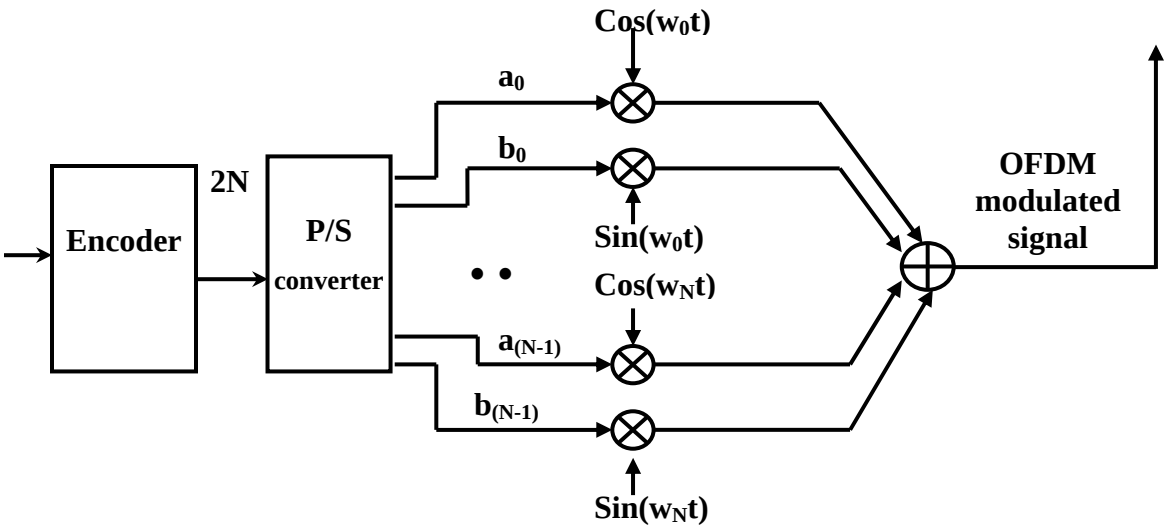


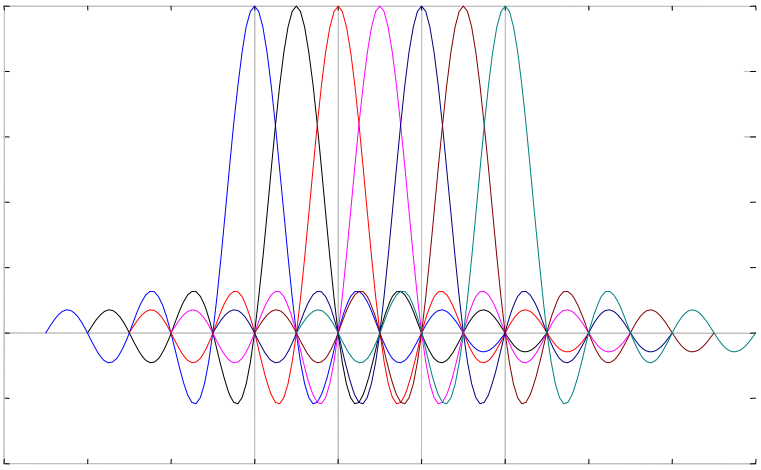
Fig.5.30.2 *A conceptual diagram highlighting orthogonal multiple carrier modulation ($\omega_o = 2\pi f_o = 2\pi/T$)*

Mathematically, if $\{\widetilde{X}(k)\}$ represents a sequence of N complex modulating symbols, the complex baseband modulated OFDM symbol is represented as

$$\widetilde{x}(t) = \frac{1}{N}.\sum_{k=0}^{N-1} \widetilde{X}(k) e^{j2\pi k f_0 t}, \quad 0 \le t \le T \text{ and k = 0, 1, …, (N-1)} \qquad 5.30.1$$

$\widetilde{X}(k)$ is the k-th symbol modulating the k-th sub carrier and f_o is the inter-sub carrier spacing.

An interesting and practically useful feature of the OFDM modulation scheme is that pulse shaping is not necessary for the modulating signals because a bunch of orthogonal carriers, when modulated by random pulse sequences, have a very orderly spectrum as sketched in **Fig. 5.30.3**. As indicated, the orthogonal sub-carriers occupy the spectral zero crossing positions of other sub-carriers. This ensures that a sub carrier modulated signal with seemingly infinite spectrum does not interfere with the signals modulated by other sub carriers.



→ Frequency

Figure 5.30.3: *OFDM and the orthogonality principle. The sub-carriers occupy the spectral zero crossing positions of other sub-carriers. Sub-carriers are orthogonal*

Further, it also implies that the OFDM modulated signal of **Eq. 5.30.1** can be generated by simple Inverse Discrete Fourier Transform (IDFT) which can be implemented efficiently as an N-point Inverse Fast Fourier Transform (IFFT). If the modulated signal in time domain (**Eq. 5.30.1**) is uniformly sampled with an interval $\frac{T}{N}$, the n-th sample of the OFDM signal is:

$$\tilde{x}\left(n\frac{T}{N}\right) \equiv \tilde{x}(n) = \frac{1}{N}\sum_{k=0}^{N-1}\tilde{X}(k)e^{j2\pi kf_0nT/N}, \quad 0 \le n \le (N-1) \qquad 5.30.2$$

Similarly, at the receiving end, an N-point FFT operation does the equivalent job of demodulation of OFDM signal. This makes digital design and implementation of OFDM modulator and demodulator very convenient. However, there are several practical issues which demand proper attention.

Fig.5.30.4 shows a magnitude plot of complex time domain samples $\tilde{x}\left(n\frac{T}{N}\right)$ for 10 OFDM symbols. Sixty four sub-carriers have been used to obtain the OFDM signal. Note that the baseband modulated signal looks random with great variation in amplitude. This amplitude fluctuation is expressed by a parameter called '*Peak to Average Power Ratio (PAPR)*'. Higher is this ratio, more fluctuating is the envelope of the modulated signal.

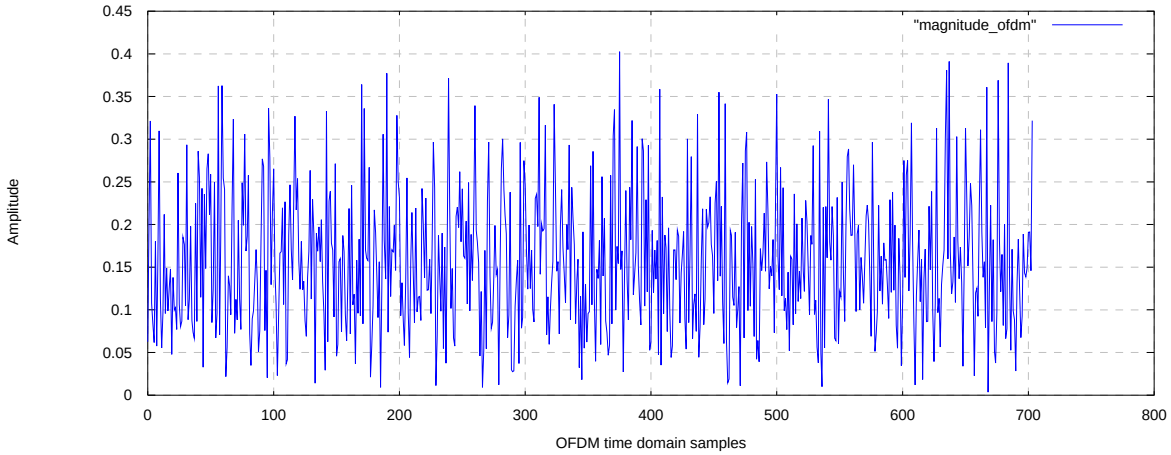


Fig.5.30.4 *Magnitude of complex time domain samples* $\tilde{x}\left(n\frac{T}{N}\right)$ *for ten OFDM symbols*

For a constant envelope modulation scheme, the PAPR is 1.0. An OFDM signal with high PAPR needs a power amplifier with a large linear range of operation in the transmitter. Otherwise, the OFDM signal gets distorted and produces harmonic distortions while being amplified by the power amplifier and the orthogonality among the sub carriers is affected in the process. This results in considerable Inter Symbol Interference (ISI) at the receiver and the quality of demodulated data degrades. Fortunately, several schemes such as clipping, filtering and special forms of coding are available to contain the PAPR of an OFDM signal before it is actually transmitted.

Another important issue is the effect of the transmission channel. Due to its high spectral efficiency (**Fig. 5.30.5** shows a representative amplitude spectrum for OFDM signal), OFDM is favoured for data transfer at high rates (typically beyond 1Mbps) in wireless networks. The transmission channels in practical environments often manifest multiple signal paths and signal fading at a receiver. These channel-induced phenomena can potentially degrade the quality of the received signal by a) affecting the sub carrier orthogonality and b) introducing ambiguities in carrier phase and symbol timing. As a result, the quality of received data degrades. For example, **Fig.5.30.6** shows the error performance of a simulated OFDM scheme using 64 sub carriers in presence of thermal noise only.

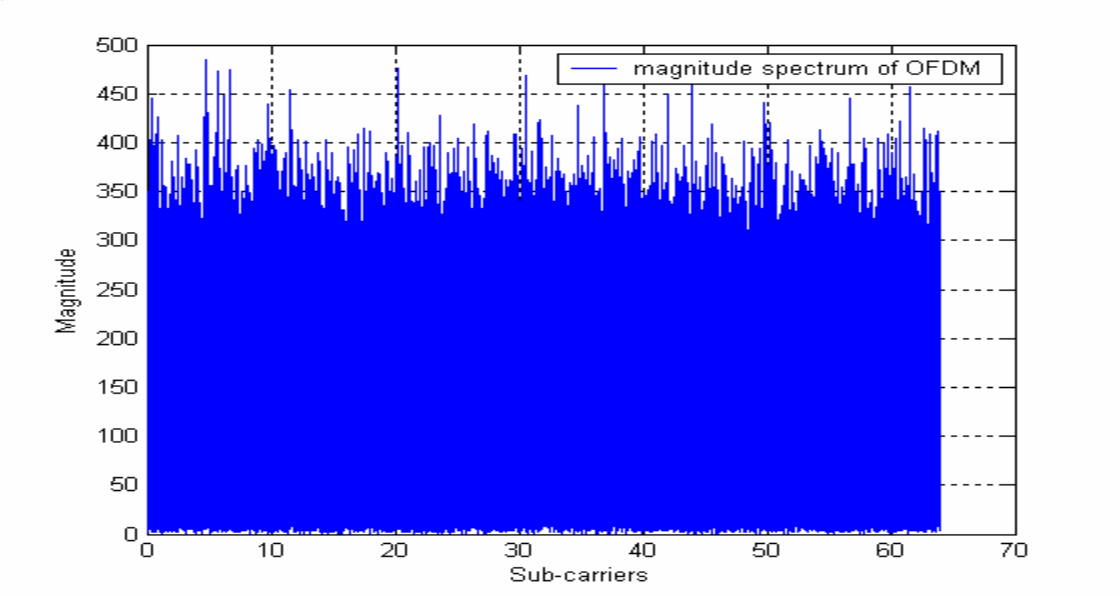


Fig.5.30.5 *Magnitude spectrum of OFDM signal with 64 subcarriers. The normalized subcarriers are spaced at 1 Hz apart. The subcarrier values are shown along the horizontal axis. The random fluctuation at the top of the spectrum is largely due to the finite number of OFDM symbols that was considered in simulation*

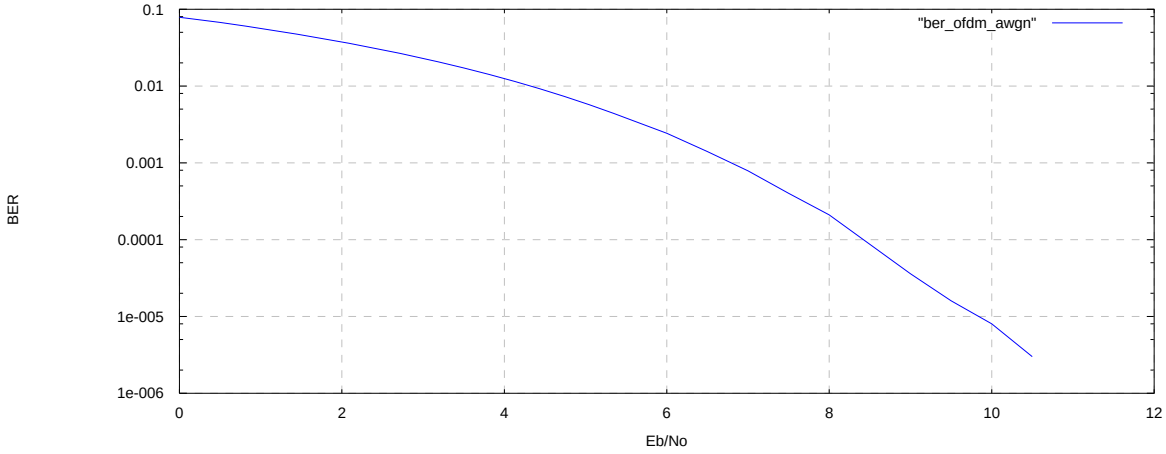


Fig.5.30.6 *BER vs Eb/No (in dB) for a 64 carrier OFDM system in AWGN channel*

One may note that the OFDM modulation scheme performs very close to the BER performance of QPSK modulation scheme. However, lack of orthogonality and lack of synchronization can make the situation very poor. **Fig.5.30.6** shows that less than 10 dB in E_b/N_o is sufficient to achieve an average error rate of 10^{-5} in presence of additive Gaussian noise and with perfect synchronization at the receiver; but in an indoor wireless

local area network environment, it may need more than 20 dB of E_b/N_o to ensure same error rate.

In a practical system, the problems of phase and timing synchronization are addressed by inserting appropriate training sequence in a frame of symbols. The receiver knows the training sequence and hence it can estimate the quality of transmission channel from a sizeable number of samples of the distorted training sequence, sent as preamble to the information-carrying symbols. The frequency or phase synchronization process is vital for restoring the orthogonality among sub carriers before data demodulation takes place in the receiver. Often this is achieved by following a two-step scheme: a coarse synchronization followed by a fine synchronization process.

Inter symbol interference (ISI) may manifest in an OFDM demodulator either a) due to channel multipath or b) due to lack of orthogonality or c) combination of both. As a result, delayed portion of the previous symbol superimposes on the initial portion of the present symbol at the receiver. Thus initial portion of the present symbol gets affected more due to multi-path fading. To reduce this effect, usually the duration of an OFDM symbol (expressed by N samples) is stretched a little in time and a few samples of the latter part of an OFDM symbol are appended at the beginning of the symbol. These samples are called 'cyclic prefix'. If N_g is the number of time domain samples of OFDM symbol used as cyclic prefix, the total number of samples in a cyclic-prefixed OFDM symbol becomes $N_s = N + N_g$.

Applications of OFDM

Some applications of OFDM in modern wireless digital transmission systems are mentioned below:

a) Asynchronous Digital Subscriber Line (ADSL), High speed DSL, Very high speed DSL use OFDM for transmission of high rate data transfer
b) Digital Audio Broadcasting (DAB) and Digital Video Broadcasting (DVB).
c) IEEE 802.11a, IEEE 802.11g and HYPERLAN2 wireless Local Area Network (WLAN) standards include OFDM for supporting higher bit rates
d) IEEE 802.16 Wireless Metropolitan Area Network (MAN) standard also includes OFDM.

Problems

Q5.30.1) Sketch two waveforms other than sinusoid which are orthogonal to each other over some time interval 'T'.

Q5.30.2) Mention two disadvantages of signaling using OFDM.

Q5.30.3) Explain how and where the concept of FFT is useful in an OFDM transceiver.

Module 5

Carrier Modulation

Lesson 31

Carrier Synchronization

After reading this lesson, you will learn about:

- ***Bit Error Rate (BER) calculation for BPSK;***
- ***Error Performance of coherent QPSK;***
- ***Approx BER for QPSK;***
- ***Performance Requirements;***

There are two major approaches in carrier synchronization:

a. To multiplex (FDM) a special signal, called pilot, which allows the receiver to extract the phase of the pilot and then use the information to synchronize its local carrier oscillator (L.O.) to the carrier frequency and phase of the received signal.

b. To employ a phase locked loop (PLL) or other strategy to acquire and track the carrier component. This approach has some advantages such as:
   a. No additional power is necessary at the transmitter for transmitting a pilot,

   b. It also saves overhead in the form of bandwidth or time. However, the complexity of the receiver increases on the whole. In this lesson, we will briefly discuss about a few basic approaches of this category.

A Basic Issue

Consider an amplitude-modulated signal with suppressed carrier

$$s(t) = A(t)\cos(\omega_c t + \bar{\varphi}) = A(t)\cos(2\pi f_c t + \bar{\varphi}) \qquad 5.31.1$$

Let the reference carrier generated in the demodulator be:

$$c(t) = \cos(2\pi f_c t + \phi) \qquad 5.31.2$$

Then, on carrier multiplication, we get,

$$s(t)c(t) = \frac{1}{2}.A(t).\cos(\phi - \bar{\phi}) + \frac{1}{2}.A(t).\cos(2\varpi_c t + \phi + \bar{\phi}) \qquad 5.31.3$$

The higher frequency term at twice the carrier frequency is removed by a lowpass filter in the demodulator and thus, we get the information-bearing signal as:

$$y(t) = \frac{1}{2}A(t)\cos(\varphi - \bar{\varphi}) \qquad 5.31.4$$

We see that the signal level is reduced by a factor $\cos(\varphi - \bar{\varphi})$ and its power is reduced by $\cos^2(\varphi - \bar{\varphi}) \rightarrow \varphi - \bar{\varphi} = 30^0$ means 1.25 dB decrease in signal power. So, if this phase offset is not removed carefully, the performance of the receiver is going to be poorer than expected. For simplicity, we did not consider any noise term in the above example and assumed that the phase offset ($\phi - \bar{\phi}$) is time-independent. Both these assumptions are not valid in practice.

Let us now consider QAM and M-ary PSK-type narrowband modulation schemes and let the modulated signal be:

$$s(t) = A(t)\cos(\omega_c t + \varphi) - B(t)\sin(\omega_c t + \varphi) \qquad 5.31.5$$

The signal is demodulated by the two quadrature carriers:

$$C_c(t) = (\cos\omega_c t + \bar{\varphi}) \text{ and } C_s(t) = -\sin(\omega_c t + \bar{\varphi}) \qquad 5.31.6$$

Now,

$$s(t)C_c(t) \xrightarrow[LPF]{after} y_I(t) = \frac{1}{2}A(t)\cos(\varphi - \bar{\varphi}) - \frac{1}{2}B(t)\sin(\varphi - \bar{\varphi}) \qquad 5.31.7$$

Similarly,

$$s(t)C_s(t) \xrightarrow[LPF]{after} y_Q(t) = \frac{1}{2}B(t)\cos(\varphi - \bar{\varphi}) + \frac{1}{2}A(t)\sin(\varphi - \bar{\varphi}) \qquad 5.31.8$$

'$y_I(t)$' and '$y_Q(t)$' are the outputs of the in-phase and Q-phase correlators. Note that the cross-talk interference from A(t) and B(t) in $y_I(t)$ and $y_Q(t)$. As the average power levels of A(t) and B(t) are similar, a small phase error ($\phi - \bar{\phi}$) results in considerable performance degradation. Hence, phase accuracy in a QAM or M-ary PSK receiver is very important.

Squaring Loop

Let us consider a modulated signal of the type:

$$s(t) = A(t)\cos(2\pi f_c t + \varphi)$$

The signal is dependent on the modulating signal A(t) and its mean, i.e. E[s(t)] = 0 when the signal levels are symmetric about zero. However, $s^2(t)$ contains a frequency component at $2f_c$. So, $s^2(t)$ can be used to drive a phase locked loop (PLL) tuned to $2f_c$ and the output of the voltage controlled oscillator (VCO) can be divided appropriately to get 'f_c' and 'φ' (refer **Fig.5.31.1**).

$$s^2(t) = A^2(t)\cos^2(2\pi f_c t + \varphi) = \frac{1}{2}A^2(t) + \frac{1}{2}A^2(t)\cos(4\pi f_c t + 2\varphi) \qquad 5.31.9$$

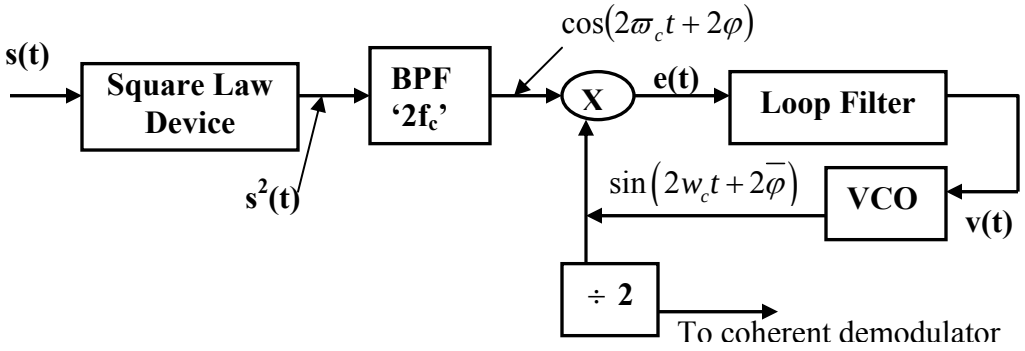


Fig. 5.31.1 *Block diagram illustrating the concept of a squaring loop*

Now, the expected value of $s^2(t)$ is:

$$E[s^2(t)] = \frac{1}{2}E[A^2(t)] + \frac{1}{2}E[A^2(t)]\cos(4\pi f_c t + 2\varphi) \qquad 5.31.10$$

Therefore, mean value of the output of the bandpass filter (BPF) at $2f_c$ is a sinusoid of frequency $2f_c$ and phase 2φ. So, the transmitted frequency can be recovered at the receiver from the modulated signal. An important point here is that the squaring operation has removed the sign information of the modulating signal A(t). As you may identify, the multiplier, along with the VCO and the loop filter constitute a PLL structure.

Principle of PLL

We will now briefly discuss about the principle of PLL. The peak amplitude of the input signal to the PLL is normalized to unit value. In practice, an amplitude limiter may be used to ensure this.

Let, $\bar{\varphi}$ represent the phase estimate at the output of VCO so that the 'error signal' at the output of the multiplier can be written as:

$$e(t) = \cos(4\pi f_c t + 2\varphi)\sin(4\pi f_c t + 2\bar{\varphi})$$

$$= \frac{1}{2}\sin 2(\bar{\varphi} - \varphi) + \frac{1}{2}\sin(8\pi f_c t + 2\varphi + 2\bar{\varphi}) \qquad 5.31.11$$

The loop filter is an LPF which responds to $\frac{1}{2}\sin 2(\bar{\varphi} - \varphi)$.

For a first order PLL (refer **Fig 5.31.2** for the block diagram of a first order PLL), the transfer function of the filter is of the type

$$G(s) = \frac{1 + \tau_2 s}{1 + \tau_1 s} \; ; \qquad \tau_1 >> \tau_2 \qquad 5.31.12$$

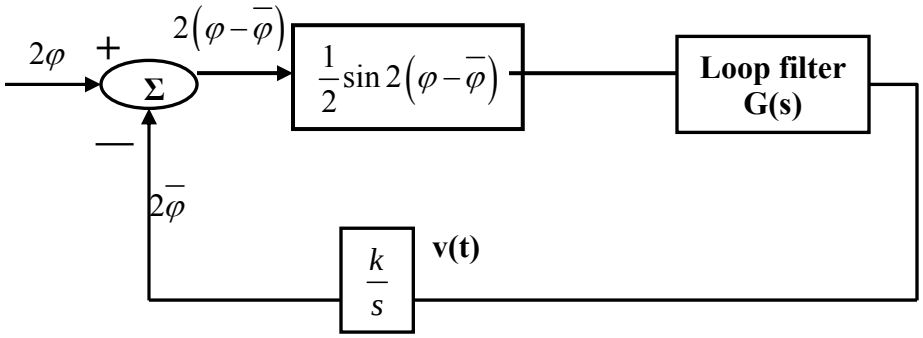


Fig. 5.31.2 *Equivalent block diagram of a first order PLL*

Now, the output of LPF, v(t) drives the VCO whose instant phase is given by,

$$4\pi f_c t + 2\bar{\varphi}(t) = 4\pi f_c t + k\int_{-\alpha}^{t} v(\tau)d\tau \qquad 5.31.13$$

$$\therefore 2\bar{\varphi} = k\int_{-\alpha}^{t} v(\tau)d\tau \qquad 5.31.14$$

The closed loop transfer function of this first order PLL is:

$$H(s) = \frac{kG(s)/s}{1+kG(s)/s} \qquad 5.31.15$$

Now, for small$(\varphi-\bar{\varphi})$, we may write $\frac{1}{2}\sin 2(\varphi-\bar{\varphi}) \cong (\varphi-\bar{\varphi})$

With this approximation, the PLL is linear and has a closed loop transfer function:

$$H(s) = \frac{k\ G(s)/s}{1+k\ G(s)/s} = \frac{1+\tau_2 s}{1+\left(\tau_2+\frac{1}{k}\right)s+\frac{\tau_1}{k}s^2} \qquad 5.31.16$$

The parameter 'τ_2' controls the position of 'zero' while 'k' and 'τ_1' together control the position of poles of the closed loop system.
It is customary to express the denominator of H(s) as:

$$D(s) = 1+\left(\tau_2+\frac{1}{k}\right)s+\frac{\tau_1}{k}s^2 = s^2+2\sigma\omega_n s+\omega_n{}^2 \qquad 5.31.17$$

Where, σ: Loop damping factor and $\left[=\frac{\tau_2+1/k}{2\omega n}\right]$;

ω_n : Natural frequency of the loop. $\left[\omega_n = \sqrt{k/\tau_1}\right]$

Now, the closed loop transfer function can be expressed as,

$$H(s) = \frac{\left(2\sigma\omega_n - \omega_n^2/k\right)s + \omega_n^2}{s^2 + 2\sigma\omega_n s + \omega_n^2} \qquad 5.31.18$$

The one-sided noise equivalent BW of the loop is

$$B_{eq} = \frac{\tau_2^2\left(\frac{1}{\tau_2^2} + \frac{k}{\tau_1}\right)}{4\left(\tau_2 + \frac{1}{k}\right)} = \frac{1 + (\tau_2\omega_n)^2}{8\sigma\omega_n} \qquad 5.31.19$$

Effect of noise on phase estimation

Now, we assume that the phase of the carrier varies randomly but slowly with time and that the received signal is also corrupted by Gaussian noise.

