DIGITAL COMMUNICATION SYSTEMS

Subject Code : 15A04502
Regulations : JNTUA – R15
Class : III B.Tech I-Sem(ECE)

CHADALAWADA RAMANAMMA ENGINEERING COLLEGE
(AUTONOMOUS)
Chadalawada Nagar, Renigunta Road, Tirupati – 517 506

Department of Electronics and Communication Engineering
Course Objectives:

- The students to be able to understand, analyze, and design fundamental digital communication systems.
- The course focuses on developing a thorough understanding of digital communication systems by using a series of specific examples and problems.

Course Outcomes:

After the completion of the course, student will be able to:

- Understand the elements of DCS & the fundamentals concepts of sampling theorem along with different coding and modulation techniques
- Understand the basic principles of baseband and passband digital modulation schemes
- Analyze probability of error performance of digital systems and are able to design digital communication systems

UNIT – I
Source Coding Systems: Introduction, sampling process, quantization, quantization noise, conditions for optimality of quantizer, encoding, Pulse-Code Modulation (PCM), Line codes, Differential encoding, Regeneration, Decoding & Filtering, Noise considerations in PCM systems, Time-Division Multiplexing (TDM), Synchronization, Delta modulation (DM), Differential PCM (DPCM), Processing gain, Adaptive DPCM (ADPCM), Comparison of the above systems.

UNIT – II
UNIT – III
Signal Space Analysis: Introduction, Geometric representation of signals, Gram-Schmidt orthogonalization procedure, Conversion of the Continuous AWGN channel into a vector channel, Coherent detection of signals in noise, Correlation receiver, Equivalence of correlation and Matched filter receivers, Probability of error, Signal constellation diagram.

UNIT - IV
Passband Data Transmission: Introduction, Passband transmission model, Coherent phase-shift keying – binary phase shift keying (BPSK), Quadrature shift keying (QPSK), Binary Frequency shift keying (BFSK), Error probabilities of BPSK, QPSK, BFSK, Generation and detection of Coherent BPSK, QPSK, & BFSK, Power spectra of above mentioned modulated signals, M-array PSK, M-array quadrature amplitude modulation (M-array QAM), Non-coherent orthogonal modulation schemes - Differential PSK, Binary FSK, Generation and detection of non-coherent BFSK, DPSK, Comparison of power bandwidth requirements for all the above schemes.

UNIT – V
Channel Coding: Error Detection & Correction - Repetition & Parity Check Codes, Interleaving, Code Vectors and Hamming Distance, Forward Error Correction (FEC) Systems, Automatic Retransmission Query (ARQ) Systems, Linear Block Codes – Matrix Representation of Block Codes, Convolutional Codes – Convolutional Encoding, Decoding Methods.

TEXT BOOKS:

REFERENCES:
INTRODUCTION

The purpose of a Communication System is to transport an information bearing signal from a source to a user destination via a communication channel.

MODEL OF A COMMUNICATION SYSTEM (ANALOG)

![Block diagram of Communication System](image)

The three basic elements of every communication systems are Transmitter, Receiver and Channel.

The Overall purpose of this system is to transfer information from one point (called Source) to another point, the user destination.

The message produced by a source, normally, is not electrical. Hence an input transducer is used for converting the message to a time – varying electrical quantity called message signal. Similarly, at the destination point, another transducer converts the electrical waveform to the appropriate message.

The transmitter is located at one point in space, the receiver is located at some other point separate from the transmitter, and the channel is the medium that provides the electrical connection between them.

The purpose of the transmitter is to transform the message signal produced by the source of information into a form suitable for transmission over the channel.

The received signal is normally corrupted version of the transmitted signal, which is due to channel imperfections, noise and interference from other sources. The
receiver has the task of operating on the received signal so as to reconstruct a recognizable form of the original message signal and to deliver it to the user destination.

Communication Systems are divided into 3 categories:

1. Analog Communication Systems are designed to transmit analog information using analog modulation methods.

2. Digital Communication Systems are designed for transmitting digital information using digital modulation schemes, and

3. Hybrid Systems that use digital modulation schemes for transmitting sampled and quantized values of an analog message signal.