Let, us express the narrowband signal and noise separately as:

$s(t) = A_c \cos[w_c t + \varphi(t)]$ and

$n(t) = x(t) \cos w_c t - y(t) \sin w_c t$, with power spectral density of $N_0 / 2$ (w/Hz)

$= n_c(t) \cos[w_c t + \varphi(t)] - n_s(t) \sin[w_c t + \varphi(t)]$

where

$n_c(t) = x(t) \cos \varphi(t) + y(t) \sin \varphi(t)$

$n_s(t) = - x(t) \cos \varphi(t) + y(t) \cos \varphi(t)$

Note that,

$n_c(t) + j\, n_s(t) = [\, x(t) + j\, y(t)]\, e^{-j\,\varphi(t)}$

$\therefore$ $n_c(t)$ and $n_s(t)$ have same stat. as $x(t)$ and $y(t)$.

So, the input to loop filter e(t) can be expressed as [see **Fig.5.31.3(a)**],

$e(t) = A_c \sin \Delta\varphi(t) + n_1(t)$

$= A_c \sin \Delta\varphi(t) + n_c(t) \sin \Delta\varphi(t) - n_s(t) \cos \Delta\varphi(t)$ 5.31.20

Note that the model in **Fig.5.31.3(a)** is not linear. **Fig. 5.31.3(b)** shows the linearized but approximate model. In the model, $n_2(t)$ is a scaled version of $n_1(t)$.

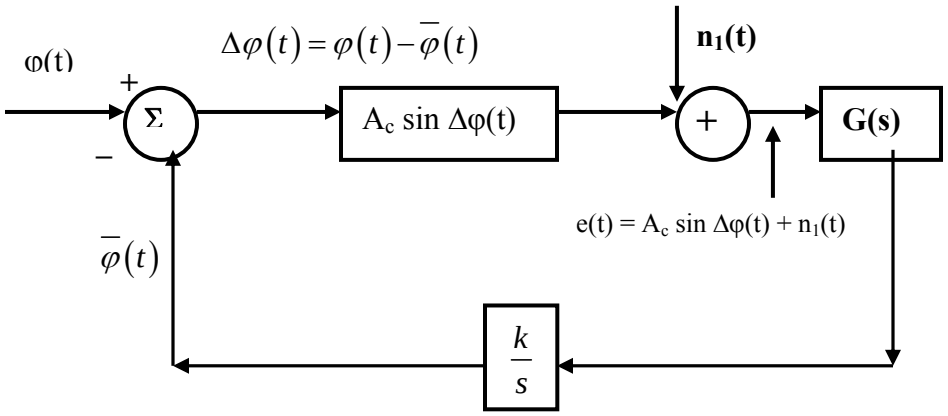


Fig.5.31.3(a) *The equivalent PLL model in presence of additive noise*

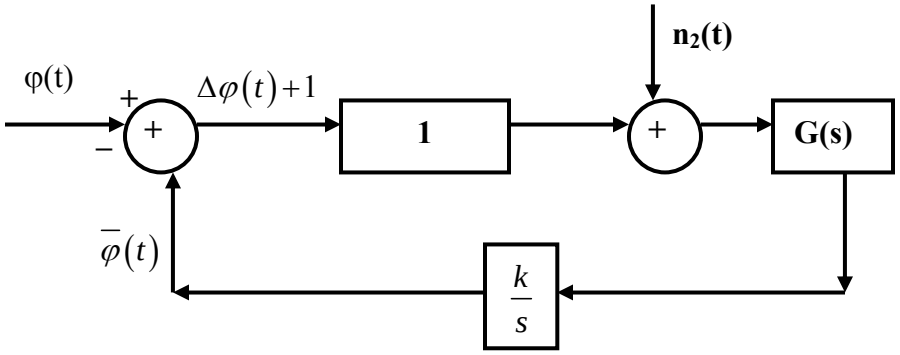


Fig. 5.31.3(b) *A linear equivalent PLL model in presence of additive noise*

To be specific,

$$n_2(t) = \frac{n_1(t)}{A_c} = \frac{n_c(t)}{A_c}\sin\Delta\varphi(t) - \frac{n_s(t)}{A_c}\cos\Delta\varphi(t) \qquad 5.31.21$$

Now the variance of phase error Δφ(t) is also the variance of the VCO output. After some analysis, this variance can be shown as:

$$\sigma^2_{\hat{\varphi}} = \frac{2N_0B_{eq}}{A_c^2} = \frac{1}{\gamma_L}, \text{ where } \gamma_L = loopSNR = \frac{A_c^2/2}{N_0B_{eq}} \qquad 5.31.22$$

An important message from the above summary on the principle of PLL is that, the variance of the VCO output, which should be small for the purpose of carrier synchronization, is inversely related to the loop SNR and hence much depends on how nicely the PLL is designed.

Costas Loop

2nd method for generating carrier phase ref. for DSB – SC type signal.

$$y_c(t) = \left[s(t) + n(t)\right]\cos\left(w_c t + \bar{\varphi}\right)$$

$$= \frac{1}{2}\left[A(t) + n_c(t)\right]\cos\Delta\varphi + \frac{1}{2}n_s(t)\sin\Delta\varphi + \text{double frequency term}$$

$$y_s(t) = \left[s(t) + n(t)\right]\sin\left(w_c t + \bar{\varphi}\right)$$

$$= \frac{1}{2}\left[A(t) + n_c(t)\right]\sin\Delta\varphi - \frac{1}{2}n_s(t)\cos\Delta\varphi + \text{double frequency term}$$

Refer **Fig 5.31.4** for the block diagram of a Costas Loop

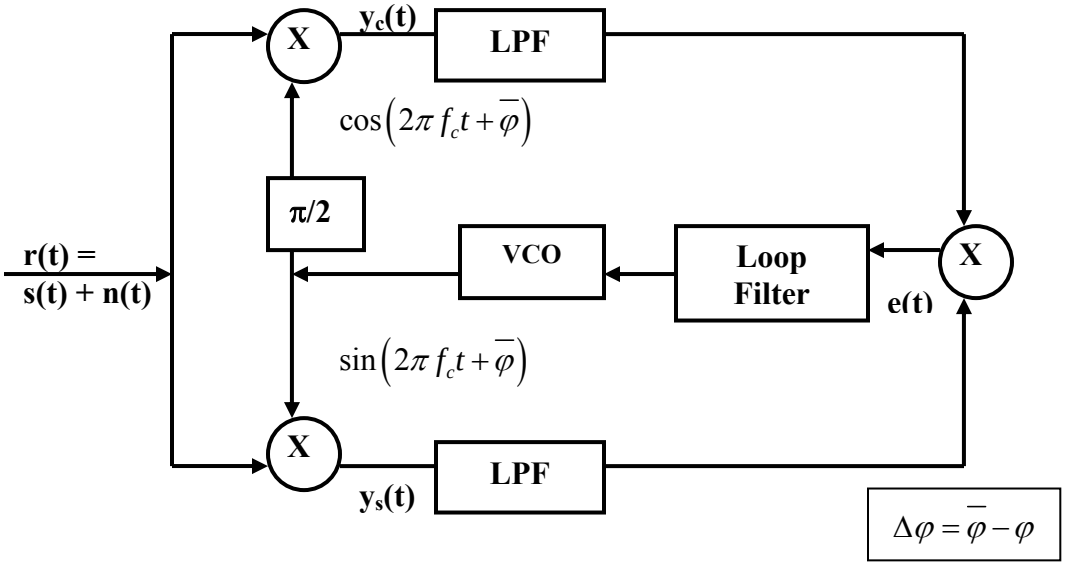


Fig. 5.31.4 *Block diagram of a Costas Loop*

Error signal

e(t) = multiplication of the low frequency components of $y_c(t)$ and $y_s(t)$

$$= \frac{1}{4}\left\{\left(\left[A(t) + n_c(t)\right]\cos\Delta\varphi + n_s(t)\sin\Delta\varphi\right)\times\left(\left[A(t) + n_c(t)\right]\sin\Delta\varphi - n_s(t)\cos\Delta\varphi\right)\right\}$$

$$= \left\{\frac{1}{8}\left(\left[A(t) + n_c(t)\right]^2\sin 2\Delta\varphi - n_s^2(t)\sin 2\Delta\varphi\right) + \frac{1}{4}n_s(t)\left[A(t) + n_c(t)\right]\cos 2\Delta\varphi\right\}$$

$$= \left\{\frac{1}{8}\left(\left[A(t) + n_c(t)\right]^2 - n_s^2(t)\right)\sin 2\Delta\varphi + \frac{1}{4}n_s(t)\left[A(t) + n_c(t)\right]\cos 2\Delta\varphi\right\}$$

This composite error signal is filtered by the loop filter to generate the control voltage.

Note that the desired term is:

$$\frac{1}{8}A^2(t)\sin 2\left(\bar{\varphi}-\varphi\right)$$

The other terms are (signal × noise) and (noise × noise) type. For a good design of the loop filter, performance similar to a squaring loop may be obtained [without using a square circuit]. Also, the LPF in the I & Q path, are identical to the matched filters, matched to the signal pulse.

Decision Feedback Loop

[For DSB-SC type mod.]

$$y_c(t)=r(t)\cos\left(w_c t+\bar{\varphi}\right)=\left[s(t)+n(t)\right]\cos\left(w_c t+\bar{\varphi}\right)$$

$$=\frac{1}{2}\left[A(t)+n_c(t)\right]\cos\Delta\varphi+\frac{1}{2}n_s(t)\sin\Delta\varphi+\text{double frequency term}$$

This 'y_c (t)' is used to recover information A(t).

Now the error input e(t) to the loop filter, in absence of decision error is:

$$e(t)=\frac{1}{2}A(t)\left\{\left[A(t)+n_c(t)\right]\sin\Delta\varphi-n_s(t)\cos\Delta\varphi\right\}+\text{double frequency term}$$

$$=\frac{1}{2}A^2(t)\sin\Delta\varphi+\frac{1}{2}A(t)\left[n_c(t)\sin\Delta\varphi-n_s(t)\cos\Delta\varphi\right]+\text{double frequency term}$$

— Performs better than PLL or Costas Loop, if BER<10^{-2} [4 to 10 times better]

Refer **Fig 5.31.5** for the block diagram of a Decision Feedback Loop

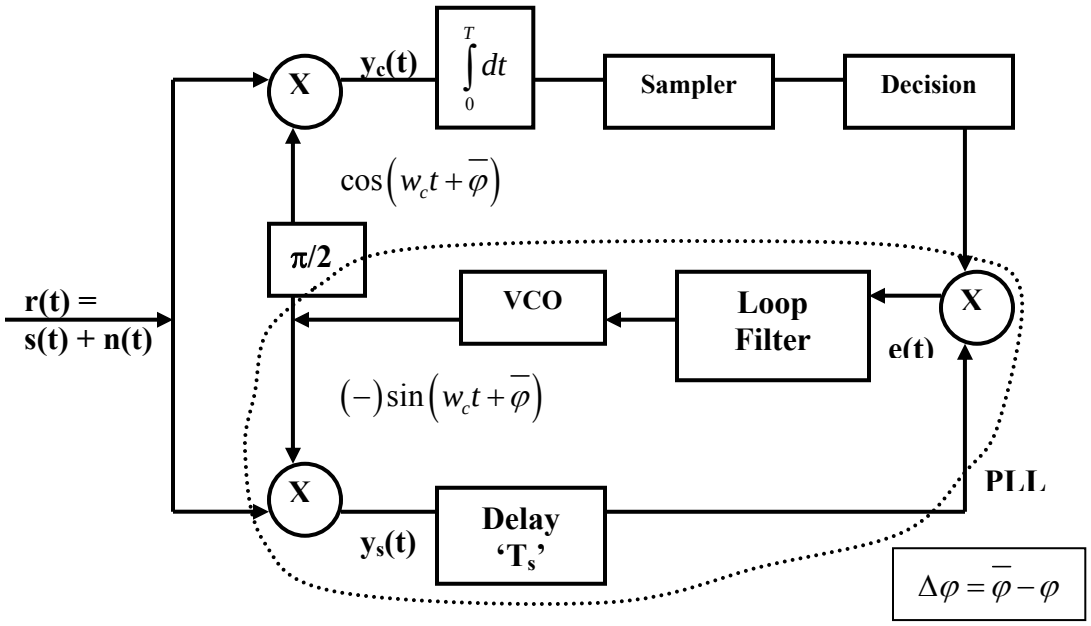


Fig. 5.31.5 *Block diagram of a decision feedback loop*

Decision Feedback PLL for M-ary PSK Modulation

This is a relatively simple scheme with good performance

$$s(t) = A_c \cos\left[w_c t + \varphi + \frac{2\pi}{M}(m-1)\right], \qquad m=1,2,\ldots,M.$$

$$= A_c \cos\left[w_c t + \varphi + \theta_m\right]$$

The received signal is of the form:

$$r(t) = s(t) + n(t)$$

The carrier recovery / tracking scheme removes the information-dependent phase component to obtain $\cos(w_c t + \varphi)$ as the received phase reference.

The received signal is demodulated (using coherent demodulator) to obtain a phase estimate

$$\overline{\theta_m} = \frac{2\pi}{M}(m-1)$$

In absence of decision error, $\overline{\theta_m} = \theta_m$, the transmitted signal phase.

Refer **Fig 5.31.6** for the block diagram of a Decision Feedback Loop for PSK Modulation.

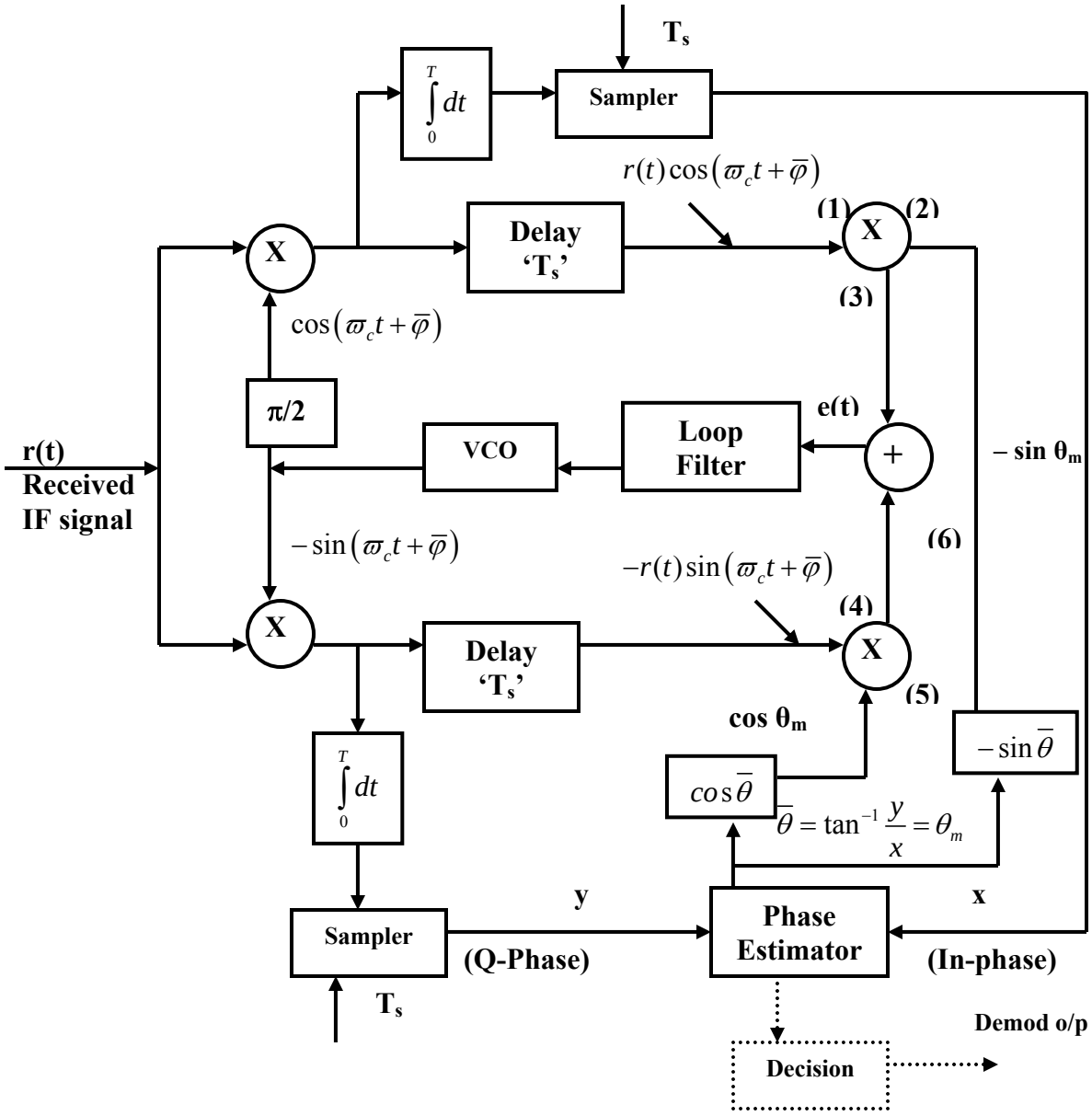


Fig. 5.31.6 *Block diagram of a decision feedback loop for PSK modulations*

Let us assume that there is no decision error and hence, $\overline{\theta_m} = \overline{\theta} = \theta_m$

Now the signal at (1) in the diagram may be written as $\gamma(t)\cos\left(w_c t + \overline{\varphi}\right)$.

Assuming no decision error, the signal at (2) is $-\sin\overline{\theta} = -\sin\overline{\theta_m} = -\sin\theta_m$

At (3), the signal is $-\gamma(t)\cos\left(w_c t + \overline{\varphi}\right).\sin\theta_m$

$= -\left[s(t) + n(t)\right]\cos\left(w_c t + \overline{\varphi}\right).\sin\theta_m$

$$= -\left\{A_c \cos\left[w_c t + \varphi + \theta_m\right] + n_c(t)\cos\left[w_c t + \varphi\right] - n_s(t)\sin\left[w_c t + \varphi\right]\right\}\cos\left[w_c t + \overline{\varphi}\right].\sin\theta_m$$

$$= -A_c \sin\theta_m \cos\left(w_c t + \varphi + \theta_m\right)\cos\left(w_c t + \overline{\varphi}\right) \quad \ldots\ldots\ldots\ldots\ldots\ldots \text{ (A)}$$

$$-n_c(t)\sin\theta_m \cos\left(w_c t + \varphi\right)\cos\left(w_c t + \overline{\varphi}\right) \quad \ldots\ldots\ldots\ldots\ldots\ldots \text{ (B)}$$

$$+n_s(t)\sin\theta_m \sin\left(w_c t + \varphi\right)\cos\left(w_c t + \overline{\varphi}\right) \quad \ldots\ldots\ldots\ldots\ldots\ldots \text{ (C)}$$

Now consider the terms at (A):

$$-A_c \sin\theta_m \cos\left(w_c t + \varphi + \theta_m\right)\cos\left(w_c t + \overline{\varphi}\right)$$

$$= -A_c \sin\theta_m \left(\frac{1}{2}\right)\left\{\cos\left(2w_c t + \varphi + \overline{\varphi} + \theta_m\right) + \cos\left(\varphi - \overline{\varphi} + \theta_m\right)\right\}$$

Of interest are the low frequency terms as we are using a loop filter later:
Low frequency terms in (A):

$$-\frac{1}{2}A_c \sin\theta_m \cos\left(\varphi - \overline{\varphi} + \theta_m\right)$$

$$= -\frac{1}{2}A_c \sin\theta_m \left[\cos\left(\varphi - \overline{\varphi}\right)\cos\theta_m - \sin\left(\varphi - \overline{\varphi}\right)\sin\theta_m\right]$$

Now consider terms (B):

$$-n_c(t)\sin\theta_m \cos\left(w_c t + \varphi\right)\cos\left(w_c t + \overline{\varphi}\right)$$

$$= -\frac{1}{2}n_c(t)\sin\theta_m \left[\cos\left(2w_c t + \varphi + \overline{\varphi}\right) + \cos\left(\varphi - \overline{\varphi}\right)\right]$$

Considering Low Frequency component:

$$-\frac{1}{2}n_c(t)\sin\theta_m \cos\left(\varphi - \overline{\varphi}\right)$$

Now consider terms (C):

$$n_s(t)\sin\theta_m \sin\left(w_c t + \varphi\right)\cos\left(w_c t + \overline{\varphi}\right)$$

$$= \frac{1}{2}n_s(t)\sin\theta_m \left[\sin\left(2w_c t + \varphi + \overline{\varphi}\right) + \sin\left(\varphi - \overline{\varphi}\right)\right]$$

Considering low frequency term:

$$\frac{1}{2}n_s(t)\sin\theta \sin\left(\varphi - \overline{\varphi}\right)$$

So, the overall low – frequency component at (3):

$(A)_{LF} + (B)_{LF} + (C)_{LF}$

$$= -\frac{1}{2}A_c \sin\theta_m \left[\cos\theta_m \cos\left(\varphi - \overline{\varphi}\right) - \sin\theta_m \sin\left(\varphi - \overline{\varphi}\right)\right]$$

$$-\frac{1}{2}n_c(t)\sin\theta_m\cos\left(\varphi-\overline{\varphi}\right)+\frac{1}{2}n_s(t)\sin\theta_m\sin\left(\varphi-\overline{\varphi}\right)$$

$$=-\frac{1}{2}\left[A_c\cos\theta_m+n_c(t)\right]\sin\theta_m\cos\left(\varphi-\overline{\varphi}\right)$$

$$+\frac{1}{2}\left[A_c\sin\theta_m+n_s(t)\right]\sin\theta_m\sin\left(\varphi-\overline{\varphi}\right)$$

By similar straightforward expansion, the overall low-frequency term at (6) may be shown as:

$$\frac{1}{2}\left[A_c\cos\theta_m+n_c(t)\right]cos\,\theta_m\sin\left(\varphi-\overline{\varphi}\right)+\frac{1}{2}\left[A_c\sin\theta_m+n_s(t)\right]\cos\theta_m\cos\left(\varphi-\overline{\varphi}\right)$$

These two signals [at (3) and (6)] are added to obtain the error signal e(t):

$$e(t)=\frac{1}{2}A_c\left\{\begin{matrix}-\cos\theta_m\sin\theta_m\cos\left(\varphi-\overline{\varphi}\right)+\sin^2\theta_m\sin\left(\varphi-\overline{\varphi}\right)\\+cos^2\theta_m\sin\left(\varphi-\overline{\varphi}\right)+\sin\theta_m\cos\theta_m\cos\left(\varphi-\overline{\varphi}\right)\end{matrix}\right\}$$

$$+\frac{1}{2}n_c(t)\sin\left(\varphi-\overline{\varphi}-\theta_m\right)+\frac{1}{2}n_s(t)cos\left(\varphi-\overline{\varphi}-\theta_m\right)$$

$$=\frac{1}{2}A_c\sin\left(\varphi-\overline{\varphi}\right)+\frac{1}{2}n_c(t)\sin\left(\varphi-\overline{\varphi}-\theta_m\right)+\frac{1}{2}n_s(t)cos\left(\varphi-\overline{\varphi}-\theta_m\right)$$

Module 5

Carrier Modulation

Lesson 32

Timing Synchronization

After reading this lesson, you will learn about:

- ***Bit Error Rate (BER) calculation for BPSK;***
- ***Error Performance of coherent QPSK;***
- ***Approx BER for QPSK;***
- ***Performance Requirements;***

All digital communication systems require various timing control measures for specific purposes. For example, timing information is needed to identify the rate at which bits are transmitted. It is also needed to identify the start and end instants of an information-bearing symbol or a sequence of symbols. Note that all the demodulation schemes that we have discussed are based on the principle of symbol-by-symbol detection scheme and we assumed that precise symbol-timing information is always available at the receiver.

Further, information, when available in binary digits, is often treated in groups called blocks. A block is a small segment of data that is treated together for the purpose of transmission and reception. Each block is added with time stamps marking the beginning and end of the block and these time stamps should also be recovered properly at the receiving end to ensure proper sequence of blocks at the user end. In the context of radio transmission and reception, such as for communication through a satellite or in wireless LAN, a block of binary digits called 'frame', along with necessary overhead bits, needs synchronization.

For reliable data communication at moderate and high rates, timing information about the transmitter clock is obtained in the receiver directly or indirectly from the received signal. Such a transmission mode is known as synchronous. In this lesson, we will discuss about synchronous mode of digital transmission, primarily applicable for wireless communications. Though an additional channel may be used in a communication system to transmit the timing information, it is wastage of bandwidth. For baseband transmission schemes, it is a popular practice to insert the timing signal within the transmitted data stream by use of suitable line encoding technique.

A straight forward approach to insert the timing signal in a binary data stream is to encode the binary signal in some way to ensure a high-low (or low-high) transition with each bit. Such a transition in each bit can be used easily to recover the time reference (e.g. the clock) at the receiver.

Non-Return to Zero (NRZ) Encoding

In this encoding scheme, logic '1' is sent as a high value and logic '0' is sent as a low value (or vice versa). This is simple but not an elegant technique. A long run of '0' or '1' will have no transition for a fairly long duration of time and this may cause drift in the timing recovery circuit at the receiver.

Return to Zero (RZ) Encoding:

This is a three level encoding scheme where logic '1' is encoded as a high positive value for a portion (typically 50 %) of the bit duration while nothing is transmitted over the remaining duration. Similarly, logic '0' is encoded as a negative value for a portion (typically 50 %) of the bit duration while nothing is transmitted over the remaining duration. So, a long run of '1' or '0' will have level transitions.

Manchester Encoding:

In Manchester encoding the actual binary data to be transmitted are not sent as a sequence of logic 1's and 0's. The bits are translated into a different format which offers several advantages over NRZ coding. A logic '0' is indicated by a transition from '0' to '1' at the middle of the duration of the bit while logic '1' is indicated by a transition from '1' to '0' at the middle of the duration of the bit. So, there is always a transition at the centre of each bit. The Manchester code is also known as a Biphase Code as a bit is encoded by $\pm$ 90^0 phase transition. As a Manchester coded signal has frequent level transitions, the receiver can comfortably extract the clock signal using a phase locked loop. However, a Manchester coded signal needs more bandwidth than the NRZ coded signal. Thus this line coding scheme finds more use in optical fiber communications systems where additional bandwidth is available.

Symbol Synchronization

We will now discuss about time synchronization techniques, specifically necessary for demodulating narrowband carrier modulated signals. Such techniques are also known as 'symbol synchronization' techniques.

Let us recollect from our discussion on matched filtering that if a signal pulse s(t), $0 \leq t \leq T$, is passed through a filter, matched to it, the output of the filter is maximum at $t = T$.

Example 5.32.1: Consider a rectangular pulse s(t) as shown in **Fig. 5.32.1(a).** The corresponding matched filter output is sketched in **Fig. 5.32.1(b).** It may be observed that, if we sample the filter output early, i.e., at $t = T - \delta$ or late, at $t = T + \delta$, the sampled values are not maximum. In presence of noise (AWGN), it implies that the average sampled value may result in wrong symbol decision. A symbol synchronization scheme attempts to ensure that even in presence of noise, the time offset 'δ' is acceptably small.

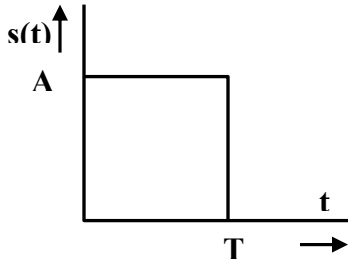


Fig. 5.32.1(a): *A rectangular pulse s(t) of duration T*

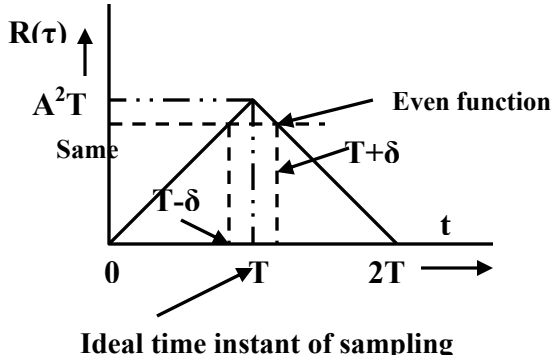


Fig. 5.32.1(b): *Autocorrelation of a rectangular pulse s(t)*

Now, note that the auto correlation function R (τ) is an even function & its values for t = T ± δ are the same. So, the aim of a symbol synchronization scheme is to ensure small 'δ' first and then declare the ideal time instant of sampling as the mid-point of 'T-δ' and 'T+ δ'. This is also the basic approach of a symbol synchronizer popularly known as *'Early-Late gate Synchronizer'* **[Fig. 5.32.2(a)]**. The received IF signal is multiplied by the recovered carrier $\cos(\omega_c t + \hat{\varphi})$ and the resultant signal is treated in two parallel paths. In the upper branch, the signal is correlated with a little advanced version of the expected ideal symbol pulse while it is correlated with a little delayed version of the symbol pulse. Outputs of the two correlators are then sampled at the same time instant which gets corrected depending on the difference of the two squaring circuit outputs. If the present time instant of sampling is very close to the ideal sampling instant, the two squaring circuits, following the two correlators, produce very similar outputs and hence the low frequency component at the output of the difference unit is almost zero. The VCO in this situation is locked to the desired clock frequency with correct phase. So, In essence, the VCO output is the desired symbol clock which can be used for the purpose of demodulation. Another way of implementing the same concept of early-late-gate synchronizer, which is sometimes easier for practical implementation, is shown in **Fig. 5.32.2(b)**.

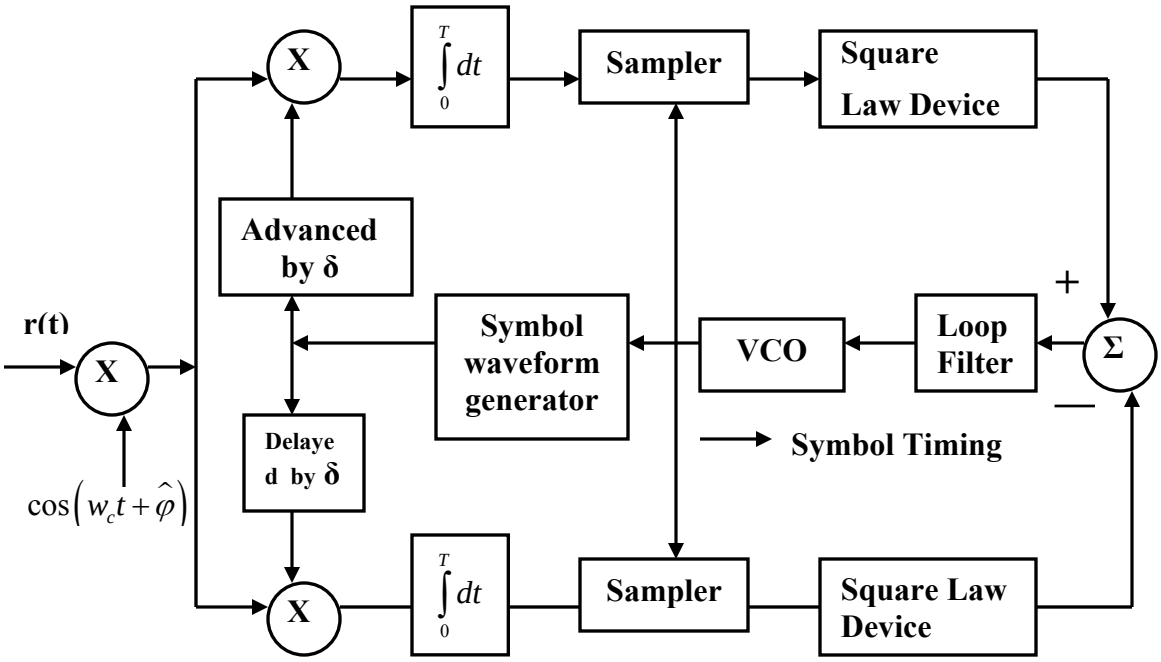


Fig. 5.32.2 (a): *Schematic diagram of an Early-Late-Gate synchronizer*

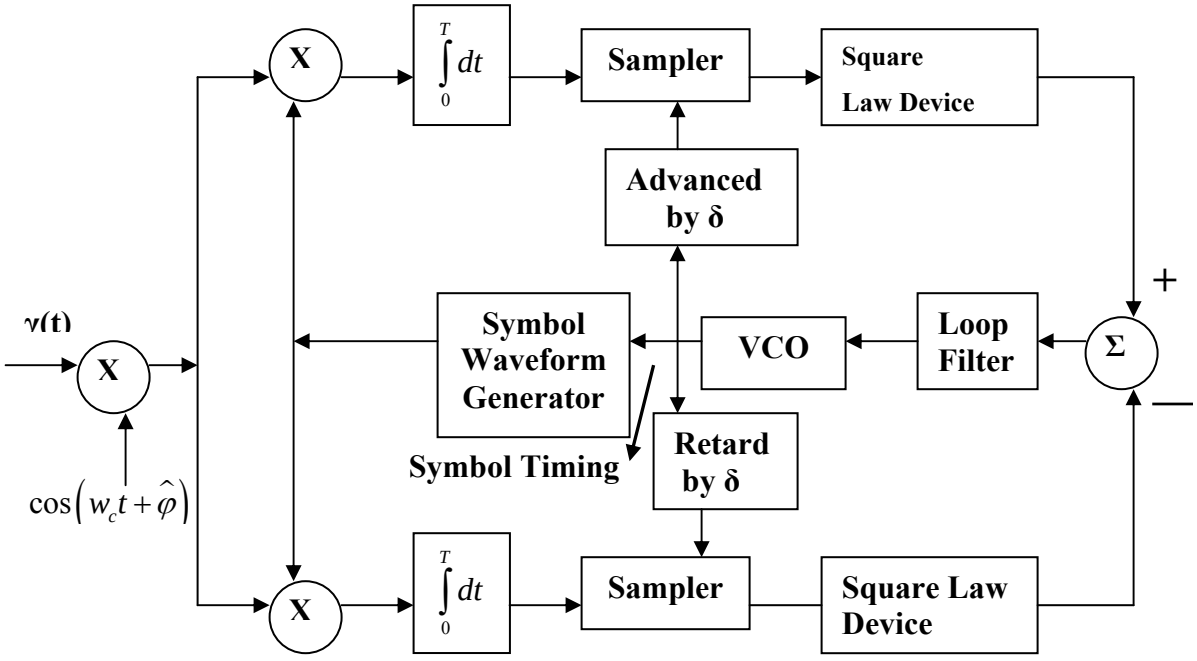


Fig. 5.32.2 (b): *An alternative scheme for Early-Late-Gate synchronizer*

Module 6

Channel Coding

Lesson 33

Introduction to Error Control Coding

After reading this lesson, you will learn about

- ***Basic concept of Error Control Coding;***
- ***Categorizes of error control processes;***
- ***Factors describing FEC codes;***
- ***Block Codes – Encoding and Decoding;***

The primary function of an error control encoder-decoder pair (also known as a codec) is to enhance the reliability of message during transmission of information carrying symbols through a communication channel. An error control code can also ease the design process of a digital transmission system in multiple ways such as the following:

a) The transmission power requirement of a digital transmission scheme can be reduced by the use of an error control codec. This aspect is exploited in the design of most of the modern wireless digital communication systems such as a cellular mobile communication system.

b) Even the size of a transmitting or receiving antenna can be reduced by the use of an error control codec while maintaining the same level of end-to-end performance [example: VSAT (Very Small Aperture Terminal) network terminals].

c) Access of more users to same radio frequency in a multi-access communication system can be ensured by the use of error control technique [example: cellular CDMA].

d) Jamming margin in a spread spectrum communication system can be effectively increased by using suitable error control technique. Increased jamming margin allows signal transmission to a desired receiver in battlefield and elsewhere even if the enemy tries to drown the signal by transmitting high power in-band noise.

In this section we present a short overview of various error control codes.

Since C. E. Shannon's pioneering work on mathematical theory for digital communications in 1948-49, the subject of error control coding has emerged as a powerful and practical means of achieving efficient and reliable communication of information in a cost effective manner. Suitability of numerous error control schemes in digital transmission systems, wire line and wireless, has been studied and reported in detail in the literature.

The major categories of activities on error control coding can broadly be identified as the following:

a) to find codes with good structural properties and good asymptotic error performance,

b) to devise efficient encoding and decoding strategies for the codes and

c) to explore the applicability of good coding schemes in various digital transmission and storage systems and to evaluate their performance.

The encoding operation for a (n, k) error control code is a kind of mapping of sequences, chosen from a k-dimensional subspace to a larger, n-dimensional vector space of n-tuples defined over a finite field and with n > k. Decoding refers to a reverse mapping operation for estimating the probable information sequence from the knowledge of the received coded sequence. If the elements (bit, dibit or a symbol made of group of bits) of the message sequence at the input to the encoder are defined over a finite field of q_i elements and the sequence elements at the output of the encoder are defined over (same or a different) finite field with q_o elements, the code rate or 'coding efficiency' R of the code is defined as:

$R = (L_{in} \log_2 q_i) / (L_{out} \log_2 q_o)$, where L_{in} and L_{out} denote the lengths of input and output sequences respectively. The code rate is a dimensionless proper fraction.

For a binary code, $q_i = q_o = 2$ and hence, $R = L_{in} / L_{out}$. A (7,4) Hamming code is an example of a binary block code whose rate R = 4/7. This code will be addressed later in greater detail. For an error correction code, R < 1.0 and this implies that some additional information (in the form of 'parity symbol' or 'redundant symbol') is added during the process of encoding. This redundant information, infused in a controlled way, help us in decoding a received sequence to extract a reliable estimate of the information bearing sequence.

Now, it is interesting to note that the purpose of error control can be achieved in some situations even without accomplishing the complete process of decoding. Accordingly, the process of error control can be categorized into the following:

a) Forward Error Correction (FEC)

Complete process of decoding is applied on the received sequence to detect error positions in the sequence and correct the erroneous symbols. However, the process of error correction is not fool-proof and occasionally the decoder may either fail to detect presence of errors in a received sequence or, may detect errors at wrong locations, resulting in a few more erroneous symbols. This happens if, for example, too much noise gets added to the signal during transmission through a wireless channel.

b) Auto Repeat reQuest (ARQ)

In some applications (such as in data communications) it is important to receive only error-free information, even if it means more than usual delay in transmission and reception. A conceptually simple method of error detection and retransmission is useful in such situations. The error control decoder, at the receiver, only checks the presence of any error in a received sequence (this is a relatively easy task compared to full error correction). In case any error is detected, a request is sent back to the transmitter (via return channel, which must be available for this purpose) for re-transmitting the sequence (or packet) once again. The process ideally continues till an error-free sequence is

received and, this may involve considerable delay in receipt and may result in delay for subsequent sequences.

Another aspect of this scheme is that the transmitter should have enough provision for storing new sequences while a packet is repeated several times. One may think of several interesting variations of the basic scheme of ARQ. Three important and popular variations are: i) Stop and Wait ARQ, ii) Continuous ARQ and iii) Selective Repeat ARQ.

c) Hybrid ARQ

After a brief recollection of the above two error control schemes, viz. FEC and ARQ, one may suggest combining the better features of both the schemes and this is, indeed, feasible and meaningful. Significant reduction in retransmission request is possible by using a moderately powerful FEC in an ARQ scheme. This saves considerable wastage in resources such as time and bandwidth and increases the throughput of the transmission system at an acceptably small packet error rate compared to any ARQ scheme with only error detection feature. This scheme is popular especially in digital satellite communication systems.

Henceforth, in this section we focus on some additional features of FEC schemes only. Application of an FEC code and a judicious choice of the code parameters are guided by several conflicting factors. Some of these factors are described in brief:

a) Nature of communication channel

Effects of many physical communication channel manifest in random and isolated errors while some channels cause bursty errors. The modulation technique employed for transmission of information, sensitivity level of a receiver (in dBm), rate of information transmission are some other issues.

b) Available channel bandwidth

As mentioned, use of an error-control scheme involves addition of controlled redundancy to original message. This redundancy in transmitted massage calls for larger bandwidth than what would be required for an encoded system. This undesirable fact is tolerable because of the obtainable gains or advantages of coded communication system over an encoded one for a specified overall system performance in terms of BER or cost.

c) Hardware complexity cost and delay

Some FEC codes of larger block length asymptotically satisfy the requirements of high rate as well as good error correcting capability but the hardware complexity,

volume, cost and decoding delay of such decoders may be enormous. For a system designer, the choice of block length is somewhat limited.

d) The coding gain

FEC codes of different code rates and block sizes offer different coding gains in E_b/N_o over an uncoded system. At the first level, the coding gain is defined as:

[($\frac{E_b}{N_0}$ in dB needed by a uncoded system to achieve a specified BER of 10^{-x}) –

($\frac{E_b}{N_0}$ in dB needed by an FEC coded system to achieve the same BER of 10^{-x}).]

Fig. 6.33.1 shows a tree classifying some FEC codes based on their structures.

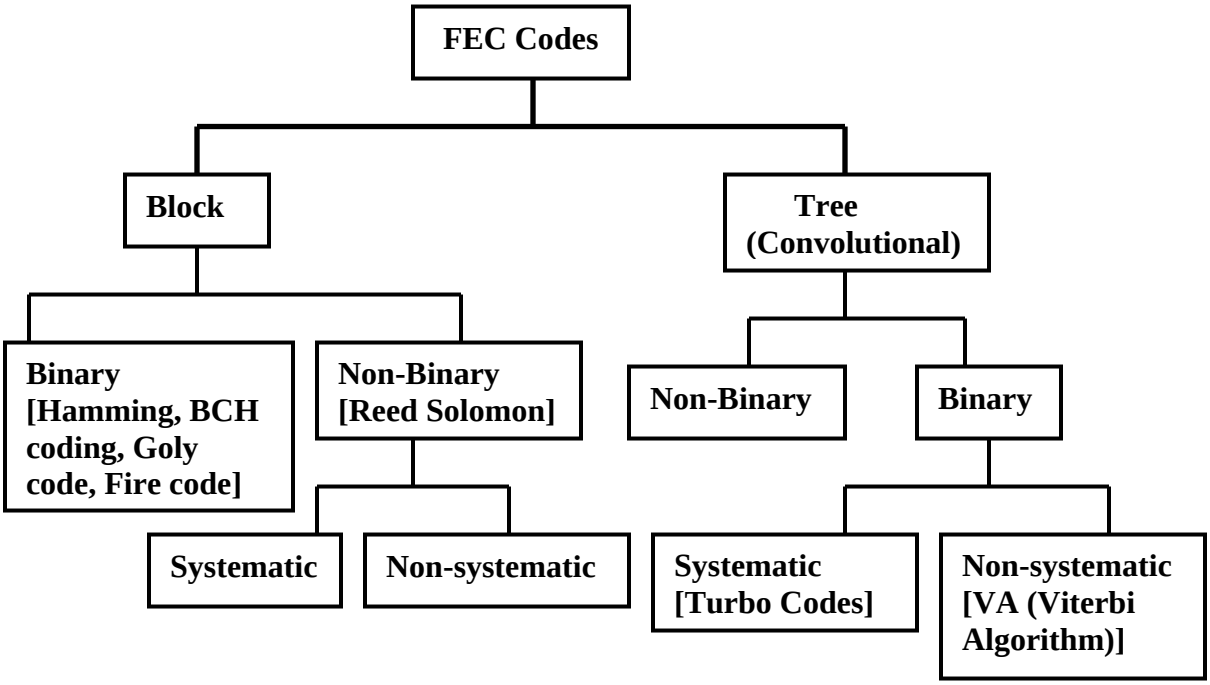


Fig. 6.33.1 *Classification of FEC codes*

Block Codes

The encoder of a block code operates on a group of bits at a time. A group or 'block' of 'k' information bits (or symbols) are coded using some procedure to generate a larger block of 'n' bits (or symbols). Such a block code is referred as an *(n ,k)* code.

Encoding of block code

The encoder takes k information bits and computes $(n-k)$ parity bits from these bits using a specific code generator matrix. A codeword of 'n' bits (or symbols) is

generated in the process. This operation may be 'systematic' or 'nonsystematic'. In a systematic encode the first (or last) k bits in a codeword are the information bits and the rest $(n-k)$ bits are the parity check bits. For a nonsystematic code, the information symbols do not occupy such fixed positions in a codeword. In fact, they may not be identified by a simple mean.

Following matrix notation, the encoding operation can be described as a matrix multiplication:

$$\mathbf{C} = \mathbf{d}.\mathbf{G} \qquad 6.33.1$$

where, **d** : information matrix and **G** : generator matrix.

For a systematic block code, the generator matrix can be expressed as

$$G = [I_k / P],$$

where 'I' is a (k x k) identity matrix and 'P' is a (k x [n-k]) parity check matrix.

Following an equivalent polynomial notation, the encoder starts with a 'message polynomial' defined as below.

$$m(x) = m_0 + m_1 x + m_2 x_2 + ... + m_{k-1} x_{k-1} = \sum_{i=0}^{k=1} m_i x_i \qquad 6.33.2$$

Here m_i's are the information bits (or symbols) and 'x' is an indeterminate representing unit delay. The exponents of 'x' indicate number of unit delays. For example, the above polynomial indicates that the first bit of the information sequence is 'm_0', the second bit is 'm_1' and the k-th bit is 'm_k'.

For a binary block code, m_i's are '0' or '1', i.e. they are elements of Galois Field [GF(2)]. GF(2) is a finite field consisting of two elements. The '+' in the above expression indicates the 'addition' operation as defined in the finite field. Though addition of m_0 and $m_1 x$ does not mean anything, the 'addition' operator is useful in adding two or more polynomials.
For example, if,

$$p(x) = p_0 + p_1 x + p_2 x^2 + ... + p_{k-1} x^{k-1} \text{ and } q(x) = q_0 + q_1 x + q_2 x^2 + ... + q_{k-1} x^{k-1}$$

then $p(x) + q(x) = (p_0 + q_0) + (p_1 + q_1) x + (p2 + q_2) x^2 + ...$

Interestingly, over GF(2), the 'addition' and 'subtraction' operations are the same and following Boolean logic, it is equivalent to Exclusive-OR operation.

The codeword polynomial c(x) is defined as:

$$c(x) = \sum_{i=1}^{n-1} c_i x^i = c_0 + c_1 x + ...c_{n-1} x^{n-1} \qquad 6.33.3$$

For a binary code $c_i \in GF(2)$. Note that the codeword polynomial is of degree 'n-1', implying that there are 'n' coefficients in the polynomial. The encoding strategy is described using a 'generator polynomial' g(x):

$$g(x) = \sum_{i=0}^{n-k} g_i x^i = g_0 + g_1 x + ... + g_{n-k} x^{n-k} \qquad 6.33.4$$

The generator polynomial g(x) is of degree (n-k), implying that it has (n-k)+1 coefficients. The coefficient of x^0 is '1' for all binary codes.

The nonsystematic encoding procedure for a block code is described as a 'multiplication' of the message polynomial and the generator polynomial:

$$c(x) = m(x).g(x) \qquad 6.33.5$$

The codeword for systematic encoding is described as:

$$c'(x) = x^{n-k}.m(x) - R_{g(x)}\left[x^{n-k}.m(x)\right] \qquad 6.33.6$$

Here, $R_{g(x)}[y(x)]$ denotes the remainder polynomial when y(x) is divided by g(x). So, the degree of the remainder polynomial is (n-k-1) or less. The polynomial $x^{n-k}.m(x)$ denotes a shifted version of the message polynomial m(x), delayed by (n-k) units.

Note that irrespective of whether a code is systematic or nonsystematic, a codeword polynomial c(x) or $c'(x)$ is fully divisible by its generator polynomial. This is an important property of block codes which is exploited in the receiver during decoding operation. Another interesting point is that the generator polynomial of a binary block code happens to be a valid codeword having minimum number of '1'-s as its constant coefficients g_i-s.

An intuitive approach to decoding for block codes

With 'k' information bits in a message block, the number of possible message patterns is 2^k. Now, imagine a 'k' dimensional signal space, where each block of 'k' information bits, representing a k-tuple, is a point. The 'k' dimensional signal space is completely filled with all possible message patterns. The operation of encoding adds (n-k) redundant bits. By this, a point from 'k' dimensional filled space is mapped to a bigger n-dimensional signal space which has 2^n positions. Mapping for each point is one to one. Let us call the bigger space as the code space. So, out of 2^n points, only 2^k points will be occupied by the encoded words and the other possible points in the n-dimensional code space remain vacant. So, we can talk about a 'distance' between two codeword.

A popular measure for distance between two code words is called Hamming distance (d_H), which is the number of places in which two binary tuples differ in the code space. An encoding scheme is essentially a strategy of mapping message tuples into a code space. Let, d_{Hmin} indicate the minimum Hamming distance among any two code words in the code space. A good encoding strategy tries to maximize d_{Hmin}. Higher the d_{Hmin}, more robust or powerful is the code in terms of error detection and error correction capability.

For a block code, if 'e_d' denotes the number of errors it can detect in a code word and't' denotes the number of errors it can correct in a received word, then

$$e_d = d_{H_{min}} - 1 \text{ and } t = \frac{d_{H_{min}} - 1}{2} \qquad 6.33.7$$

Following the matrix description approach, a parity check matrix H is used for decoding several block codes. The H matrix is related to the generator matrix G and if 'C' is the matrix of encoded bits,

$$C.H^T = 0 \qquad 6.33.8$$

But during transmission or due to imperfect reception, the matrix Y of received bits may be different from C:

$$Y = C + e \qquad 6.33.9$$

where 'e' denotes an error vector.

$$\text{Now, } Y.H^T = (C+e).H^T = C.H^T + e.H^T \qquad 6.33.10$$

The matrix $S = e.H^T$ is known as a 'syndrome matrix'. It is interesting to note that the syndrome matrix S is independent of the coded matrix C. It is dependent only on the error vector and the parity check matrix. So, the decoder attempts to recover the correct error vector from the syndrome vector. A null syndrome matrix mostly means that the received matrix Y is error-free. In general, the relationship between S and the error vector 'e' is not one-to-one. For a given H, several error vectors may result in the same syndrome S. So, the decoder specifically attempts to make a best selection from a set of possible error vectors than could result in a specific syndrome S. The procedure may turn out to be very involved in terms of number of computations or time etc. As a compromise, some 'incomplete' decoding strategies are also adopted in practice. The family of 'Algebraic Decoding' is practically important in this regard.

Polynomial Description: Let us define two polynomials:

r(x) = Received word polynomial = $\sum_{i=0}^{n-1} r_i x^i = r_0 + r_1 x + ... + r_{n-1} x^{n-1}$

and e(x) = Error Polynomial $= \sum_{i=0}^{n-1} e_i x^i = e_0 + e_1 x + ... + e_{n-1} x^{n-1}$, 6.33.11

where for a binary code, $r_i \in GF(2)$, $e_i \in GF(2)$ and $r(x) = c(x) \oplus e(x)$. 'c(x)' is the transmitted code word polynomial. The notations '+' and $\oplus$ are equivalent. Both the notations are used by authors.