**ELEMENTS OF DIGITAL COMMUNICATION SYSTEMS:**

The figure below shows the functional elements of a digital communication system.

**Source of Information:**

1. Analog Information Sources.
2. Digital Information Sources.

**Analog Information Sources** → Microphone actuated by a speech, TV Camera scanning a scene, continuous amplitude signals.

**Digital Information Sources** → These are teletype or the numerical output of computer which consists of a sequence of discrete symbols or letters.

An Analog information is transformed into a discrete information through the process of sampling and quantizing.
DIGITAL COMMUNICATION SYSTEM

Fig: Block Diagram of a Digital Communication System

SOURCE ENCODER / DECODER:

The Source encoder (or Source coder) converts the input i.e. symbol sequence into a binary sequence of 0’s and 1’s by assigning code words to the symbols in the input sequence. For eg. :- If a source set is having hundred symbols, then the number of bits used to represent each symbol will be 7 because \(2^7 = 128\) unique combinations are available. The important parameters of a source encoder are block size, code word lengths, average data rate and the efficiency of the coder (i.e. actual output data rate compared to the minimum achievable rate).

At the receiver, the source decoder converts the binary output of the channel decoder into a symbol sequence. The decoder for a system using fixed-length code words is quite simple, but the decoder for a system using variable-length code words will be very complex.

Aim of the source coding is to remove the redundancy in the transmitting information, so that bandwidth required for transmission is minimized. Based on the probability of the symbol code word is assigned. Higher the probability, shorter is the codeword.
Ex: Huffman coding.

CHANNEL ENCODER / DECODER:

Error control is accomplished by the channel coding operation that consists of systematically adding extra bits to the output of the source coder. These extra bits do not
convey any information but helps the receiver to detect and/or correct some of the errors in
the information bearing bits.

There are two methods of channel coding:

1. **Block Coding:** The encoder takes a block of \( k \) information bits from the source encoder and adds \( r \) error control bits, where \( r \) is dependent on \( k \) and error control capabilities desired.

2. **Convolution Coding:** The information bearing message stream is encoded in a continuous fashion by continuously interleaving information bits and error control bits.

The Channel decoder recovers the information bearing bits from the coded binary stream. Error detection and possible correction is also performed by the channel decoder.

The important parameters of coder/decoder are: Method of coding, efficiency, error control capabilities and complexity of the circuit.

**MODULATOR:**

The Modulator converts the input bit stream into an electrical waveform suitable for transmission over the communication channel. Modulator can be effectively used to minimize the effects of channel noise, to match the frequency spectrum of transmitted signal with channel characteristics, to provide the capability to multiplex many signals.

**DEMODULATOR:**

The extraction of the message from the information bearing waveform produced by the modulation is accomplished by the demodulator. The output of the demodulator is bit stream. The important parameter is the method of demodulation.

**CHANNEL:**

The Channel provides the electrical connection between the source and destination. The different channels are: Pair of wires, Coaxial cable, Optical fibre, Radio channel, Satellite channel or combination of any of these.

The communication channels have only finite Bandwidth, non-ideal frequency response, the signal often suffers amplitude and phase distortion as it travels over the channel. Also, the signal power decreases due to the attenuation of the channel. The signal is corrupted by unwanted, unpredictable electrical signals referred to as noise.
9. **Output Transducer:**
Finally we get the desired signal in desired format analog or digital.

**Advantages of digital communication:**

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

**Disadvantages**

- TDM digital transmission is not compatible with the FDM
- A Digital system requires large bandwidth.
Fig. 2 Conversion of Analog Signal to Digital Signal

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

Elements of PCM System:

- Sampling:
  - Process of converting analog signal into discrete signal.
  - Sampling is common in all pulse modulation techniques
• The signal is sampled at regular intervals such that each sample is proportional to amplitude of signal at that instant
• Analog signal is sampled every $T_s$ Secs, called sampling interval. $f_s=1/T_s$ is called sampling rate or sampling frequency.
• $f_s=2f_m$ is Min. sampling rate called Nyquist rate. Sampled spectrum ($\omega$) is repeating periodically without overlapping.
• Original spectrum is centered at $\omega=0$ and having bandwidth of $\omega_m$. Spectrum can be recovered by passing through low pass filter with cut-off $\omega_m$.
• For $f_s<2f_m$ sampled spectrum will overlap and cannot be recovered back. This is called aliasing.