Note that, the job of the decoder is to determine the most probable error vector e(x) after receiving the polynomial r(x). The decoder has complete knowledge of the g(x), i.e. the encoding strategy. The decoder divides whatever r(x) it receives by g(x) and looks at the remainder polynomial:

$$R_{g(x)}[r(x)] = R_{g(x)}[c(x) + e(x)]$$

$$= R_{g(x)}\left[c(x)\right] \oplus R_{g(x)}\left[e(x)\right]$$
$$= 0 + R_{g(x)}\left[e(x)\right] \qquad 6.33.12$$

We know that c(x) is divisible by g(x). So, if the remainder is non zero, then r(x) contains one or more errors. This is the error detection part of decoding. If there is error, the decoder proceeds further to identify the positions where errors have occurred and then prepares to correct them. The remainder polynomial, which is of degree (n-k-1) or less, is called the syndrome polynomial:

$$s(x) = R_{g(x)}\left[e(x)\right] \qquad 6.33.13$$

The syndrome polynomial plays an important role in algebraic decoding algorithms and similar to the S matrix, is dependent only on the errors, though multiple error sequences may lead to the same s(x).

Module 6

Channel Coding

Lesson 34

Block Codes

After reading this lesson, you will learn about

- ***Hamming Code;***
- ***Hamming Decoder;***
- ***Reed-Solomon Codes;***

In this lesson, we will have short and specific discussions on two block codes, viz. the Hamming code and the Reed- Solomon code. The Hamming code is historically important and it is also very suitable for explaining several subtle aspects of encoding and decoding of linear binary block codes. The Reed- Solomon code is an important example of linear non-binary code, which can correct random as well as burst errors. It is one of the most extensively applied block codes in digital transmission an digital storage media.

Hamming Code

Invented in 1948 by R. W. Hamming, this single error correcting code is an elegant one. It should be easy to follow the encoding and decoding methods. Though the original code, developed by Hamming, was a specific (7,4) code with n = 7 and k = 4, subsequently all single error correcting linear block codes have come to be known as Hamming codes too.

In general, a Hamming code is specified as $\left(2^m-1, 2^m-m-1\right)$, i.e., a block of k = $\left(2^m-m-1\right)$ bits are encoded together to produce a codeword of length n = $\left(2^m-1\right)$bits. So, 'm' parity bits are added for every 'k' information bits. Some examples of Hamming codes are (7, 4), (15, 11), (31, 26) and (63, 57). All these codes are capable of detecting 2 errors and correcting single error as the Hamming distance for them is 3.

By adding an overall parity bit, an (n, k) Hamming code can be modified to yield an (n+1, k+1) code with minimum distance 4. Such a code is known as extended Hamming code.

We follow the polynomial approach to describe the (7,4) Hamming code. The generator polynomial for (7,4) Hamming code is:

g(x)= Generator polynomial = x^3+x+1 6.34.1

Here, the coefficients of x^i –s are elements of GF(2), i.e. the coefficients are '0' or '1'.

However, all the polynomials are defined over an 'extended' field of GF(2). The extension field for (7,4) Hamming code is GF (2^3). In general, the extension field, over which the generator polynomial is defined, for $\left(2^m-1, 2^m-m-1\right)$Hamming code is GF($2^m$).

A finite field of polynomials is defined in terms of i) a finite set of polynomials as its elements and ii) a clearly defined set of rules for algebraic operations. For convenience, the algebraic operations are often named as 'addition', 'subtraction', 'multiplication' and 'inverse'. It is also mandatory that there will be one identity element for addition (which is termed as '0') and one identity element for 'multiplication' (which

is termed as '1') in the set of polynomials. Note that '0' and '1' are viewed as polynomials in 'x', which is an indeterminate. The field is said to be finite as the result of any allowed algebraic operations one two or more elements will be a polynomial belonging to the same finite set of polynomials.

Interestingly, if we can identify a polynomial in 'x', say p(x) defined over GF(2), such that p(x) is a prime and irreducible polynomial with degree 'm', then we can define a finite field with 2^m polynomial elements, i.e., GF(2^m) easily. Degree of a polynomial is the highest exponent of 'x' in the polynomial.

For the (7,4) Hamming code, m =3 and a desired prime, irreducible p(x) = g(x) = $x^3 + x + 1$. **Table 6.34.1** shows one possible representation of GF(2^3). Note that all polynomials whose degree are less than m = 3, are elements of GF(2^3) and it takes only a group of 'm' bits to represent one element in binary form.

Field Elements	Polynomials	Possible binary representation
0	0	000
$\alpha^0 = 1 = \alpha^7$	1	001
α	X	010
α^2	X^2	100
α^3	$x^3 = x + 1$	011
α^4	$x^4 = x.x^3 = x^2 + x$	110
α^5	$x^5 = x.x^4 = x^2 + x + 1$	111
α^6	$x^6 = x.x^5 = x^2 + 1$	101

Table 6.34.1: *An illustration on the elements of GF(2^3). Note that, $x^7 = 1$, $x^8 = x$ and so on*

Let us note that, for the (7,4) Hamming code, the degree of a message polynomial m(x) ≤ 3, the degree of the generator polynomial g(x) = 3 = (n – k) and the degree of a codeword polynomial c(x) ≤ 6.

As an example, let us consider a block of 4 message bits (0111) where '0' is the last bit in the sequence. So, the corresponding message polynomial is, $m(x) = x^2 + x + 1$.
If we follow non-systematic coding, the codeword polynomial is:
$c(x) = m(x).g(x) = (x^2 + x + 1)(x^3 + x + 1) = x^5 + x^3 + x^2 + x^4 + x^2 + x + x^3 + x + 1$
$= x^5 + x^4 + (x^3 \oplus x^3) + (x^2 \oplus x^2) + (x \oplus x) + 1 = x^5 + x^4 + 1$
So in binary form, we get the code word (0110001).
Similarly, if the message polynomial is $m(x) = x^3 + x^2 + x + 1$, verify that the codeword is (1101001).

If we consider a general form of the message polynomial, $m(x) = m_3 x^3 + m_2 x^2 + m_1 x + m_0$ where $m_i \in$ GF(2), a general form of the code word is

$$c(x) = m_3 x^6 + m_2 x^5 + (m_3 \oplus m_1) x^4 + (m_3 \oplus m_2 \oplus m_0) x^3 + (m_2 \oplus m_1) x^2 + (m_1 \oplus m_0) x + m_0$$

The above expression can used for direct implementation of the encoder with five two-input Exclusive-OR gates.

If we wish to go for systematic encoding where the parity bits will be appended to the message bits, we use the following expression for the codeword polynomial as introduced in Lesson #33:

$$c'(x) = x^{n-k}m(x) - R_{g(x)}\left[x^{n-k}m(x)\right] \qquad 6.34.3$$

Now, m(x) = $x^2 + x + 1$.

$\therefore\ x^{n-k}.m(x) = x^3\,(x^2 + x + 1) = x^5 + x^4 + x^3$

Now, divide this by the generator polynomial g(x) as bellow and remember that addition and subtraction operations are the same over GF(2).

x^3+x+1) $x^5+x^4+x^3$ (x^2+x

$x^5+\quad +x^3+x^2$

\-

$0+x^4+0.x^3+x^2$

$x^{4+}\qquad +x^2\ +x$

x

So, the remainder is x.

Therefore, the codeword in systematic form is:

$c'(x) = (x^5+x^4+x^3) + x$

$= (0\ 1\ 1\ 1\ 0\ 1\ 0)$

Message bits (0 1 1 1), Parity bits (0 1 0)

Hamming Decoder

Upon receiving a polynomial r(x) = c(x) + e(x), the decoder computes the syndrome polynomial s(x) $= R_{g(x)}[r(x)]$. If the remainder is '0', the decoder declares the received word as a correct one. If s(x) is a non-zero polynomial, the decoder assumes that there is a single error in one of the 7 bits. In fact, the precise location of the erroneous bit is uniquely related to the syndrome polynomial. Note that the degree of the syndrome polynomial is 2 or less. So, the number of non-zero syndrome polynomials is $(2^3-1) = 7$. Each polynomial form indicates a single error at a bit position. Once the location of the erroneous bit is found from the syndrome polynomial, the bit is simply inverted to obtain the correct transmitted codeword. For a systematic code, the information bits can be retrieved easily from the corrected

codeword. For a nonsystematic code, another division operation is necessary. However, for practical implementation, one can use a Look Up Table (LUT) for the purpose.

Reed-Solomon Codes

Reed Solomon (R-S) codes form an important sub-class of the family of Bose-Chaudhuri-Hocquenghem (BCH) codes and are very powerful linear non-binary block codes capable of correcting multiple random as well as burst errors. They have an important feature that the generator polynomial and the code symbols are derived from the same finite field. This enables to reduce the complexity and also the number of computations involved in their implementation. A large number of R-S codes are available with different code rates. A few R-S codes with their code parameters are shown in **Table 6.34.2.**

Code No.	**Code Name**	**Block Length (n)**	**No.of information symbols in a code word (k)**	**Error correcting power (t)**	**Code Rate (R)**	**Field of Definition**
1	(31, 27) R – S	31	27	2	0.871	
2	(31, 21) R – S	31	21	5	0.677	
3	(31, 15) R – S	31	15	8	0.484	$GF(2^5)$
4	(31, 11) R – S	31	11	10	0.355	
5	(63, 55) R – S	63	55	4	0.873	
6	(63, 47) R – S	63	47	8	0.746	
7	(63, 39) R – S	63	39	12	0.619	$GF(2^6)$
8	(63, 31) R – S	63	31	16	0.492	
9	(63, 23) R – S	63	23	20	0.365	
10	(63, 15) R – S	63	15	24	0.238	
11	(255, 233) R – S	255	233	11	0.913	
12	(255, 225) R – S	255	225	15	0.882	
13	(255, 205) R – S	255	205	25	0.804	
14	(255, 191) R – S	255	191	32	0.749	$GF(2^8)$
15	(255, 183) R – S	255	183	36	0.718	
16	(255, 175) R – S	255	175	40	0.686	
17	(255, 165) R – S	255	165	45	0.647	
18	(255, 135) R – S	255	135	60	0.529	

Table 6.34.2 *Parameters of a few selected Reed Solomon codes*

Like other forward error correcting (FEC) decoders the decoding of the R-S codes is more complex than their encoding operation. An iterative algorithm due to E.R Berlekamp with J. L. Massey's extension, a search algorithm due to R. T. Chien and G. D. Forney's method for calculation for error values, together constitute a basic approach for decoding the R-S codes.

An R-S code is described by a generator polynomial g(x) and other usual important code parameters such as the number of message symbols per block (k), number of code symbols per block (n), maximum number of erroneous symbols (t) that can surely be corrected per block of received symbols and the designed minimum symbol Hamming distance (d). A parity-check polynomial h (X) of order k also plays a role in designing the code. The symbol x, used in polynomials is an indeterminate which usually implies unit amount of delay. Even though the R-S codes are non-binary in construction and defined over GF(2^m), the symbols can be expressed and processed in groups of 'm' bits.

For positive integers m and t, a primitive (n, k, t) R-S code is defined as below:
Number of encoded symbols per block: $n = 2^m - 1$
Number of message symbols per block: k
Code rate: R= k/n
Number of parity symbols per block: $n - k = 2t$
Minimum symbol Hamming distance per block: $d = 2t + 1$.

It can be noted that the block length n of an (n, k, t) R-S code is bounded by the corresponding finite field GF(2^m). Moreover, as $n - k = 2t$, an (n, k, t) R-S code has optimum error correcting capability. As we have seen earlier in this lesson, each field elements can be represented by m bits. So, a codeword of (n,k,t) R-S code over GF (2^m), consisting of n symbols, can be represented by $(n \times m)$ bits and the code has sufficient inherent algebraic structure to correct at least t erroneous symbols per block. If the number of erroneous symbols per block exceeds t, there is no guarantee that the errors will be corrected. Practical decoders in such cases behave differently depending on the error pattern. Possibility of correcting more than t erroneous symbols, for some error pattern, also exists. The t erroneous symbols, when represented by m bits each, can be affected in two extreme cases in the following manner:

(i) one bit of each of the t-symbols is corrupted.
(ii) all m bits of each of the t-symbols are corrupted.

So, the random error correcting capability of an (n,k,t) R-S code is up to at least t bits per block of $(n \times m)$ bits. When errors due to a communication channel are not totally uncorrelated, i.e. the errors tend to be bursty in nature, the use of R-S code becomes more relevant and important. Note that a particular symbol, consisting of m bits, may be erroneous due to error in one or more (up to m) bits and the code ensures correct decoding of up to t such erroneous symbols. Thus if there are $(m \times t)$ or less erroneous bits, distributed in t symbols only, the code corrects the erroneous symbols without failure.

The generator polynomial g(X) for the (n, k, t) R-S code is written as,

$$g(x) = \prod_{i=1}^{2t} \left(x - \alpha^i\right) \qquad 6.34.4$$

Here 'α' is the primitive root of an irreducible polynomial of degree 'm' over GF(2) and 'α' is defined over GF(2^m),the extension field. We have briefly discussed about the definition of GF(2^m) earlier in this lesson.

Let us consider an example of an R-S code:

Example #6.34.1: Let the allowable bandwidth expansion for error correcting code alone be about 50% and let the hardware complexity permit us to consider block length of 31.

For such a situation one can think of an R-S code with following parameters:

$N = 31 = 2^5 - 1$; $m = 5$;

Based on the allowed bandwidth expansion, the approximate code rate is 2/3.

So, we can choose $k = 21$; $R = 21/31 = 0.678$.

$n-k = 2t = 10$; $t = 5$ and $d = 2t + 1 = 11$.

Thus the chosen code is a (31, 21, 5) R-S code.

The generator polynomial g(x) of this code is given below:

$$g(X) = \prod_{i=1}^{10}\left(X + a^i\right)$$

$$= X^{10} + a^{18}X^9 + X^8 + a^{25}X^7 + a^{10}X^6 + a^9X^5 + a^{12}X^4 + a^{16}X^3 + a^2X^2 + X + a^{24} \qquad 6.34.5$$

Here, 'α' is the primitive root of the irreducible polynomial $p(x) = x^5 + x^2 + 1$ of degree 5 and defined over GF (2)

■

A general block diagram of a digital communication system employing Reed-Solomon forward error correcting codec is shown in **Fig. 6.34.1**. Serial message bits at the output of the source encoder are de-multiplexed suitably and groups of such m bits are fed to the R-S encoder at a time. Let the polynomial m (x) of degree (k – 1) or less denote the message polynomial and the polynomial $rem(x)$ of degree (n – k – 1) or less denote the remainder parity polynomial.

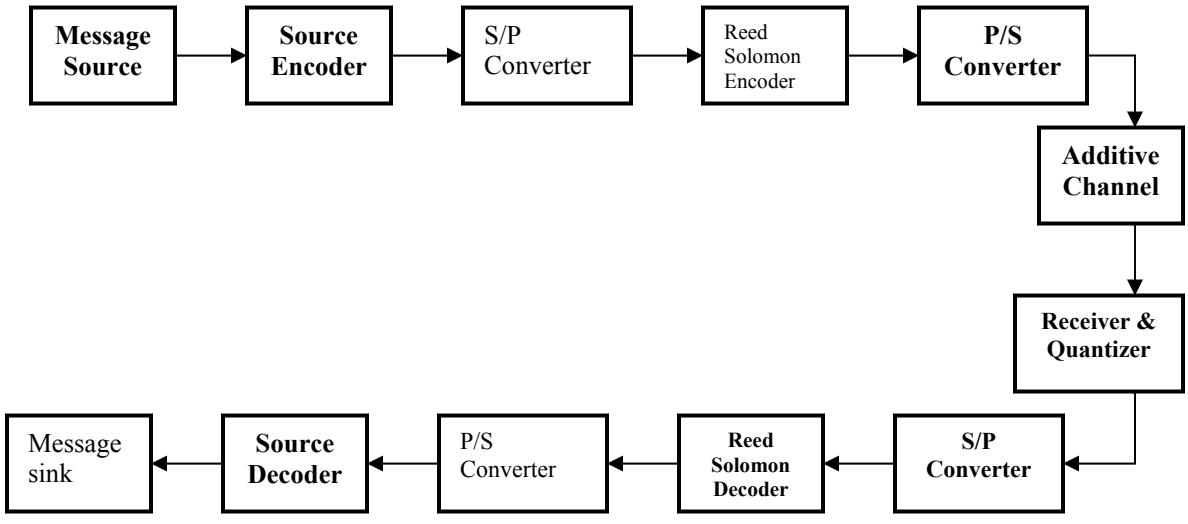


Fig 6.34.1 *Block diagram of a digital communication system using Reed-Solomon codes for error control*

We can write the encoded codeword polynomial v (x) of degree (n – 1) or less in the following form:

$$v(x) = \sum_{i=0}^{n-k} v_i x^i = x^{n-k}.m(x) + rem(x) = q(x).g(x) \qquad 6.34.6$$

Here q (X) is a polynomial of degree (k – 1) or less. The coefficients of all the polynomials are elements of GF(2^m). The parity-check polynomial $rem(x)$ can be found as the remainder polynomial when the message polynomial, shifted by (n – k) times, is divided by the generator polynomial g(x). **Fig.6.34.2** shows a block diagram of a systematic Reed-Solomon encoder employing (n – k) stage shift registers. Each stage consists of 'm' set of delay units. The encoder outputs m bits at a time and this set of m bits are multiplexed suitably depending on the front-end modulation technique employed.

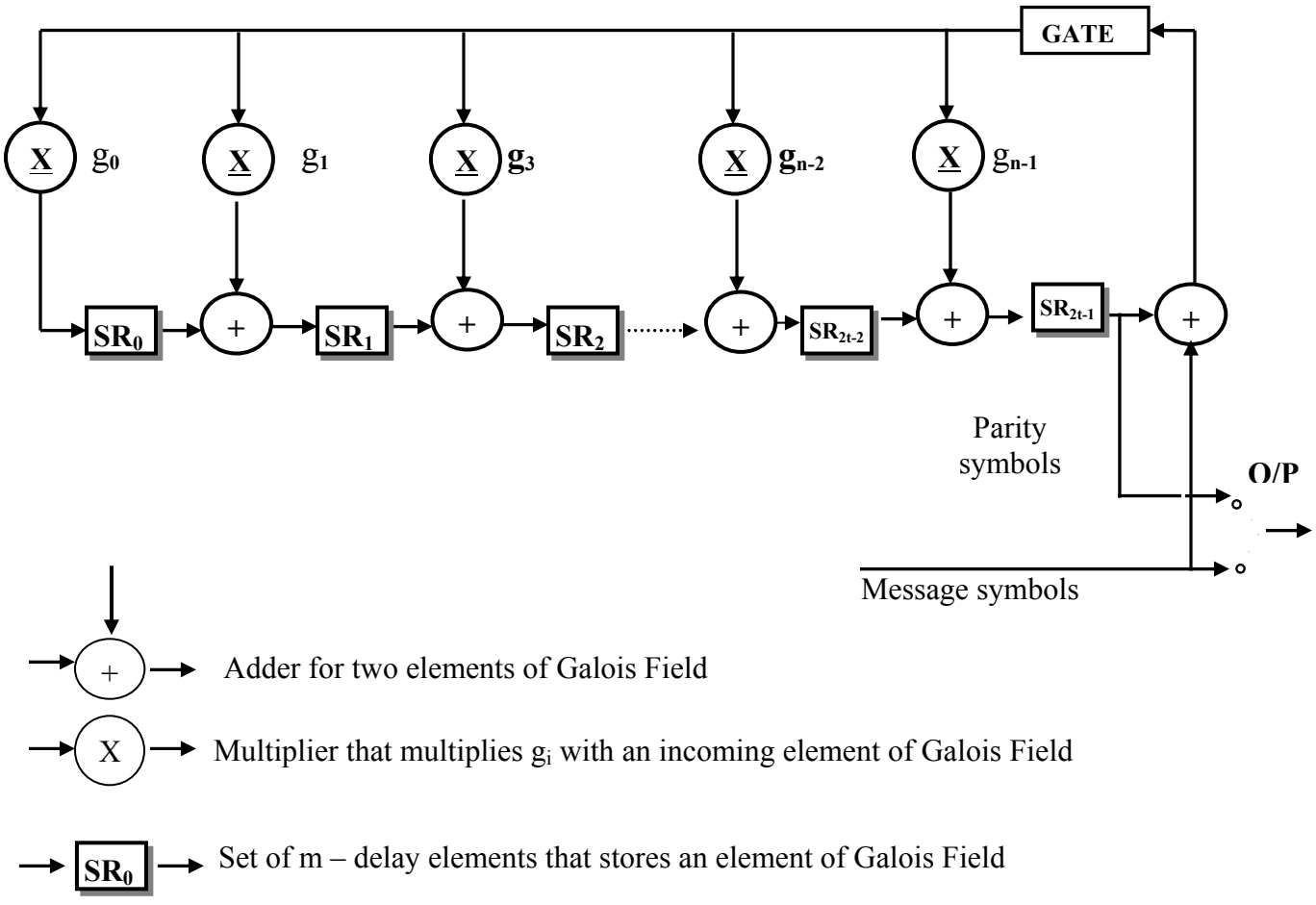


Fig 6.34.2 *Reed-Solomon encoder employing (n – k) shift register stages*

At the input-end of the R-S decoder in the receiver path in **Fig. 6.34.1**, hard-quantization of received signal has been assumed. Now, due to noisy transmission channel or receiver imperfection or constraints in other system parameters (e.g. transmitter power limitation, intentional degradation of difficult-to-maintain system components etc.) the code symbols received by the R-S decoder may be erroneous. The task of the decoder is to recover correct set of message symbols efficiently from these received symbols. A logical approach towards this is outlined in the form of a flow chart shown in **Fig. 6.34.3**. The received code symbols are represented sequentially by the coefficients of the received code polynomial f (x). This polynomial can be considered as the sum of the transmitted code polynomial v(x) and the error polynomial e(x). Since v(x) is a valid code polynomial, it is divisible by the generator polynomial g(x).

Moreover, as $g(x) = \prod_{i=1}^{2t}\left(x - \alpha^i\right)$ it can be easily seen that, $v(a^j) = 0$, for $1 \le j \le 2t$

so, we can write, for$1 \le j \le 2t$,

$$f(a^j) = \sum_{i=0}^{n-1} f_t (a^j)^i \; = v(a^j) + e(a^j) = 0 + e(a^j)$$

This shows that the quantities $f\left(\alpha^j\right)$ contain information about the error polynomial e(x) only. These quantities, named syndromes, are only 2t in number and they play pivotal role in the decoding process.

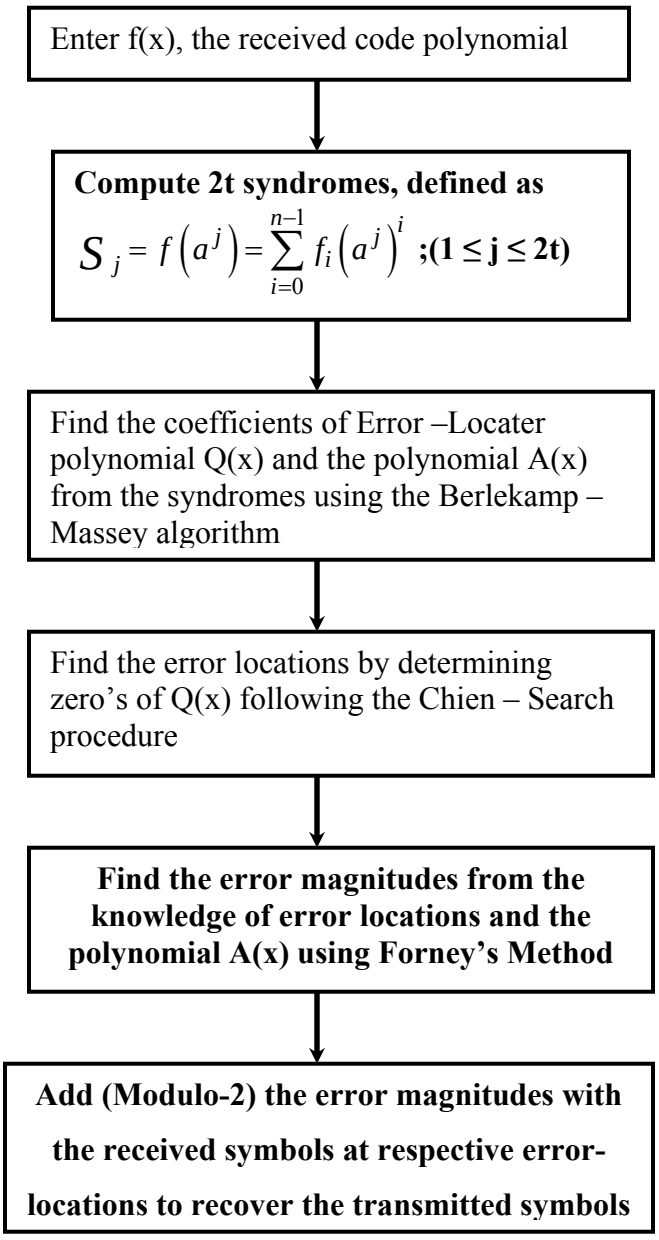


Fig 6.34.3 *Major steps in the decoding of the Reed-Solomon codes*

After determining the syndromes, the next step is to find the coefficients of a suitably defined error-locator polynomial Q(x). This is carried out following the Berlekamp-Massey algorithm. The coefficients of the error-locator polynomial are necessary for finding the error-locations which is done following the cyclic Chien search

procedure in the third step. The next task of finding the error magnitudes is performed efficiently using Forney's method. The remaining task is to subtract the error polynomial from the received polynomial and then extract the message polynomial from the corrected codeword.

Module 6

Channel Coding

Lesson 35

Convolutional Codes

After reading this lesson, you will learn about

- ***Basic concepts of Convolutional Codes;***
- ***State Diagram Representation;***
- ***Tree Diagram Representation;***
- ***Trellis Diagram Representation;***
- ***Catastrophic Convolutional Code;***
- ***Hard - Decision Viterbi Algorithm;***
- ***Soft - Decision Viterbi Algorithm;***

Convolutional codes are commonly described using two parameters: the code rate and the constraint length. The code rate, k/n, is expressed as a ratio of the number of bits into the convolutional encoder (k) to the number of channel symbols output by the convolutional encoder (n) in a given encoder cycle. The constraint length parameter, K, denotes the "length" of the convolutional encoder, i.e. how many k-bit stages are available to feed the combinatorial logic that produces the output symbols. Closely related to K is the parameter m, which indicates how many encoder cycles an input bit is retained and used for encoding after it first appears at the input to the convolutional encoder. The m parameter can be thought of as the memory length of the encoder.

Convolutional codes are widely used as channel codes in practical communication systems for error correction. The encoded bits depend on the current k input bits and a few past input bits. The main decoding strategy for convolutional codes is based on the widely used Viterbi algorithm. As a result of the wide acceptance of convolutional codes, there have been several approaches to modify and extend this basic coding scheme. Trellis coded modulation (TCM) and turbo codes are two such examples. In TCM, redundancy is added by combining coding and modulation into a single operation. This is achieved without any reduction in data rate or expansion in bandwidth as required by only error correcting coding schemes.

A simple convolutional encoder is shown in **Fig. 6.35.1**. The information bits are fed in small groups of k-bits at a time to a shift register. The output encoded bits are obtained by modulo-2 addition (EXCLUSIVE-OR operation) of the input information bits and the contents of the shift registers which are a few previous information bits.

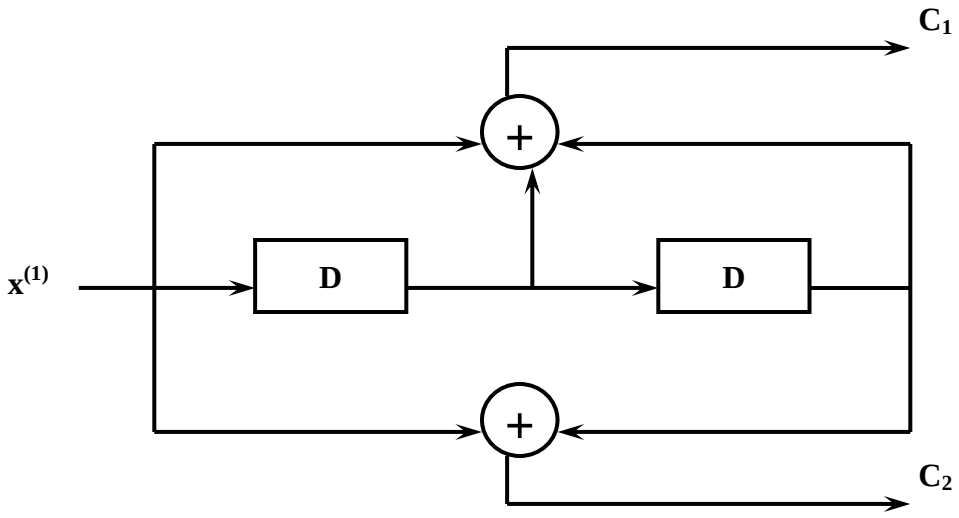


Fig. 6.35.1 *A convolutional encoder with k=1, n=2 and r=1/2*

If the encoder generates a group of 'n' encoded bits per group of 'k' information bits, the code rate R is commonly defined as R = k/n. In **Fig. 6.35.1**, k = 1 and n = 2. The number, K of elements in the shift register which decides for how many codewords one information bit will affect the encoder output, is known as the constraint length of the code. For the present example, K = 3.

The shift register of the encoder is initialized to all-zero-state before encoding operation starts. It is easy to verify that encoded sequence is 00 11 10 00 01 ….for an input message sequence of 01011….

The operation of a convolutional encoder can be explained in several but equivalent ways such as, by a) state diagram representation, b) tree diagram representation and c) trellis diagram representation.

a) State Diagram Representation

A convolutional encoder may be defined as a finite state machine. Contents of the rightmost (K-1) shift register stages define the states of the encoder. So, the encoder in **Fig. 6.35.1** has four states. The transition of an encoder from one state to another, as caused by input bits, is depicted in the state diagram. **Fig. 6.35.2** shows the state diagram of the encoder in **Fig. 6.35.1**. A new input bit causes a transition from one state to another. The path information between the states, denoted as b/c_1c_2, represents input information bit 'b' and the corresponding output bits (c_1c_2). Again, it is not difficult to verify from the state diagram that an input information sequence **b** = (1011) generates an encoded sequence **c** = (11, 10, 00, 01).

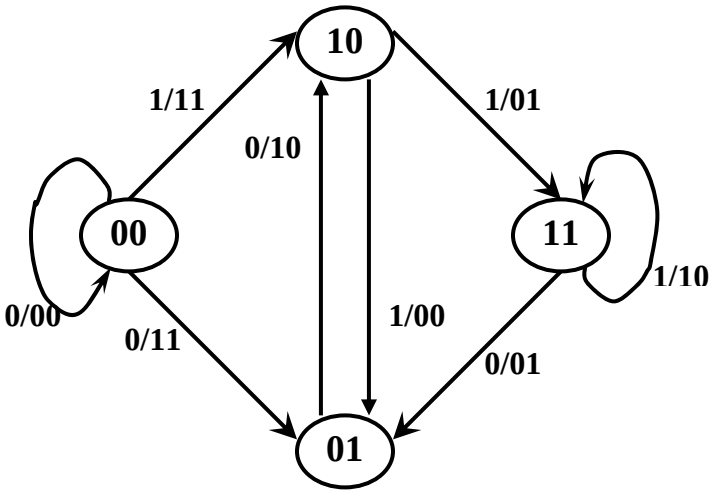


Fig.6.35.2 *State diagram representation for the encoder in Fig. 6.35.1*

b) Tree Diagram Representation

The tree diagram representation shows all possible information and encoded sequences for the convolutional encoder. **Fig. 6.35.3** shows the tree diagram for the encoder in **Fig. 6.35.1**. The encoded bits are labeled on the branches of the tree. Given an input sequence, the encoded sequence can be directly read from the tree. As an example, an input sequence (1011) results in the encoded sequence (11, 10, 00, 01).

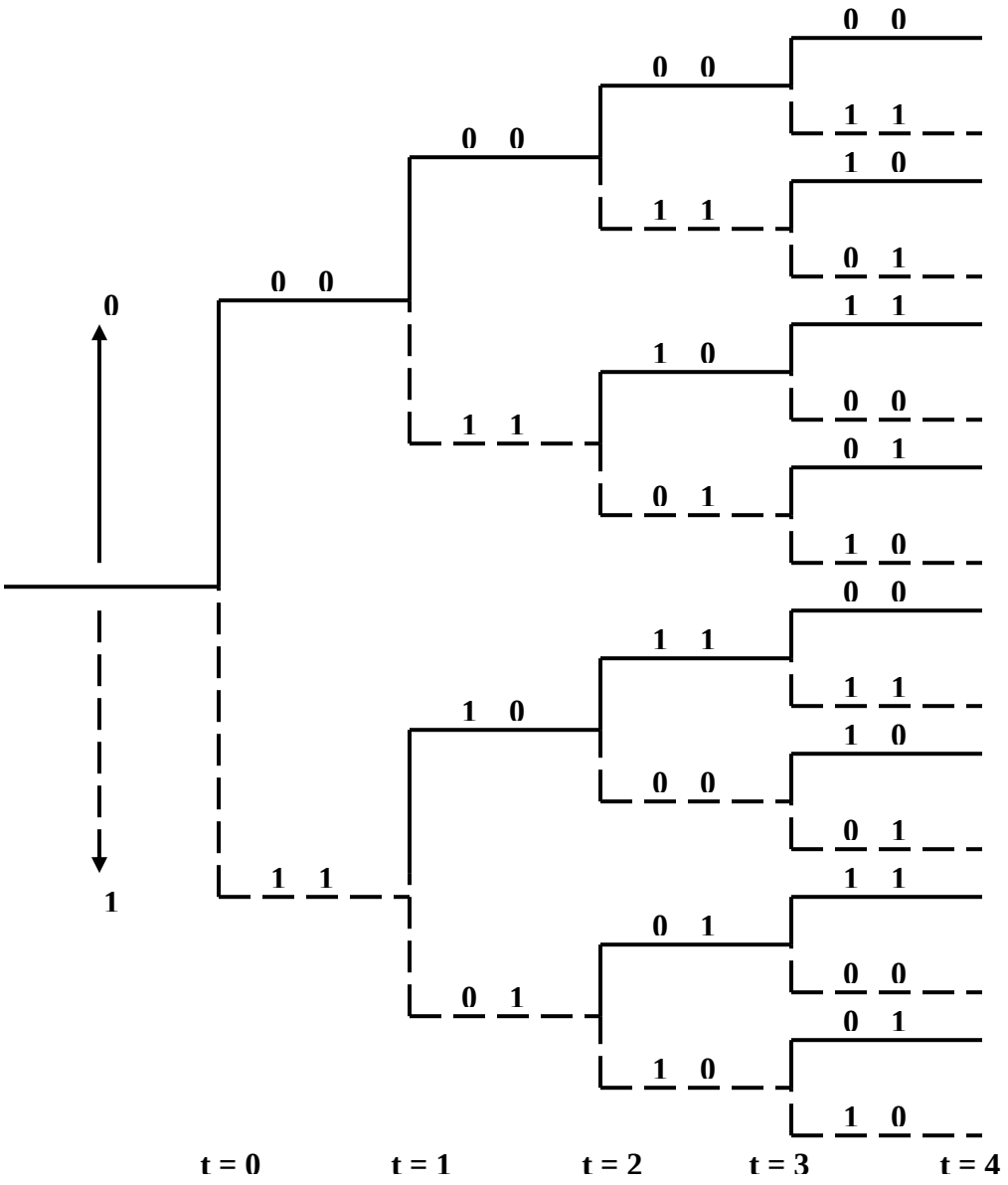


Fig. 6.35.3 *A tree diagram for the encoder in Fig. 6.35.1*

c) Trellis Diagram Representation

The trellis diagram of a convolutional code is obtained from its state diagram. All state transitions at each time step are explicitly shown in the diagram to retain the time dimension, as is present in the corresponding tree diagram. Usually, supporting descriptions on state transitions, corresponding input and output bits etc. are labeled in the trellis diagram. It is interesting to note that the trellis diagram, which describes the operation of the encoder, is very convenient for describing the behavior of the

corresponding decoder, especially when the famous 'Viterbi Algorithm (VA)' is followed. **Figure 6.35.4** shows the trellis diagram for the encoder in **Figure 6.35.1**.

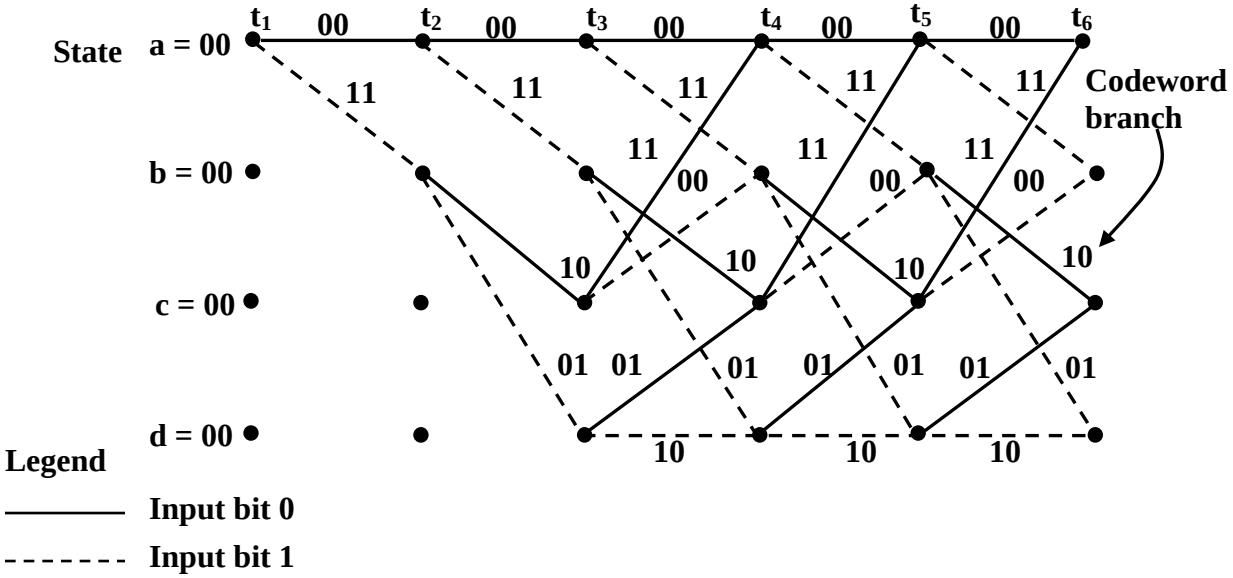


Fig.6.35.4(a) *Trellis diagram for the encoder in Fig. 6.35.1*

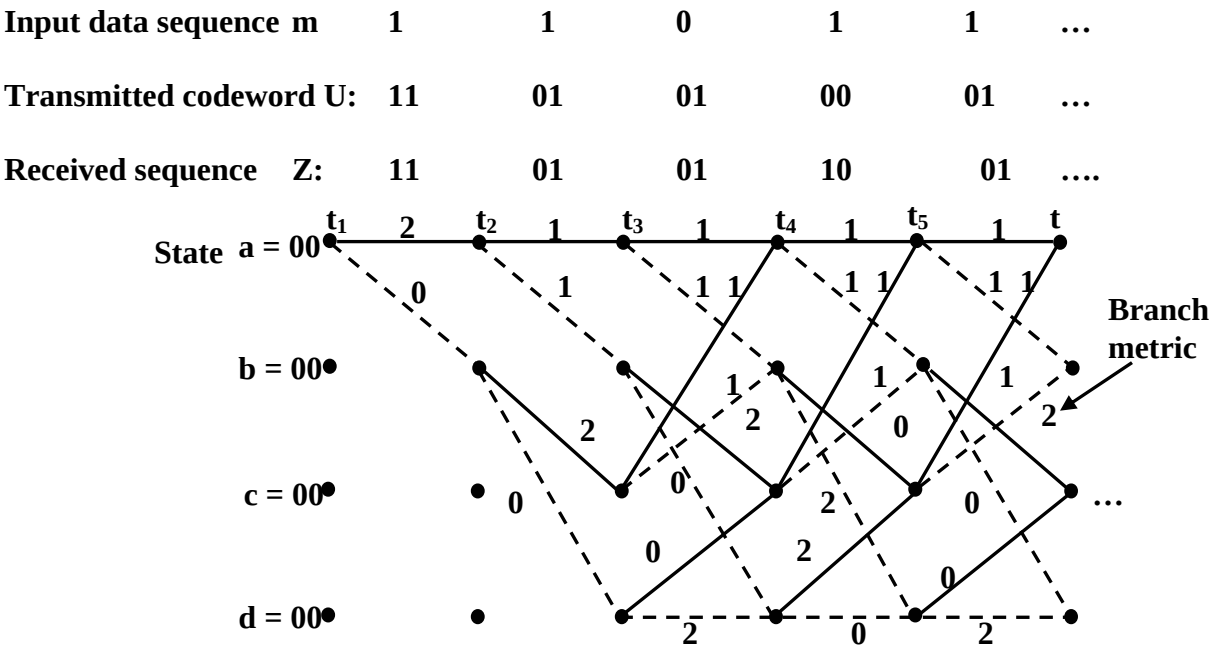


Fig.6.35.4(b) *Trellis diagram, used in the decoder corresponding to the encoder in Fig. 6.35.1*

Catastrophic Convolutional Code

The taps of shift registers used for a convolutional encoder are to be chosen carefully so that the code can effectively correct errors in received data stream. One measure of error correction capability of a convolutional code is its 'minimum free distance, d_{free}', which indicates the minimum weight (counted in number of '1'-s) of a path that branches out from the all-zero path of the code trellis and again merges with the all-zero path. For example, the code in **Fig.6.35.1** has d_{free}= 5. Most of the convolutional codes of present interest are good for correcting random errors rather than correcting error bursts. If we assume that sufficient number of bits in a received bit sequence are error free and then a few bits are erroneous randomly, the decoder is likely to correct these errors. It is expected that (following a hard decision decoding approach, explained later) the decoder will correct up to $(d_{free}-1)/2$ errors in case of such events. So, the taps of a convolutional code should be chosen to maximize d_{free}. There is no unique method of finding such convolutional codes of arbitrary rate and constraint length that ensures maximum d_{free}. However, comprehensive description of taps for 'good ' convolutional codes of practical interest have been prepared through extensive computer search techniques and otherwise.

While choosing a convolutional code, one should also avoid 'catastrophic convolutional code'. Such codes can be identified by the state diagram. The state diagram of a 'catastrophic convolutional code' includes at least one loop in which a nonzero information sequence corresponds to an all-zero output sequence. Tree diagram of a 'catastrophic convolutional code' is shown in **Fig.6.35.5**.

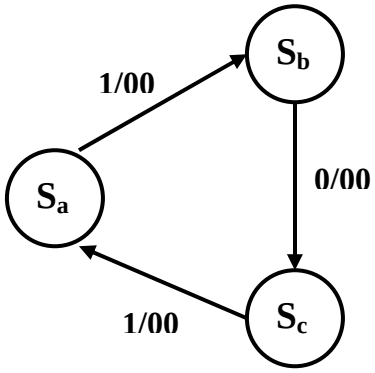


Fig.6.35.5 *Example of a catastrophic code*

Hard-Decision and Soft-Decision Decoding

Hard-decision and soft-decision decoding are based on the type of quantization used on the received bits. Hard-decision decoding uses 1-bit quantization on the received samples. Soft-decision decoding uses multi-bit quantization (e.g. 3 bits/sample) on the received sample values.

Hard-Decision Viterbi Algorithm

The Viterbi Algorithm (VA) finds a maximum likelihood (ML) estimate of a transmitted code sequence **c** from the corresponding received sequence **r** by maximizing the probability p(**r**|**c**) that sequence **r** is received conditioned on the estimated code sequence **c.** Sequence **c** must be a valid coded sequence.

The Viterbi algorithm utilizes the trellis diagram to compute the path metrics. The channel is assumed to be memory less, i.e. the noise sample affecting a received bit is independent from the noise sample affecting the other bits. The decoding operation starts from state '00', i.e. with the assumption that the initial state of the encoder is '00'. With receipt of one noisy codeword, the decoding operation progresses by one step deeper into the trellis diagram. The branches, associated with a state of the trellis tell us about the corresponding codewords that the encoder may generate starting from this state. Hence, upon receipt of a codeword, it is possible to note the 'branch metric' of each branch by determining the Hamming distance of the received codeword from the valid codeword associated with that branch. Path metric of all branches, associated with all the states are calculated similarly.

Now, at each depth of the trellis, each state also carries some 'accumulated path metric', which is the addition of metrics of all branches that construct the 'most likely path' to that state. As an example, the trellis diagram of the code shown in **Fig. 6.35.1**, has four states and each state has two incoming and two outgoing branches. At any depth of the trellis, each state can be reached through two paths from the previous stage and as per the VA, the path with lower accumulated path metric is chosen. In the process, the 'accumulated path metric' is updated by adding the metric of the incoming branch with the 'accumulated path metric' of the state from where the branch originated. No decision about a received codeword is taken from such operations and the decoding decision is deliberately delayed to reduce the possibility of erroneous decision.

The basic operations which are carried out as per the hard-decision Viterbi Algorithm after receiving one codeword are summarized below:

a) All the branch metrics of all the states are determined;

b) Accumulated metrics of all the paths (two in our example code) leading to a state are calculated taking into consideration the 'accumulated path metrics' of the states from where the most recent branches emerged;

c) Only one of the paths, entering into a state, which has minimum 'accumulated path metric' is chosen as the 'survivor path' for the state (or, equivalently 'node');

d) So, at the end of this process, each state has one 'survivor path'. The 'history' of a survivor path is also maintained by the node appropriately (e.g. by storing the codewords or the information bits which are associated with the branches making the path);

e) Steps a) to d) are repeated and decoding decision is delayed till sufficient number of codewords has been received. Typically, the delay in decision making = Lx k codewords where L is an integer, e.g. 5 or 6. For the code in **Fig. 6.35.1**, the decision delay of 5x3 = 15 codewords may be sufficient for most occasions. This means, we decide about the first received codeword after receiving the 16^{th} codeword. The decision strategy is simple. Upon receiving the 16^{th} codeword and carrying out steps a) to d), we compare the 'accumulated path metrics' of all the states (four in our example) and chose the state with minimum overall 'accumulated path metric' as the 'winning node' for the first codeword. Then we trace back the history of the path associated with this winning node to identify the codeword tagged to the first branch of the path and declare this codeword as the most likely transmitted first codeword.

The above procedure is repeated for each received codeword hereafter. Thus, the decision for a codeword is delayed but once the decision process starts, we decide once for every received codeword. For most practical applications, including delay-sensitive digital speech coding and transmission, a decision delay of Lx k codewords is acceptable.

Soft-Decision Viterbi Algorithm

In soft-decision decoding, the demodulator does not assign a '0' or a '1' to each received bit but uses multi-bit quantized values. The soft-decision Viterbi algorithm is very similar to its hard-decision algorithm except that squared Euclidean distance is used in the branch metrics instead of simpler Hamming distance. However, the performance of a soft-decision VA is much more impressive compared to its HDD (Hard Decision Decoding) counterpart [**Fig. 6.35.6 (a) and (b)**]. The computational requirement of a Viterbi decoder grows exponentially as a function of the constraint length and hence it is usually limited in practice to constraint lengths of K = 9.

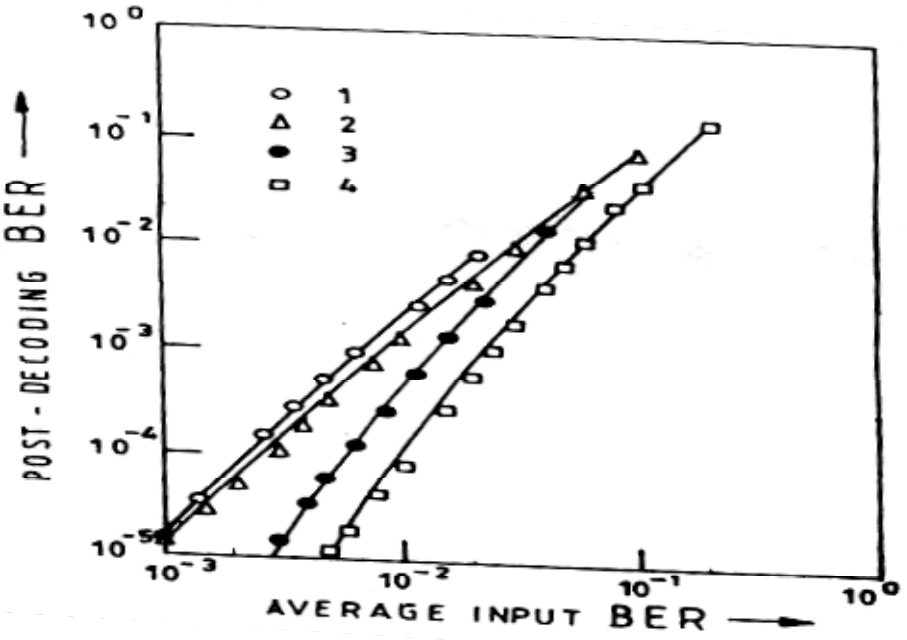


Fig. 6.35.6 (a) *Decoded BER vs input BER for the rate – half convolutional codes with Viterbi Algorithm ; 1) k = 3 (HDD),2) k = 5 (HDD),3) k = 3 (SDD), and 4) k= 5 (SDD). HDD: Hard Decision Decoding; SDD: Soft Decision Decoding.*

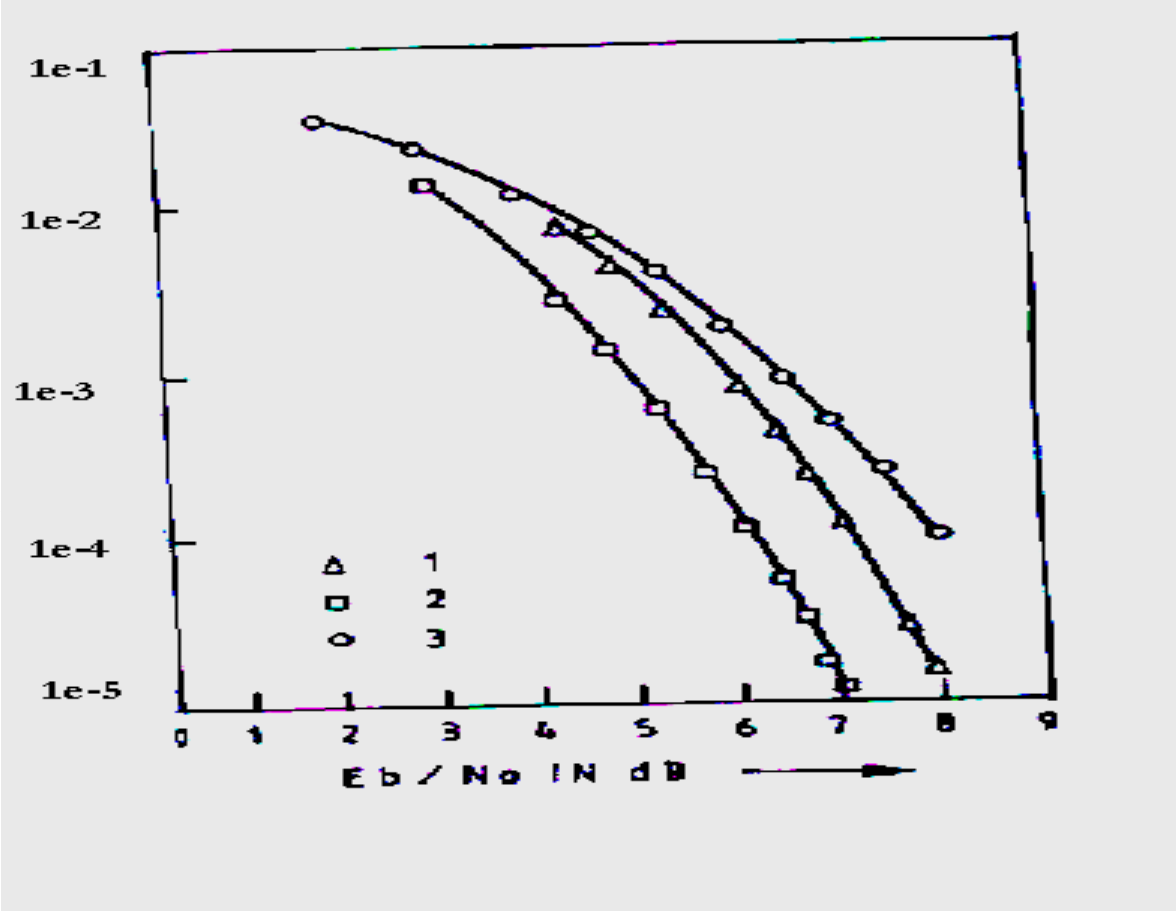


Fig. 6.35.6 (b) *Decoded BER vs E_b/N_o (in dB) for the rate – half convolutional codes with Viterbi Algorithm ; 1) Uncoded system; 2) with k = 3 (HDD) and 3) k = 3 (SDD). HDD: Hard Decision Decoding; SDD: Soft Decision Decoding.*

Module 6

Channel Coding

Lesson 36

Coded Modulation Schemes

After reading this lesson, you will learn about

- ***Trellis Code Modulation;***
- ***Set partitioning in TCM;***
- ***Decoding TCM;***

The modulated waveform in a conventional uncoded carrier modulation scheme contains all information about the modulating message signal. As we have discussed earlier in **Module #5**, the modulating signal uses the quadrature carrier sinusoids in PSK and QAM modulations. The modulator accepts the message signal in discrete time discrete amplitude from (binary or multilevel) and processes it following the chosen modulation scheme to satisfy the transmission and reception requirements. The modulating signal is generally treated as a random signal and the modulating symbols are viewed as statistically independent and having equal probability of occurrence. This traditional approach has several merits such as, (i) simplified system design and analysis because of modular approach, (ii) design of modulation and transmission schemes which are independent of the behavior of signal source, (iii) availability of well developed theory and practice for the design of receivers which are optimum (or near optimum).