**Sampling methods:**

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

![Types of Sampling](https://via.placeholder.com/150)

---

**Sampling of band-pass Signals:**

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

  Min. sampling rate $=2 \times \text{Bandwidth}$

  $=2 \times 2f_m = 4f_m$

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

**Instantaneous Sampling or Impulse Sampling:**

- Sampling function is train of spectrum remains constant impulses throughout frequency range. It is not practical.
Natural sampling:
- The spectrum is weighted by a sinc function.
- Amplitude of high frequency components reduces.

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

Sampling Theorem:

Statement of sampling theorem

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

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

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

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

\[
f_s \geq 2W
\]

Here \( f_s \) is the sampling frequency and

\( W \) is the higher frequency content

Fig. 5  CT and Its DT signal

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

Comments:

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

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

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

![PCM Generator Diagram](image)

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

\[
f_s \geq 2W
\]

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

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

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

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

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

\[
q = 2^v \quad \ldots \quad 1
\]

Here \( q \) represents total number of digital levels of \( q \)-level quantizer.

For example if \( v = 3 \) bits, then total number of levels will be,

\[
q = 2^3 = 8 \text{ levels}
\]

Each sample is converted to \( v \) binary bits. i.e. Number of bits per sample = \( v \)

We know that, Number of samples per second = \( f_s \)

\[
\therefore \text{ Number of bits per second is given by,}
\]

\[
\begin{align*}
\text{(Number of bits per second)} & = \text{(Number of bits per samples)} \\
& \times \text{(Number of samples per second)} \\
& = v \text{ bits per sample} \times f_s \text{ samples per second} \quad \ldots \quad 2
\end{align*}
\]

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

\[
\text{Signaling rate in PCM : } r = v f_s \quad \ldots \quad 3
\]

Here \( f_s \geq 2W \).

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

\[
B_T \geq \frac{1}{2} r \quad \ldots \quad 4
\]

Transmission Bandwidth of PCM:

\[
\begin{align*}
B_T & \geq \frac{1}{2} v f_s \quad \text{Since } f_s \geq 2W \\
B_T & \geq v W
\end{align*} \quad \ldots \quad 5
\]

\[
B_T \geq v W \quad \ldots \quad 6
\]
PCM Receiver:

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

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

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

Quantization

- The quantizing of an analog signal is done by discretizing the signal with a number of quantization levels.
• **Quantization** is representing the sampled values of the amplitude by a finite set of levels, which means converting a continuous-amplitude sample into a discrete-time signal.

• Both sampling and quantization result in the loss of information.

• The quality of a Quantizer output depends upon the number of quantization levels used.

• The discrete amplitudes of the quantized output are called as **representation levels** or **reconstruction levels**.

• The spacing between the two adjacent representation levels is called a **quantum** or **step-size**.

• There are two types of Quantization
  - Uniform Quantization
  - Non-uniform Quantization.

• The type of quantization in which the quantization levels are uniformly spaced is termed as a **Uniform Quantization**.

• The type of quantization in which the quantization levels are unequal and mostly the relation between them is logarithmic, is termed as a **Non-uniform Quantization**.

**Uniform Quantization:**

• There are two types of uniform quantization.
  - Mid-Rise type
  - Mid-Tread type.

• The following figures represent the two types of uniform quantization.

![Fig 1: Mid-Rise type Uniform Quantization](image1)

![Fig 2: Mid-Tread type Uniform Quantization](image2)

• The **Mid-Rise** type is so called because the origin lies in the middle of a raising part of the stair-case like graph. The quantization levels in this type are even in number.

• The **Mid-tread** type is so called because the origin lies in the middle of a tread of the stair-case like graph. The quantization levels in this type are odd in number.