To put it simply, the modular approach to system design allows one to design a modulation scheme almost independent of the preceding error control encoder and the design of an encoder largely independent of the modulation format. Hence, the end-to-end system performance is made up of the gains contributed by the encoder, modulator and other modules separately.

However, it is interesting to note that the demarcation between a coder and a modulator is indeed artificial and one may very well imagine a combined coded modulation scheme. The traditional approach is biased more towards optimization of an encoder independent of the modulation format and on optimization of a modulation scheme independent of the coding strategy. In fact, such approach of optimization at the subsystem level may not ensure an end-to-end optimized system design.

A systems approach towards a combined coding and modulation scheme is meaningful in order to obtain better system performance. Significant progress has taken place over the last two decades on the concepts of combined coding and modulation (also referred as coded modulation schemes). Several schemes have been suggested in the literature and advanced modems have been successfully developed exploiting the new concepts. Amongst the several strategies, which have been popular, trellis coded modulation (TCM) is a prominent one. A trellis coded modulation scheme improves the reliability of a digital transmission system without bandwidth expansion or reduction of data rate when compared to an uncoded transmission scheme using the same transmission bandwidth.

We discuss the basic features of TCM in this section after introducing some concepts of distance measure etc., common to all coded modulation schemes. A TCM scheme uses the concept of tree or trellis coding and hence is the name ‘TCM’. However

some interesting combined coding and modulation schemes have been suggested recently following the concepts of block coding as well.

Distance Measure

Let us consider a 2-dimensional signal space of **Fig.6.36.1** showing a set of signal points. The two dimensions are defined by two orthonormal basis functions corresponding to information symbols, which are to be modulated. The dimensions and hence the x- and y-axes may also be interpreted as 'real' and 'imaginary' axes when complex low pass equivalent representation of narrowband modulated signals is considered. The points in the constellation are distinguishable from one another as their locations are separate. Several possible subsets have been indicated in **Fig.6.36.1** capturing multiple modulation formats.

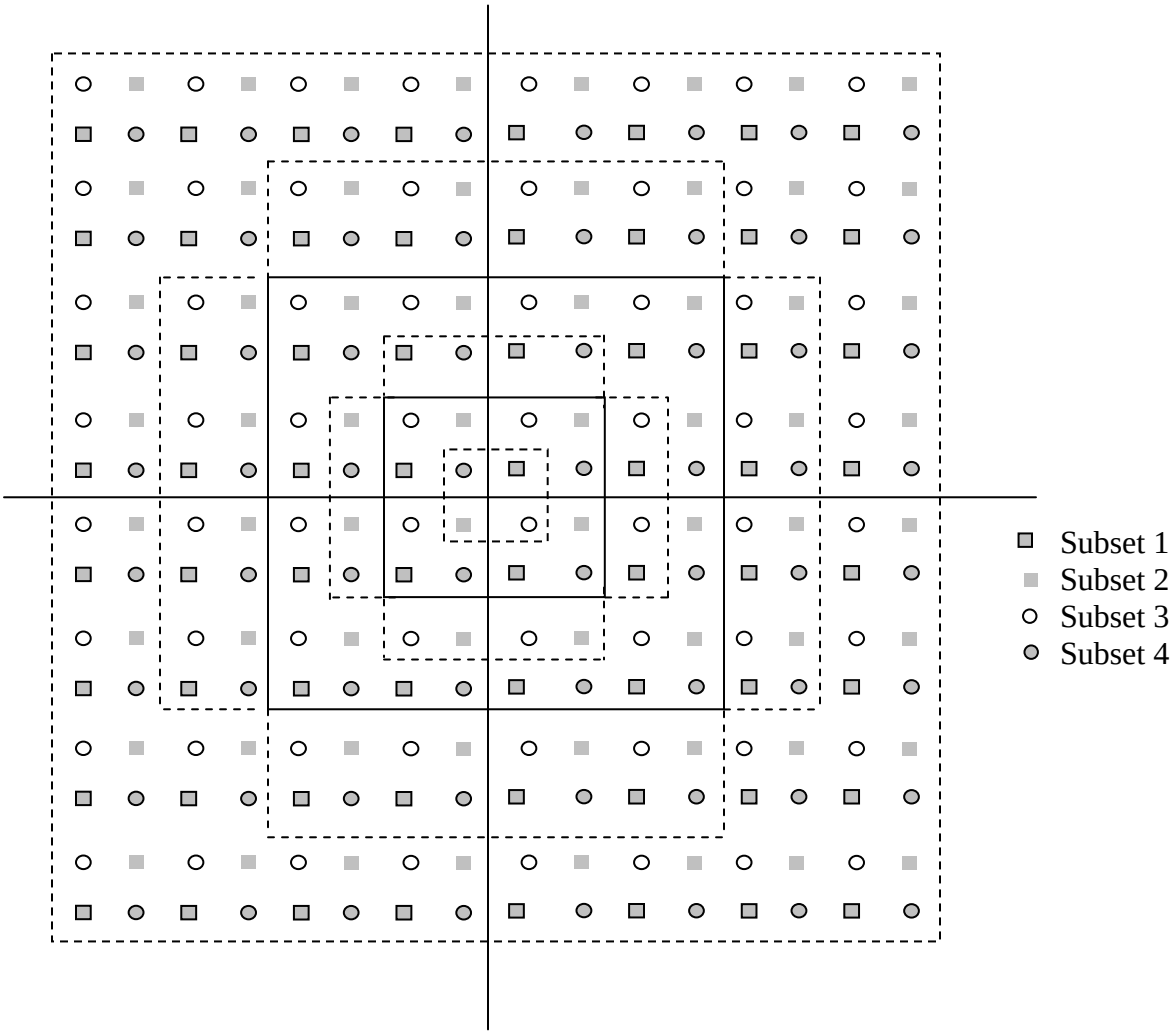


Fig.6.36.1 *Some two-dimensional signal constellations ($M = 2^m$, m= 1,…,8) of TCM codes.*

Let there be 'N' valid signal points, denoted by $(x_i + jy_i)$, $1 \leq i \leq N$, in the two dimensional signal space such that each point signifies an information bearing symbol (or signal) different from one another. Now, the Euclidean distance d_{ij} between any two signal points, say (x_i, y_i) and (x_j, y_j) in this 2 –D Cartesian signal space may be expressed as, $d^2ij = (x_i-x_j)^2 + (y_i-y_j)^2$. Note that for detecting a received symbol at the demodulator in presence of noise etc., it is important to maximize the minimum distance (say, d_{min}) between any two adjacent signal points in the space. This implies that the N signal points should be well distributed in the signal space such that the d_{min} amongst them is the largest. This strategy ensures good performance of a demodulation scheme especially when all the symbols are equally likely to occur.

Suppose we wish to transmit data from a source emitting two information bits every T seconds. One can design a system in several ways to accomplish the task such as the following:
(i) use uncoded QPSK modulation, with one signal carrying two information bits transmitted every T seconds.

(ii) use a convolutional code of rate r = 2/3 and same QPSK modulation. Each QPSK symbol now carries 4/3 information bits and hence, the symbol duration should be reduced to 2T/3 seconds. This implies that the required transmission bandwidth is 50% more compared to the scheme in (i).

(i) use a convolutional code of rate r = 2/3 and 8-Phase Shift Keying (8PSK) modulation scheme to ensure a symbol duration of T sec. Each symbol, now consisting of 3 bits, carries two bits of information and no expansion in transmission bandwidth is necessary. This is the basic concept of TCM.

Now, an M-ary PSK modulation scheme is known to be more and more power inefficient for larger values of M. That is, to ensure an average BER of, say, 10^{-5}, 8-PSK-modulation scheme needs more E_b/N_o compared to QPSK and 16-PSK scheme needs even more of E_b/N_o. So, one may apprehend that the scheme in (iii) may be power inefficient but actually this is not true as the associated convolutional code ensures a considerable improvement in the symbol detection process. It has been found that an impressive overall coding gain to the tune of 3 – 6dB may be achieved at an average BER of 10^{-5}. So, the net result of this approach of combined coding and modulation is some coding gain at no extra bandwidth. The complexity of such a scheme is comparable to that of a scheme employing coding (with soft decision decoding) and demodulation schemes separately.

TCM is extensively used in high bit rate modems using telephone cables. The additional coding gain due to trellis-coded modulation has made it possible to increase the speed of the transmission.

Set Partitioning

The central feature of TCM is based on the concept of signal-set partitioning that creates scope of redundancy for coding in the signal space. The minimum Euclidean distance (d_{min}) of a TCM scheme is maximized through set partitioning.

The concept of set partitioning is shown in **Fig. 6.36.2** for a 16-QAM signal constellation. The constellation consists of 16 signal points where each point is represented by four information bits. The signal set is successively divided into smaller sets with higher values of minimum intra-set distance. The smallest signal constellations finally obtained are labeled as D0, D1, … , D7 in ***Fig. 6.36.2.*** The following basic rules are followed for set-partitioning:

Rule #1: Members of the same partition are assigned to parallel transitions.

Rule #2: Members of the next larger partition are assigned to adjacent transitions, i.e. transitions stemming from, or merging in the same node.

Assumption: All the signal points are equally likely to occur.

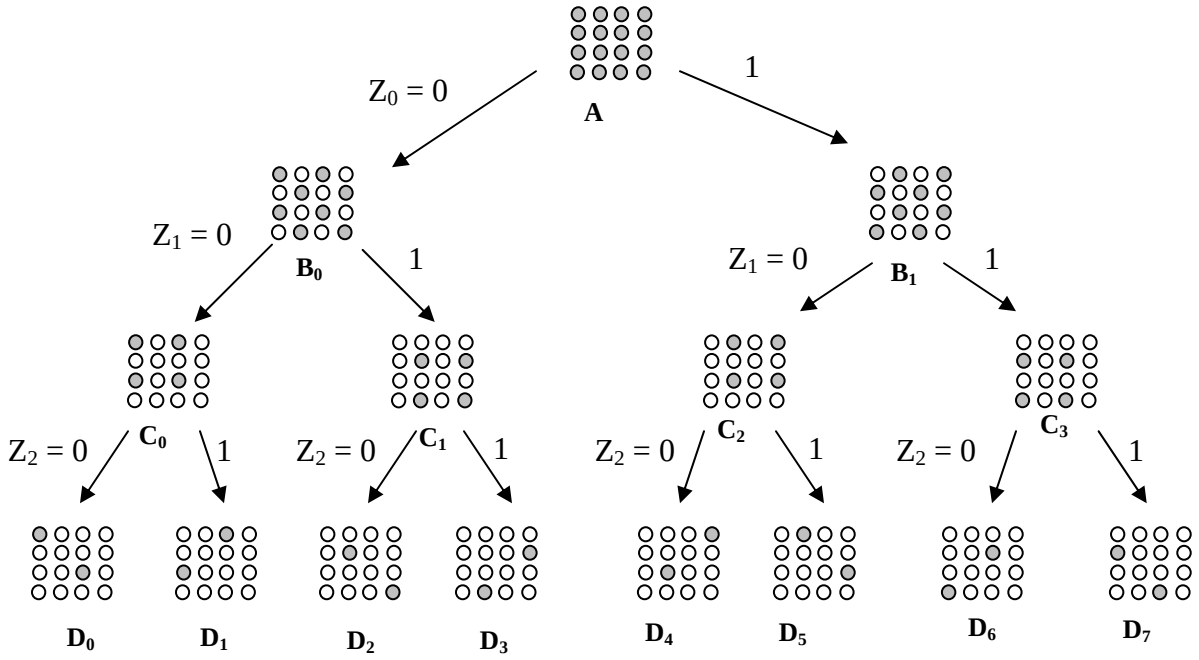


Fig. 6.36.2 *Set partitioning of the 16QAM constellation*

A TCM encoder consists of a convolutional encoder cascaded to a signal mapper. **Fig. 6.36.3** shows the general structure of a TCM encoder. As shown in the figure, a group of m-bits are considered at a time. The rate n/(n+1) convolutional encoder codes n information bits into a codeword of (n+1) bits while the remaining (m-n) bits are not encoded. However, the new group of (n+1+m-n) = m+1 bits are used to select one of the 2^{m+1} points from the signal space following the technique of set-partitioning.

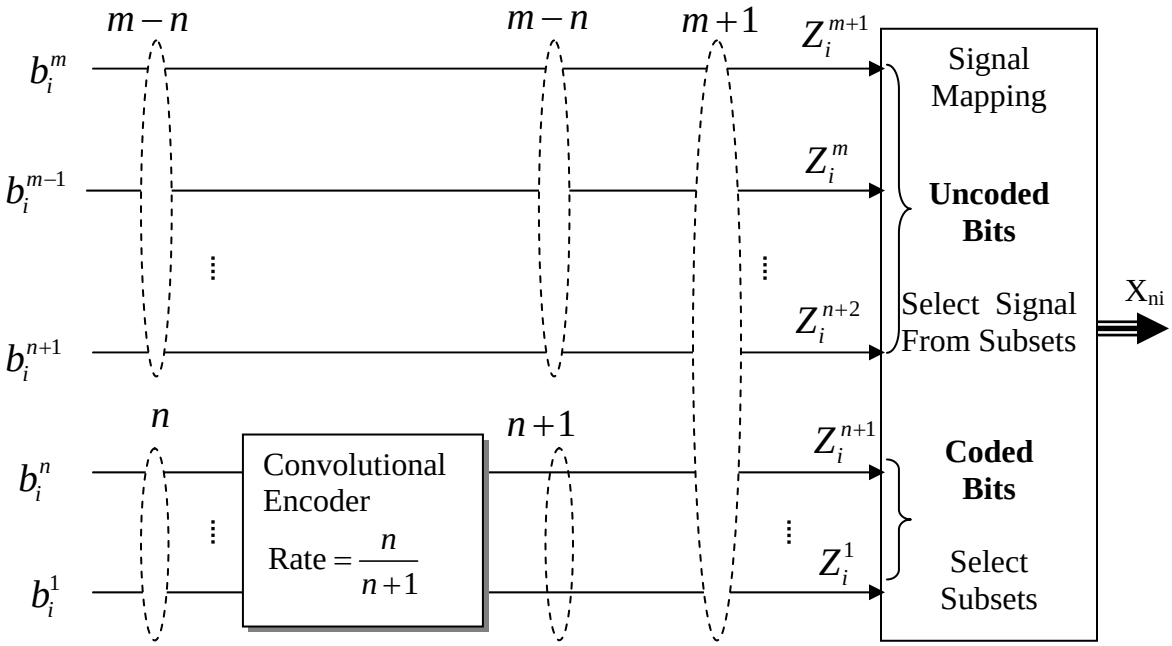


Fig. 6.36.3 *General structure of TCM encoder*

The convolutional encoder may be one of several possible types such a linear non-systematic feed forward type or a feedback type etc. **Fig. 6.36.4** shows an encoder of r = 2/3, suitable for use with a 8-point signal constellation.

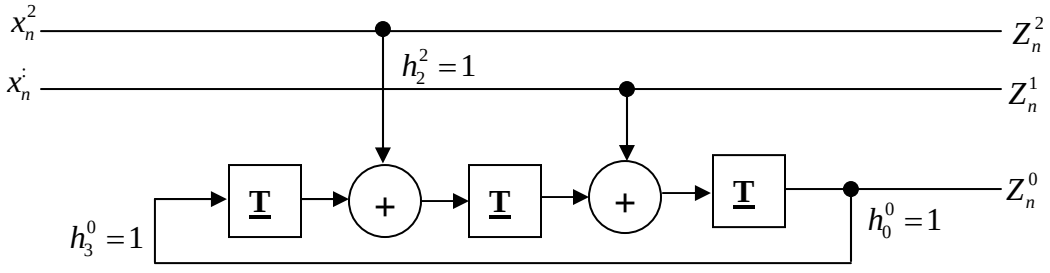


Fig. 6.36.4 *Structure of a TCM encoder based on r=2/3 convolutional coding and 8-PSK modulation format*

Decoding TCM

The concept of code trellis, as introduced in Lesson #35, is expanded to describe the operation of a TCM decoder. Distance properties of a TCM scheme can be studied through its trellis diagram in the same way as for convolutional codes. The TCM decoder-demodulator searches the most likely path through the trellis from the received

sequence of noisy signal points. Because of noise, the chosen path may not always coincide with the correct path. The Viterbi algorithm is also used in the TCM decoder. Note that there is a one-to-one correspondence between signal sequences and the paths in a trellis. Hence, the maximum-likelihood (ML) decision on a sequence of received symbols consists of searching for the trellis path with the minimum Euclidean distance to the received sequence. The resultant trellis for TCM looks somewhat different compared to the trellis of a convolutional code only. There will be multiple parallel branches between two nodes all of which will correspond to the same bit pattern for the first (n+1) bits as obtained from the convolutional encoder.

Module 6

Channel Coding

Lesson 37

Turbo Coding

After reading this lesson, you will learn about

- ***Turbo Encoding and Turbo Code Structures;***
- ***SISO Decoders;***
- ***MAP Algorithms;***
- ***Applications;***
- ***Turbo Product Codes (TCP);***

Turbo codes represent a class of parallel concatenation of two convolutional codes. The parallel-concatenated codes have several advantages over the serial concatenated ones. The parallel decoder facilitates the idea of feedback in decoding to improve the performance of the system. There are some differences between conventional convolutional code and turbo codes. Several parameters affect the performance of turbo codes such as: a) component decoding algorithms, b) number of decoding iterations, c) generator polynomials and constraint lengths of the component encoders and d) interleaver type.

Turbo Encoding and Turbo Code Structures

A turbo encoder is sometimes built using two identical convolutional codes of special type, such as, recursive systematic (RSC) type with parallel concatenation. An individual encoder is termed a component encoder. An interleaver separates the two component encoders. The interleaver is a device that permutes the data sequence in some predetermined manner. Only one of the systematic outputs from the two component encoders is used to form a codeword, as the systematic output from the other component encoder is only a permuted version of the chosen systematic output.

Fig. 6.37.1 shows the block diagram of a turbo encoder using two identical encoders. The first encoder outputs the systematic V_0 and recursive convolutional V_1 sequences while the second encoder discards its systematic sequence and only outputs the recursive convolutional V_2 sequence. There are several types of interleavers such as,

a) Block interleaver,
b) Diagonal interleaver,
c) Odd-even block interleaver,
d) Pseudo-random interleaver,
e) Convolutional interleaver,
f) Helical interleaver,
g) Uniform interleaver,
h) Cyclic shift interleaver and
i) Code matched interleaver.

Depending on the number of input bits to a component encoder it may be binary or m-binary encoder. Encoders are also categorized as systematic or non-systematic. If the component encoders are not identical then it is called an asymmetric turbo code.

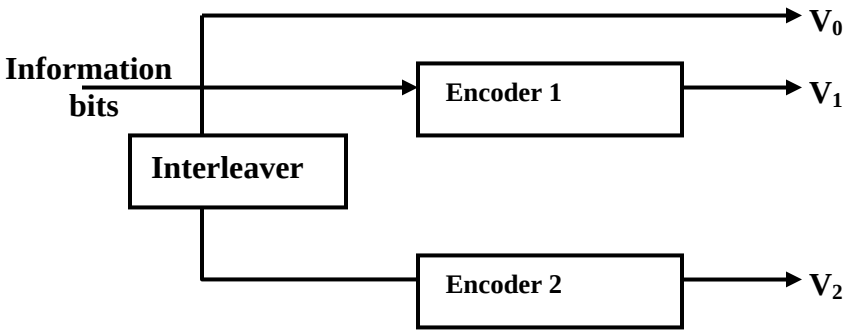


Fig. 6.37.1 *Generic block diagram of a turbo encoder*

Fig. 6.37.2 shows a schematic diagram for the iterative decoding procedure using two 'Soft-in-Soft-out' (SISO) component decoders. The first SISO decoder generates the soft output and subsequently an *extrinsic information* (EI). The extrinsic information is interleaved and used by the second SISO decoder as the estimate of the *a priori* probability (APP). The second SISO decoder also produces the extrinsic information and passes it after de-interleaving to the first SISO decoder to be used during the subsequent decoding operation.

Some of the major decoding approaches, developed for turbo decoding are:

a) Maxium Aposteriori Probability (MAP),
b) Log-MAP,
c) Max-Log-MAP and
d) Soft Output Viterbi Algorithm (SOVA).

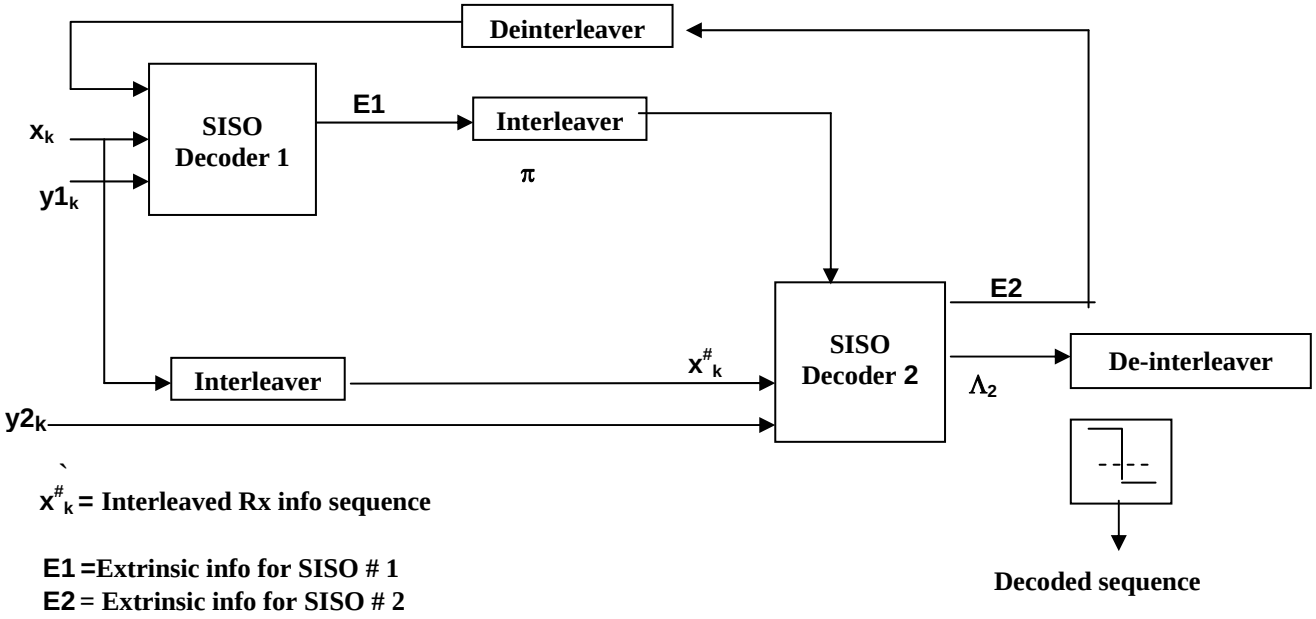


Fig. 6.37.2. *Block diagram of iterative turbo decoder*

The MAP algorithm is a Maximum Likelihood (ML) algorithm and the SOVA is asymptotically an ML algorithm at moderate and high SNR. The MAP algorithm finds the most probable information bit that was transmitted, while the SOVA finds the most probable information sequence to have been transmitted given the code sequence. That means the MAP algorithm minimizes the bit or symbol error probability, where as SOVA minimizes the word error probability. Information bits returned by the MAP algorithm need not form a connected path through the trellis while for SOVA it will be a connected path.

However the MAP algorithm is not easily implement able due to its complexities. Several approximations on the MAP algorithm are now available, such as the Max-Log-MAP algorithm where computations are largely in the logarithmic domain and hence values and operations are easier to implement. The Log-MAP algorithm avoids the approximations in the Max-Log-MAP algorithm through the use of a simple correction function at each maximization operation and thus its performance is close to that of the MAP algorithm.

A complexity comparison of different decoding methods per unit time for (n,k) convolutional code with memory order v is given in **Table 6.37.1**. Assuming that one table-look-up operation is equivalent to one addition, one may see that the Log-MAP algorithm is about three times complex than the SOVA algorithm and the Max-Log-MAP algorithm is about twice as complex as the SOVA algorithm.

	MAP	**Log-MAP**	**Max-Log-MAP**	**SOVA**
Additions	$2.2^k.2^v+6$	$6.2^k.2^v+6$	$4.2^k.2^v+8$	$2^k.2^v+9$
Multiplications	$5.2^k.2^v+8$	$2^k.2^v$	$2.2^k.2^v$	$2^k.2^v$
Max.operations		4.2^v-2	4.2^v-2	2.2^v-1
Look-ups		4.2^v-2		
Exponentiation	$2.2^k.2^v$			

Table 6.37.1 *Complexity comparison of various decoding algorithms*

Applications

Turbo codes have been proposed for various communication systems such as deep space, cellular mobile and satellite communication networks. Turbo code, due to its excellent error correcting capability, has been adopted by several standards as Consultative Committee for Space Data Systems (CCSDS) for space communication, CDMA2000, UMTS, 3GPP, W-CDMA for cellular mobile, DVB-RCS *(Digital Video Broadcasting – [with a] Return Channel [through] Satellite)* for Satellite communications.

Turbo Product Codes (TPC)

A Turbo Product Code is a concatenation of two block codes on which the principle of "turbo" or iterative soft-input/soft-output (SISO) decoding can be applied in a phased manner. The decoding process is iterated several times feeding the output of second component decoder back to the input of the first decoder. A product code may be viewed as a relatively large code built from smaller block codes. A two-dimensional product code is built from two component codes with parameters $C_1(n_1, k_1, d_1)$ and C_2 with parameter (n_2, k_2, d_2), where n_i, k_i and d_i stand for code word length, number of information bits, and minimum hamming distance respectively.

The product code $P = C_1 \text{ X } C_2$ is obtained by placing (k_1 X k_2) information bits in an array of K_1 rows and K_2 columns. The parameters of product code P are $n = n_1 X n_2$, $k = k_1 X k_2$, $d = d_1 X d_2$ and code rate is $R = R_1 X R_2$, where R_i is the code rate of C_i. Thus, very long block codes can be built with large minimum Hamming distance. **Fig. 6.37.3** shows the procedure for construction of a 2D product code using two block codes C1 and C2 .

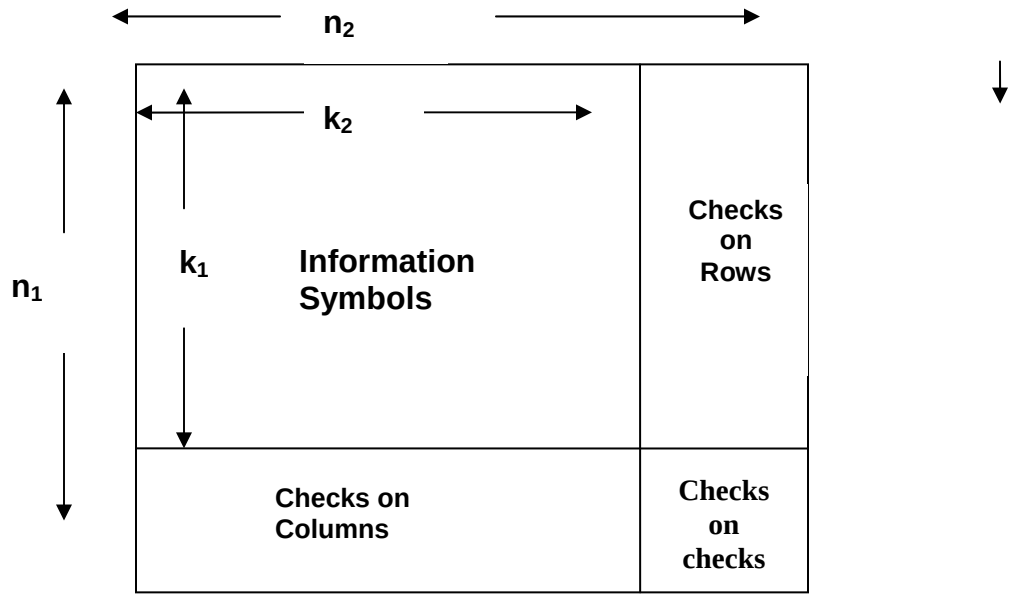


Fig. 6.37.3 *An example of a 2D product code constructed using two component codes*

Module 7

Spread Spectrum and Multiple Access Technique

Lesson 38

Introduction to Spread Spectrum Modulation

After reading this lesson, you will learn about

- ***Basic concept of Spread Spectrum Modulation;***
- ***Advantages of Spread Spectrum (SS) Techniques;***
- ***Types of spread spectrum (SS) systems;***
- ***Features of Spreading Codes;***
- ***Applications of Spread Spectrum;***

Introduction

Spread spectrum communication systems are widely used today in a variety of applications for different purposes such as access of same radio spectrum by multiple users (multiple access), anti-jamming capability (so that signal transmission can not be interrupted or blocked by spurious transmission from enemy), interference rejection, secure communications, multi-path protection, etc. However, irrespective of the application, all spread spectrum communication systems satisfy the following criteria-

(i) As the name suggests, bandwidth of the transmitted signal is much greater than that of the message that modulates a carrier.

(ii) The transmission bandwidth is determined by a factor independent of the message bandwidth.

The power spectral density of the modulated signal is very low and usually comparable to background noise and interference at the receiver.

As an illustration, let us consider the DS-SS system shown in **Fig 7.38.1(a) and (b)**. A random spreading code sequence c(t) of chosen length is used to 'spread'(multiply) the modulating signal m(t). Sometimes a high rate pseudo-noise code is used for the purpose of spreading. Each bit of the spreading code is called a 'chip'. Duration of a chip (T_c) is much smaller compared to the duration of an information bit (T). Let us consider binary phase shift keying (BPSK) for modulating a carrier by this spread signal. If m(t) represents a binary information bit sequence and c(t) represents a binary spreading sequence, the 'spreading' or multiplication operation reduces to modulo-2 or ex-or addition. For example, if the modulating signal m(t) is available at the rate of 10 Kbits per second and the spreading code c(t) is generated at the rate of 1 Mbits per second, the spread signal d(t) is generated at the rate of 1 Mega Chips per second. So, the null-to-null main lobe bandwidth of the spread signal is now 2 MHz. We say that bandwidth has been 'spread' by this operation by a factor of hundred. This factor is known as the spreading gain or process gain (PG). The process gain in a practical system is chosen based on the application.

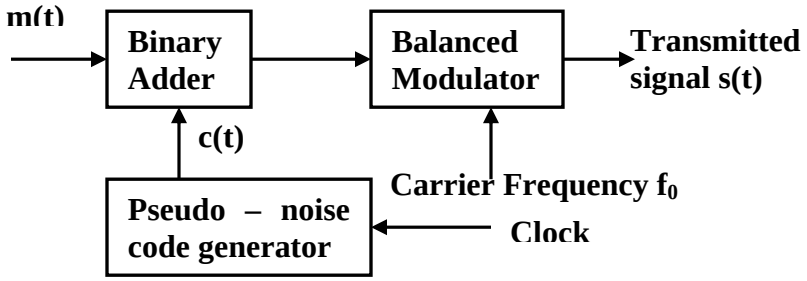


Fig: 7.38.1 (a) *Direct sequence spread spectrum transmitter*

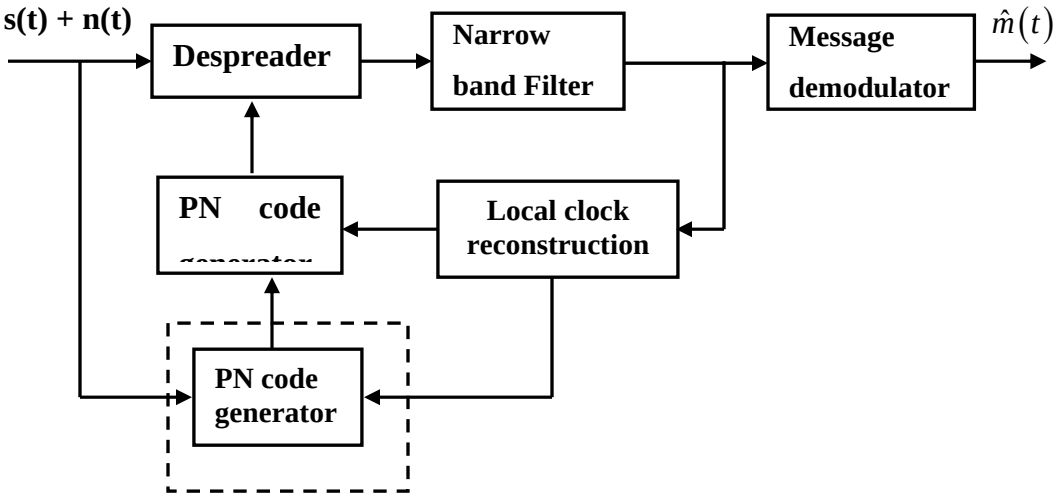


Fig: 7.38.1 (b) *Direct sequence spread spectrum receiver*

On BPSK modulation, the spread signal becomes, s(t) = d(t).coswt. **Fig.7.38.1 (b)** shows the baseband processing operations necessary after carrier demodulation. Note that, at the receiver, the operation of despreading requires the generation of the same spreading code incorrect phase with the incoming code. The pseudo noise (PN) code synchronizing module detects the phase of the incoming code sequence, mixed with the information sequence and aligns the locally generated code sequence appropriately. After this important operation of code alignment (i.e. synchronization) the received signal is 'despread' with the locally constructed spreading code sequence. The dispreading operation results in a narrowband signal, modulated by the information bits only. So, a conventional demodulator may be used to obtain the message signal estimate.

Advantages of Spread Spectrum (SS) Techniques

a) Reduced interference: In SS systems, interference from undesired sources is considerably reduced due to the processing gain of the system.

b) Low susceptibility to multi-path fading: Because of its inherent frequency diversity properties, a spread spectrum system offers resistance to degradation in signal quality due to multi-path fading. This is particularly beneficial for designing mobile communication systems.

c) Co-existence of multiple systems: With proper design of pseudo-random sequences, multiple spread spectrum systems can co-exist.

d) Immunity to jamming: An important feature of spread spectrum is its ability to withstand strong interference, sometimes generated by an enemy to block the communication link. This is one reason for extensive use of the concepts of spectrum spreading in military communications.

Types of SS

Based on the kind of spreading modulation, spread spectrum systems are broadly classified as-

(i) Direct sequence spread spectrum (DS-SS) systems
(ii) Frequency hopping spread spectrum (FH-SS) systems
(iii) Time hopping spread spectrum (TH-SS) systems.
(iv) Hybrid systems

Direct Sequence (DS) Spread Spectrum System (DSSS)

The simplified scheme shown in **Fig. 7.38.1** is of this type. The information signal in DSSS transmission is spread at baseband and then the spread signal is modulated by a carrier in a second stage. Following this approach, the process of modulation is separate from the spreading operation. An important feature of DSSS system is its ability to operate in presence of strong co-channel interference. A popular definition of the processing gain (PG) of a DSSS system is the ratio of the signal bandwidth to the message bandwidth.

A DSSS system can reduce the effects of interference on the transmitted information. An interfering signal may be reduced by a factor which may be as high as the processing gain. That is, a DSSS transmitter can withstand more interference if the length of the PN sequence is increased. The output signal to noise ratio of a DSSS receiver may be expressed as: $(SNR)_o = PG.\ (SNR)_I$, where $(SNR)_I$ is the signal to noise ratio before the dispreading operation is carried out.

A major disadvantage of a DSSS system is the 'Near-Far effect', illustrated in **Fig.7.38.2**.

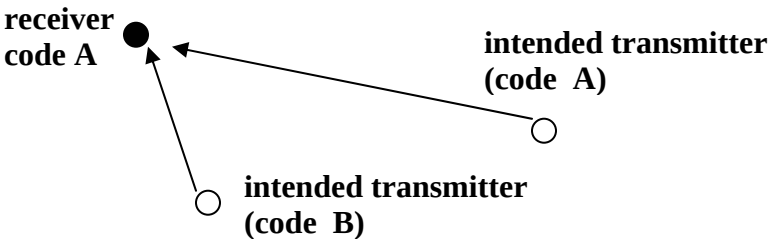


Fig.7.38.2 *Near-far effect*

This effect is prominent when an interfering transmitter is close to the receiver than the intended transmitter. Although the cross-correlation between codes A and B is low, the correlation between the received signal from the interfering transmitter and code A can be higher than the correlation between the received signal from the intended transmitter and code A. So, detection of proper data becomes difficult.

Frequency Hopping Spread Spectrum

Another basic spread spectrum technique is frequency hopping. In a frequency hopping (FH) system, the frequency is constant in each time chip; instead it changes from chip to chip. An example FH signal is shown in **Fig.7.38.3.**

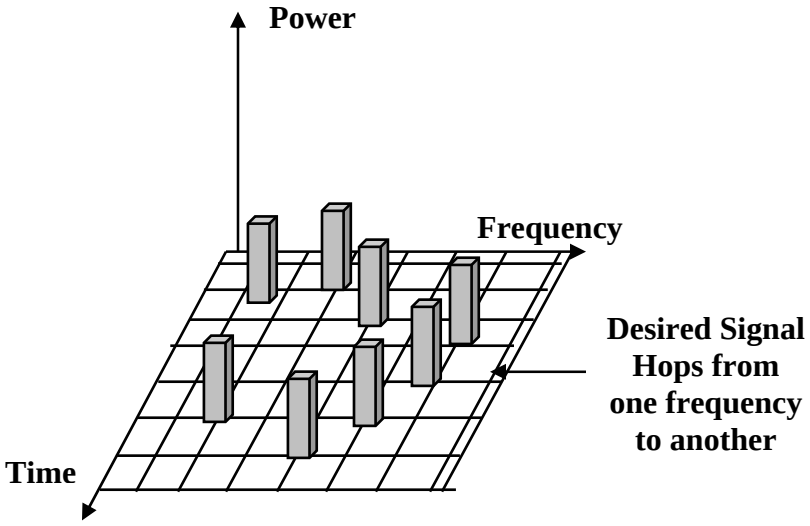


Fig. 7.38.3. *Illustration of the principle of frequency hopping*

Frequency hopping systems can be divided into fast-hop or slow-hop. A fast-hop FH system is the kind in which hopping rate is greater than the message bit rate and in the slow-hop system the hopping rate is smaller than the message bit rate. This differentiation is due to the fact that there is a considerable difference between these two FH types. The FH receiver is usually non-coherent. A typical non-coherent receiver architecture is represented in **Fig.7.38.4.**

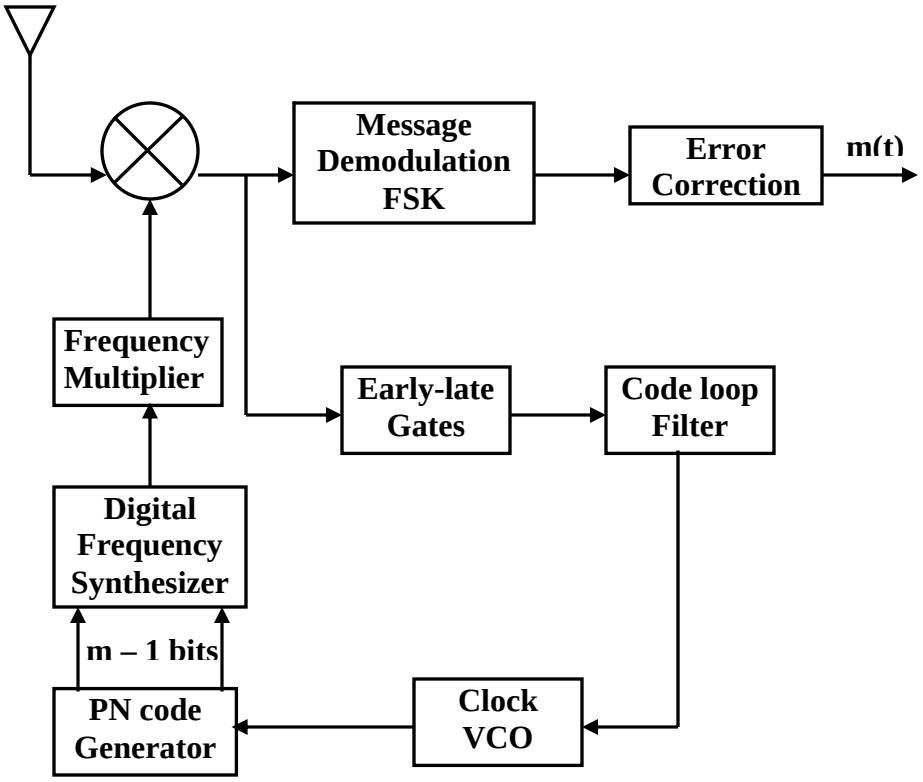


Fig. 7.38.4 *Block diagram of a non-coherent frequency-hopping receiver*

The incoming signal is multiplied by the signal from the PN generator identical to the one at the transmitter. Resulting signal from the mixer is a binary FSK, which is then demodulated in a "regular" way. Error correction is then applied in order to recover the original signal. The timing synchronization is accomplished through the use of early-late gates, which control the clock frequency

Time Hopping

A typical time hopping signal is illustrated in the figure below. It is divided into frames, which in turn are subdivided into M time slots. As the message is transmitted only one time slot in the frame is modulated with information (any modulation). This time slot is chosen using PN generator.

All of the message bits gathered in the previous frame are then transmitted in a burst during the time slot selected by the PN generator. If we let: T_f = frame duration, k = number of message bits in one frame and $T_f = k \times t_m$, then the width of each time slot in a frame is $\frac{T_f}{M}$ and the width of each bit in the time slot is $\frac{T_f}{kM}$ or just $\frac{t_m}{M}$. Thus, the transmitted signal bandwidth is 2M times the message bandwidth.

A typical time hopping receiver is shown in **Fig.7.38.5**. The PN code generator drives an on-off switch in order to accomplish switching at a given time in the frame. The output of this switch is then demodulated appropriately. Each message burst is stored and re-timed to the original message rate in order to recover the information. Time hopping is at times used in conjunction with other spread spectrum modulations such as DS or FH. **Table 7.38.1** presents a brief comparison of major features of various SS schemes.

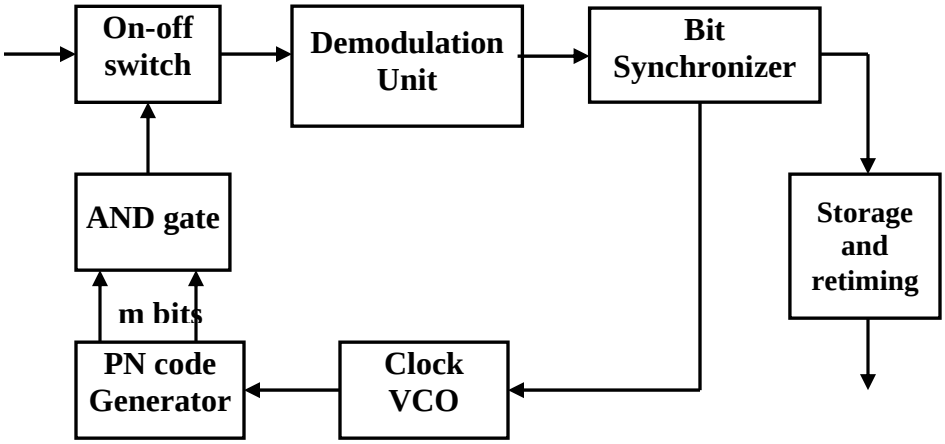


Fig. 7.38.5 *Block diagram of a time hopping receiver*

Spreading Method	Merits	Demerits
Direct Sequence	i) Simpler to implement ii) Low probability of interception iii) Can withstand multi-access interference reasonably well	i) Code acquisition may be difficult ii) Susceptible to Near-Far problem iii) Affected by jamming
Frequency Hopping	i) Less affected by Near-Far problem ii) Better for avoiding jamming iii) Less affected by multi-access interference	i) Needs FEC ii) Frequency acquisition may be difficult
Time Hopping	i) Bandwidth efficient ii) Simpler than FH system	i) Elaborate code acquisition is needed. ii) Needs FEC

Table 7.38.1 *Comparison of features of various spreading techniques*

Hybrid System: DS/(F) FH

The DS/FH Spread Spectrum technique is a combination of direct-sequence and frequency hopping schemes. One data bit is divided over several carrier frequencies (**Fig 7.38.6**).

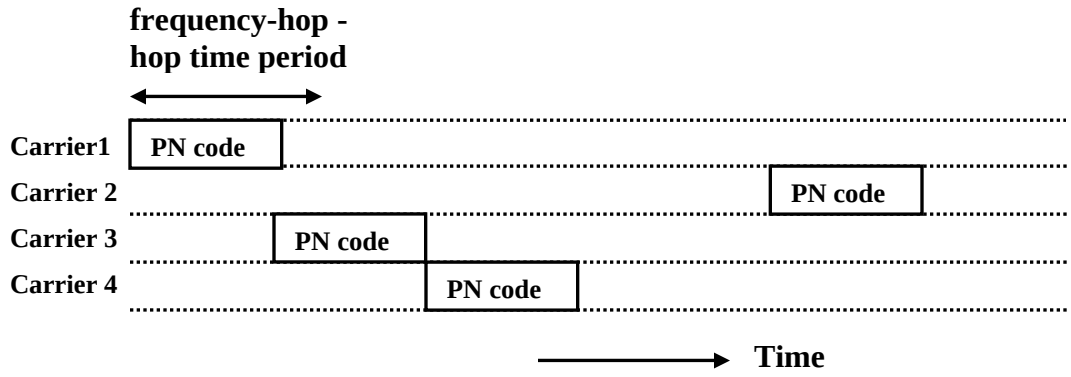


Fig. 7.38.6 *A hybrid DS-FH spreading scheme*

As the FH-sequence and the PN-codes are coupled, a user uses a combination of an FH-sequence and a PN-code.

Features of Spreading Codes

Several spreading codes are popular for use in practical spread spectrum systems. Some of these are Maximal Sequence (m-sequence) length codes, Gold codes, Kasami codes and Barker codes. In this section will be briefly discussed about the m-sequences.

These are longest codes that can be generated by a shift register of a specific length, say, L. An L-stage shift register and a few EX-OR gates can be used to generate an m-sequence of length 2^L -1. **Fig 7.38.7** shows an m-sequence generator using n memory elements, such as flip-flops. If we keep on clocking such a sequence generator, the sequence will repeat, but after 2^L -1 bits. The number of 1-s in the complete sequence and the number of 0-s will differ by one. That is, if L = 8, there will be 128 one-s and 127 zero-s in one complete cycle of the sequence. Further, the auto-correlation of an m-sequence is -1 except for relative shifts of (0 ± 1) chips (**Fig 7.38.8**). This behavior of the auto correlation function is somewhat similar to that of thermal noise as the auto correlation shows the degree of correspondence between the code and its phase-shifted version. Hence, the m-sequences are also known as, pseudo-noise or PN sequences.

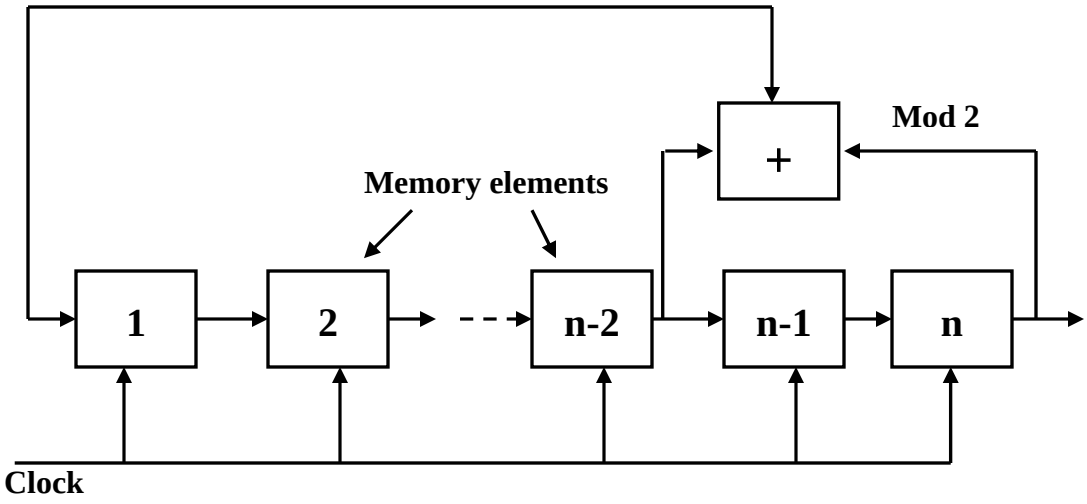


Fig. 7.38. 7 *Maximal length pseudo random sequence generator*

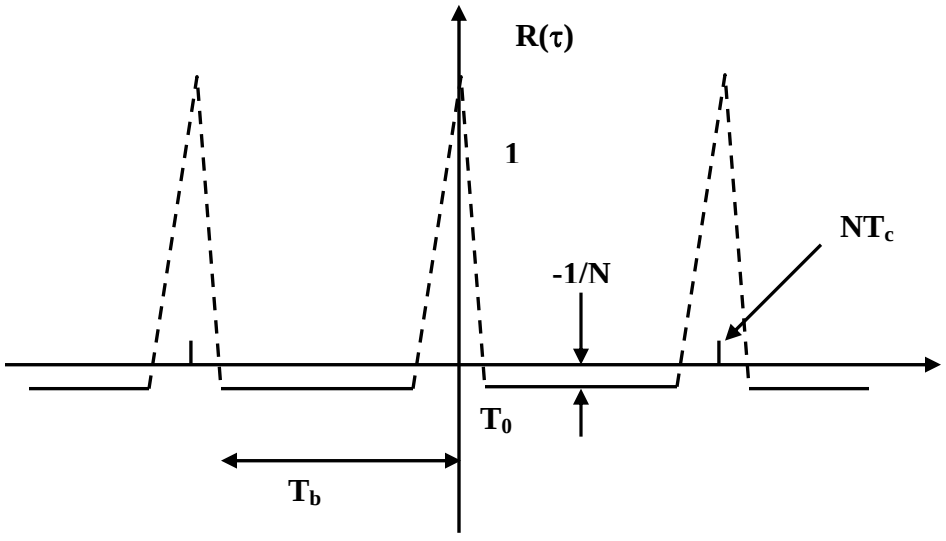


Fig. 7.38. 8 *Autocorrelation function of PN sequence*

Another interesting property of an m-sequence is that, the sequence, when added (modulo-2) with a cyclically shifted version of itself, results in another shifted version of the original sequence. For moderate and large values of L, multiple sequences exist, which are of the same length. The cross correlation of all these codes are studied. All these properties of a PN sequence are useful in the design of a spread spectrum system. Sometimes, to indicate the occurrence of specific patterns of sequences, we define 'run' as a series of ones and zero-s, grouped consecutively. For example, consider a sequence 1011010. We say, the sequence of has three runs of single '0', two runs of single '1' and one run of two ones. In a maximum length sequence of length and 2^L -1, there are exactly $2^{L-(p+2)}$ runs of length 'p' for both of ones and zeros except that there is only one run containing L one-s and one containing (L-1) zero-s. There is no run of zero-s of length L or ones of length (L-1). That is, the number of runs of each length is a decreasing power of two as the run length increases.

It is interesting to note that, multiple m-sequences exist for a particular value of L > 2. The number of available m- sequences is denoted by $\frac{\phi(2^L - 1)}{L}$. The numerator $\phi(2^L - 1)$ is known as the Euler number, i.e. the number of positive integers, including 1, that are relatively prime to L and less than (2^L -1). When (2^L -1) itself is a prime number, all positive integers less than this number are relatively prime to it. For example, if L = 5, it is easy to find that the number of possible sequences = $\frac{30}{5} = 6$.

If the period of an m-sequence is N chips, N = (2^n –1), where 'n' is the number of stages in the code generator. The autocorrelation function of an m-sequence is periodic in nature and it assumes only two values, viz. 1 and (-1/N) when the shift parameter (τ) is an integral multiple of chip duration.

Several properties of PN sequences are used in the design of DS systems. Some features of maximal length pseudo random periodic sequences (m-sequence or PN sequence) are noted below:

a) Over one period of the sequence, the number of '+1' differs from the number of '-1' by exactly one.
b) Also the number of positive runs equals the number of negative runs.
c) Half of the runs of bits in every period of the same sign (i.e. +1 or -1) are of length 1, one fourth of the runs of bits are of length 2, one eighth of the runs of bits are of length 3 and so on. The autocorrelation of a periodic sequence is two-valued.

Applications of Spread Spectrum

A specific example of the use of spread spectrum technology is the North American Code Division Multiple Access (CDMA) Digital Cellular (IS-95) standard. The CDMA employed in this standard uses a spread spectrum signal with 1.23-MHz spreading bandwidth. Since in a CDMA system every user is a source of interference to other users, control of the transmitted power has to be employed (due to near-far problem). Such control is provided by sophisticated algorithms built into control stations.

The standard also recommends use of forward error-correction coding with interleaving, speech activity detection and variable-rate speech encoding. Walsh code is used to provide 64 orthogonal sequences, giving rise to a set of 64 orthogonal 'code channels'. The spread signal is sent over the air interface with QPSK modulation with Root Raised Cosine (RRC) pulse shaping. Other examples of using spread spectrum technology in commercial applications include satellite communications, wireless LANs based on IEEE 802.11 standard etc.

Module 7

Spread Spectrum and Multiple Access Technique

Lesson 39

Code Acquisition

After reading this lesson, you will learn about

- ***Basics of Code Acquisition Schemes;***
- ***Classification of Code Acquisition Schemes;***

Code synchronization is the process of achieving and maintaining proper alignment between the reference code in a spread spectrum receiver and the spreading sequence that has been used in the transmitter to spread the information bits. Usually, code synchronization is achieved in two stages: a) code acquisition and b) code tracking. Acquisition is the process of initially attaining coarse alignment (typically within ± half of the chip duration), while tracking ensures that fine alignment within a chip duration is maintained. In this lesson, we primarily discuss about some concepts of code acquisition.

Code Acquisition Schemes

Acquisition is basically a process of searching through an uncertainty region, which may be one-dimensional, e.g. in time alone or two-dimensional, viz. in time and frequency (if there is drift in carrier frequency due to Doppler effect etc.) – until the correct code phase is found. The uncertainty region is divided into a number of cells. In the one-dimensional case, one cell may be as small as a fraction of a PN chip interval.

The acquisition process has to be reliable and the average time taken to acquire the proper code phase also should be small. There may be other operational requirements depending on the application. For example, some systems may have a specified limit in terms of the time interval 'T' within which acquisition should be complete. For such systems, an important parameter that may be maximized by the system designer is Prob ($t_{acq} \leq$ T). However, for other systems, it may be sufficient to minimize the mean acquisition time E[t_{acq}].

A basic operation, often carried out during the process of code acquisition is the correlation of the incoming spread signal with a locally generated version of the spreading code sequence (obtained by multiplying the incoming and local codes and accumulating the result) and comparing the correlator output with a set threshold to decide whether the codes are in phase or not. The correlator output is usually less than the peak autocorrelation of the spreading code due to a) noise and interference in the received signal or b) time or phase misalignment or c) any implementation related imperfections. If the threshold is set too high (equal or close to the autocorrelation peak), the correct phase may be missed i.e. probability of missed detection (P_M) is high. On the other hand, if the threshold is set too low, acquisition may be declared at an incorrect phase (or time instant), resulting in high probability of false acquisition (P_{FA}). A tradeoff between the two is thus desired.

There are several acquisition schemes well researched and reported in technical publications. Majority of the code acquisition schemes can be broadly classified

according to a) the detector structure used in the receiver (**Fig. 7.39.1**) and b) the search strategy used (**Fig.7.39.2**).

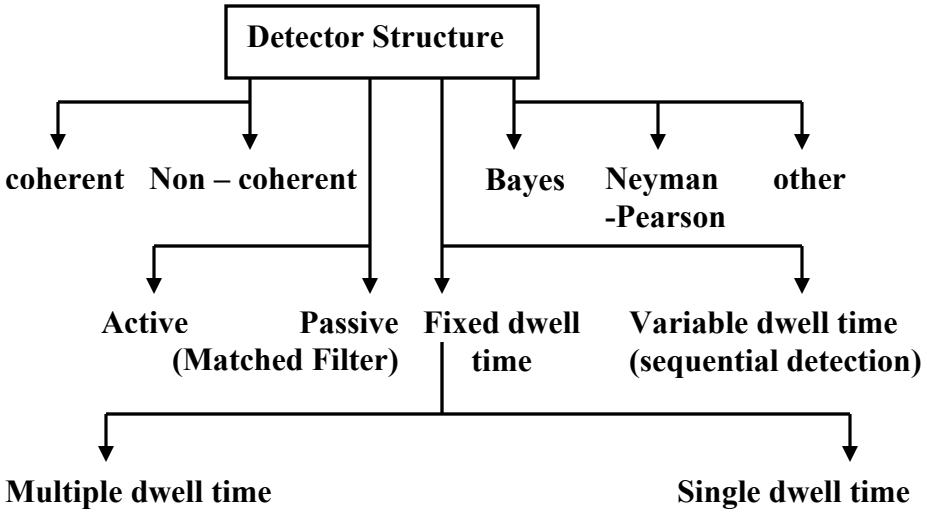


Fig.7.39.1 *Classification of detector structures used for code acquisition on DS-SS systems*

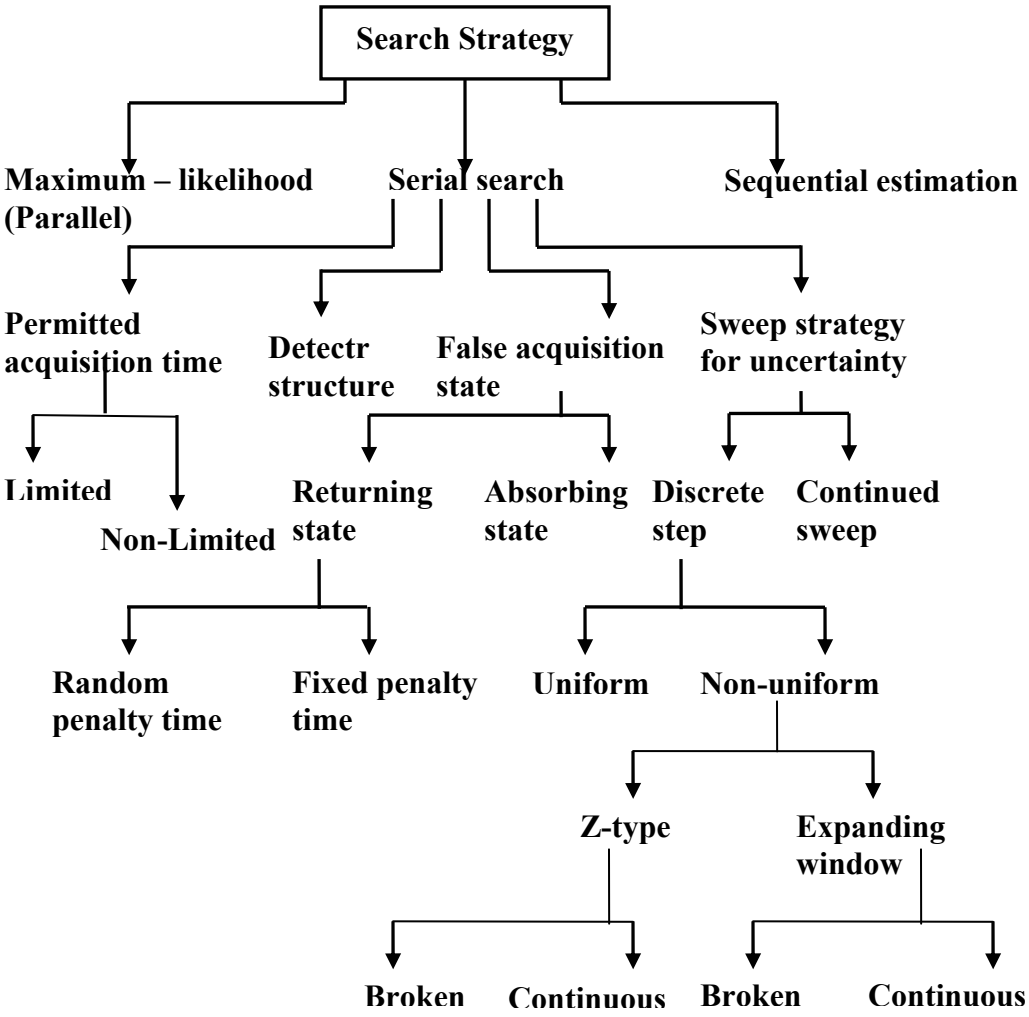


Fig. 7.39.2 *Classification of acquisition schemes based on search strategy*

The detector structure may be coherent or non-coherent in nature. If the carrier frequency and phase are not precisely known, a non-coherent structure of the receiver is preferred. **Fig 7.39.3** shows a coarse code acquisition scheme for a non-coherent detector.

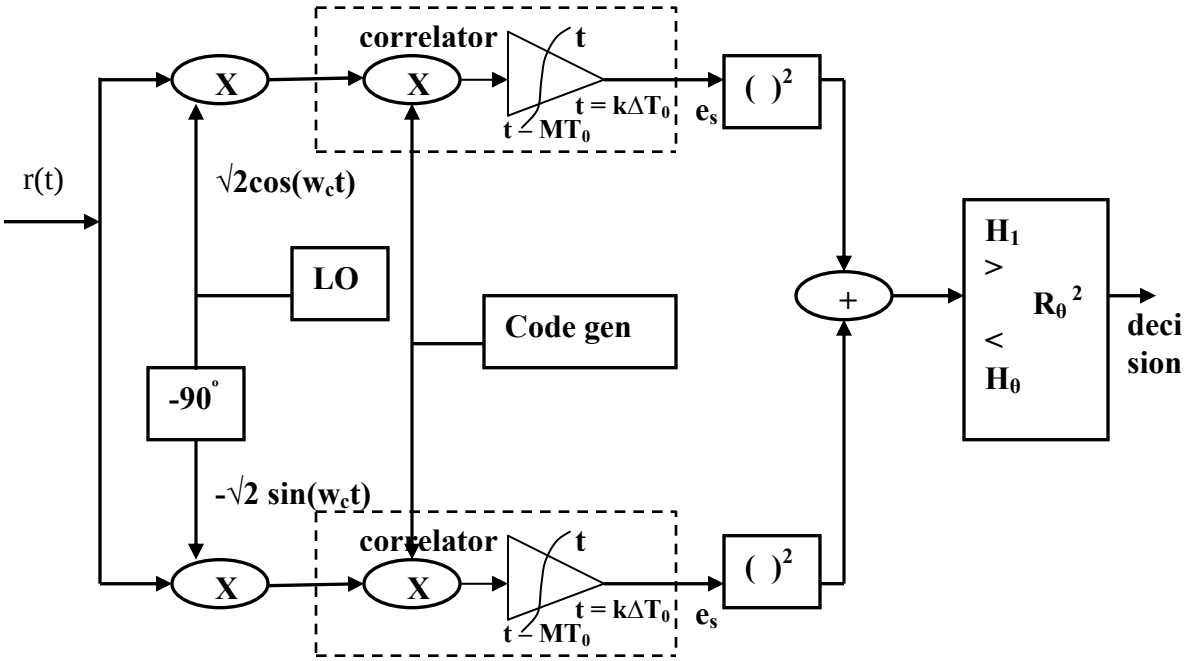


Fig. 7.39.3 *Structure of a non-coherent detector*

A code acquisition scheme may also be active or passive. In a receiver employing active code acquisition (sometimes called code detection), portions of the incoming and the local codes (a specific phase shifted version) are multiplied bit by bit and the product is accumulated over a reasonable interval before comparison is made against a decision threshold. If the accumulated value does not exceed the threshold, the process is repeated with new samples of the incoming code and for another offset version of the local code.

A passive detector, on the other hand, is much like a matched filter with the provision to store the incoming code samples in a shift register. With every incoming chip, a decision is made (high decision rate) based on a correlation interval equal to the length of the matched filter (MF) correlator. A disadvantage of this approach is the need for more hardware when large correlation intervals are necessary.

Code acquisition schemes are also classified based on the criterion used for deciding the threshold. For example, a Bayes' code detector minimizes the average probability of missed detection while a Neyman Pearson detector minimizes the probability of missed detection for a particular value of P_{FA}.

The examination or observation interval is known as 'dwell time' and several code acquisition schemes (and the corresponding detectors) are named after the dwell time that they use (**Fig.7.39.1)**. Detectors working with fixed dwell times may or may not employ a verification stage (multiple dwell times). A verification process enhances the reliability of decision and hence is incorporated in many practical systems.

Classification based on search strategy

As noted earlier, the acquisition schemes are also classified on the basis of the search strategy in the region of uncertainty (**Fig 7.39.2**). Following the maximum likelihood estimation technique, the incoming signal is simultaneously (in parallel) correlated with all possible time-shifted versions of the local code and the local code phase that yields the highest output is declared as the phase of the incoming code sequence. This method requires a large number of correlators. However, the strategy may also be implemented (somewhat approximately) in a serial manner, by correlating the incoming code with each phase-shifted version of the local code, and taking a decision only after the entire code length is scanned. While one correlator is sufficient for this approach, the acquisition time increases linearly with the correlation length. Further, since the noise conditions are usually not the same for all code phases, this approach is not strictly a maximum likelihood estimation algorithm.

Another search strategy is sequential estimation. This is based on the assumption that, in the absence of noise, if 'n' consecutive bits of the incoming PN code (where "n" is the length of the PN code generator) are loaded into the receivers code generator, and this is used as the initial condition, the successively generated bits will automatically be in phase with the incoming ones. Cross correlation between the generated and incoming codes is done to check whether synchronization has been attained or not. If not, the next n bits are estimated and loaded. This algorithm usually yields rapid acquisition and hence is called the RASE (Rapid Acquisition by Sequential Estimation) algorithm. However, this scheme works well only when the noise associated with the received spread signal is low.

An important family of code acquisition algorithms is known as Serial Search. Following the serial search technique, a group of cells (probable pockets in the uncertainty region) are searched in a serial fashion until the correct cell (implying the correct code phase) is found. This process of serial search may be successful anywhere in the uncertainty region and hence the average acquisition time is much less than that in the maximum likelihood estimation schemes, though a maximum likelihood estimation scheme yields more accurate result. Incidentally, a serial search algorithm performs better than RASE algorithm at low CNR. Serial search schemes are also easier to implement. In the following, we briefly mention a practical code search techniques known as 'Subsequence Matched Filtering'.

Proposed in 1984, this is a rapid acquisition scheme for CDMA systems, that employs Subsequence Matched Filtering (SMF). The detector consists of several correlator based matched filters, each matched to a subsequence of length M. The subsequence may or may not be contagious, depending on the length of the code and the hardware required to span it. With every incoming chip of the received waveform, the largest of the SMF value is compared against a pre-selected threshold. Threshold exceedance leads to loading that subsequence in the local PN generator for correlation over a longer time interval. If the correlation value does not exceed a second threshold, a negative feedback is generated, and a new state estimate is loaded into the local

generator. Otherwise, the PN generation and correlation process continues. ***Fig. 7.39.4*** shows the structure of a Subsequence Matched Filter based code acquisition process.

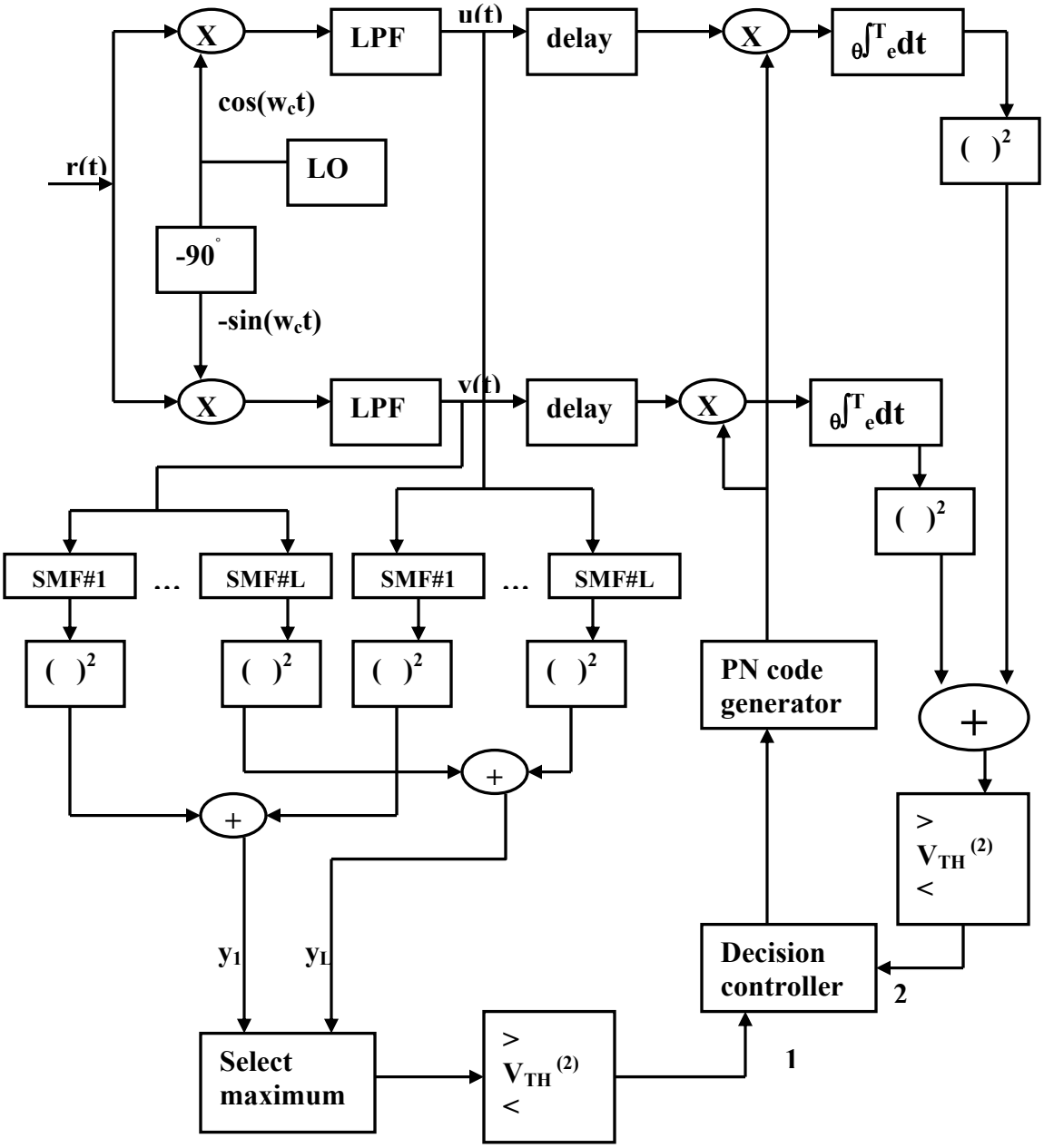


Fig. 7.39.4 *Structure of a Subsequence Matched Filter based code acquisition process*

Module 7

Spread Spectrum and Multiple Access Technique

Lesson 40

Multiple Access Techniques and Cellular CDMA

After reading this lesson, you will learn about

- ***Basic multiple access techniques;***
- ***Use of Code Division Multiple Access (CDMA) in cellular mobile communications***

An important use of the concept of spread spectrum in wireless communication systems is to allow multiple users occupy the same transmission band for simultaneous transmission of signals without considerable interference. The three basic multiple access techniques are briefly mentioned below:

a) Frequency Division Multiple Access (FDMA):

This classical technique has been in use in conventional telephone systems and satellite communication systems. Every user gets a certain frequency band assigned and can use this part of the spectrum to perform its communication. If only a small number of users is active, not the whole resource (frequency-spectrum) is used. Assignment of the channels can be done centrally or by carrier sensing in a mobile. The latter possibility enables random-access.

b) Time Division Multiple Access *(TDMA):*

Every user is assigned one or a set of well-defined time-slots within a 'Time Frame'. A transmitting user sends its own data only in the designated time-slot(s), and waits for the remaining time-frame duration till it gets another time-slot in the next time frame. Precise time synchronization among all users is an important and necessary feature of TDMA multiple access strategy. Usually, a central unit controls the synchronization and the assignment of time-slots.

c) Code Division Multiple Access (*CDMA) / Spread Spectrum Multiple Access (SSMA):*

One or more unique spreading codes are assigned to each user for accessing the RF bandwidth simultaneously for transmission and reception of signals. The spreading codes, assigned to all participating users, are carefully chosen to ensure very low cross-correlation among them. This ensures that the signals from undesired transmitters appear as noise (with no or very poor correlation with the desired signal after dispreading operation). CDMA / SSMA does not need very precise time synchronization among the users and hence, random-access is protocols can be implemented relatively easily.

In the following section, a brief account of CDMA scheme, used in cellular mobile communications, is presented.

Cellular CDMA

Mobile telephony, using the concept of cellular architecture, has been very popular world wide. Such systems are built based on accepted standards, such as GSM (Global System for Mobile communication) and IS-95(Intermediate Standard-95). Several standards of present and future generations of mobile communication systems include CDMA as an important component which allows a satisfactorily large number of users to communicate simultaneously over a common radio frequency band.

Cellular CDMA is a promising access technique for supporting multimedia services in a mobile environment as it helps to reduce the multi-path fading effects and interference. It also supports universal frequency reuse, which implies large teletraffic capacity to accommodate new calling subscribers. In a practical system, however, the actual number of users who can simultaneously use the RF band satisfactorily is limited by the amount of interference generated in the air interface. A good feature is that the teletraffic capacity is 'soft', i.e. there is no 'hard' or fixed value for the maximum capacity. The quality of received signal degrades gracefully with increase in the number of active users at a given point of time.

It is interesting to note that the quality of a radio link in a cellular system is often indicated by the Signal-to-Interference Ratio (SIR), rather than the common metric 'SNR'. Let us remember that in a practical system, the spreading codes used by all the simultaneous users in a cell have some cross-correlation amongst themselves and also due to other propagation features, the signals received in a handset from all transmitters do not appear orthogonal to each other. Hence, the signals from all users, other than the desired transmitter, manifest as interference. In a practical scenario, the total interference power may even momentarily exceed the power of the desired signal. This happens especially when the received signals fluctuate randomly (fading) due to mobility of the users. Fading is a major factor degrading the performance of a CDMA system. While large-scale fading consists of path loss and shadowing, small-scale fading refers to rapid changes in signal amplitude and phase over a small spatial separation.

The desired signal at a receiver is said to be 'in outage' (i.e. momentarily lost) when the SIR goes below an acceptable threshold level. An ongoing conversation may get affected adversely if the outage probability is high or if the duration of outage (often called as 'fade duration') is considerable. On the other hand, low outage probability and insignificant 'average fade duration' in a CDMA system usually implies that more users could be allowed in the system ensuring good quality of signal.