• Both the mid-rise and mid-tread type of uniform quantizer is symmetric about the origin.
Quantization Noise and Signal to Noise ratio in PCM System:

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

Step 1 : Quantization Error

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

\[ e = x_q(nT_s) - x(nT_s) \] ..................... (1)

Step 2 : Step size

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

Therefore the total amplitude range becomes,

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

If this amplitude range is divided into \( q \) levels of quantizer, then the step size \( \delta \) is given as,

\[ \delta = \frac{x_{\text{max}} - (-x_{\text{max}})}{q} \]

\[ = \frac{2x_{\text{max}}}{q} \] ..........................(3)

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

\[ x_{\text{max}} = 1 \]

\[ -x_{\text{max}} = -1 \] ..........................(4)

Therefore step size will be,

\[ \delta = \frac{2}{q} \] (for normalized signal) ..........................(5)

Step 3 : Pdf of Quantization error

If step size \( \delta \) is sufficiently small, then it is reasonable to assume that the quantization error \( e \) will be uniformly distributed random variable. The maximum quantization error is given by

\[ e_{\text{max}} = \left| \frac{\delta}{2} \right| \] ..........................(6)

i.e. \[ -\frac{\delta}{2} \geq e_{\text{max}} \geq \frac{\delta}{2} \] ..........................(7)
Thus over the interval \( \left( -\frac{\delta}{2}, \frac{\delta}{2} \right) \) quantization error is uniformly distributed random variable.

\[
\begin{align*}
\text{(a)} & \quad f_X(x) = \begin{cases} 
0 & \text{for } x \leq a \\
\frac{1}{b-a} & \text{for } a < x \leq b \\
0 & \text{for } x > b
\end{cases} \\
\text{(b)} & \quad f_e(e) = \begin{cases} 
1 & \text{for } -\frac{\delta}{2} < e \leq \frac{\delta}{2} \\
0 & \text{for } e > \frac{\delta}{2}
\end{cases}
\end{align*}
\]

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

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

\[
f_X(x) = \begin{cases} 
0 & \text{for } x \leq a \\
\frac{1}{b-a} & \text{for } a < x \leq b \\
0 & \text{for } x > b
\end{cases} \quad \text{(8)}
\]

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

\[
f_e(e) = \begin{cases} 
0 & \text{for } e \leq \frac{\delta}{2} \\
\frac{1}{\delta} & \text{for } -\frac{\delta}{2} < e \leq \frac{\delta}{2} \\
0 & \text{for } e > \frac{\delta}{2}
\end{cases} \quad \text{(9)}
\]
Step 4: Noise Power

Quantization error \( \varepsilon \) has zero average value.

That is mean 'm_e' of the quantization error is zero.

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

\[
\frac{S}{N} = \frac{\text{Signal power (normalized)}}{\text{Noise power (normalized)}} \quad \ldots 10
\]

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

The noise power is given as,

\[
\text{Noise power} = \frac{V_{\text{noise}}^2}{R} \quad \ldots (11)
\]

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

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

The mean square value of a random variable \( X \) is given as,

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

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

From equation 9, we can write above equation as,

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

\[
= \frac{1}{\delta} \left[ \frac{8^3}{8} \right]_{-\delta/2}^{\delta/2} = \frac{1}{8} \left[ \frac{(\delta/2)^3}{3} + \frac{(\delta/2)^3}{3} \right]
\]

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

\[\therefore \text{From equation 1.8.25, the mean square value of noise voltage is,} \]

\[
V_{\text{noise}}^2 = \text{mean square value} = \frac{\delta^2}{12}
\]
When load resistance, \( R = 1 \) ohm, then the noise power is normalized i.e.,

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

\[
= \frac{\delta^2}{12} = \frac{\delta^2}{12}
\]

Thus we have,

**Normalized noise power**

\[
\text{or Quantization noise power} = \frac{\delta^2}{12}, \quad \text{For linear quantization}.
\]

**or Quantization error (in terms of power)**

\[\text{... (16)}\]

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

signal to quantization noise ratio is given as,

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

\[
= \frac{\text{Normalized signal power}}{(\delta^2 / 12)} \quad \text{... (17)}
\]

The number of bits '\( q \)' and quantization levels 'q' are related as,

\[q = 2^q\]

**Putting this value in equation (3) we have,**

\[
\delta = \frac{2x_{\text{max}}}{2^q} \quad \text{... (19)}
\]

**Putting this value in equation 1.8.30 we get,**

\[
\frac{S}{N} = \frac{\text{Normalized signal power}}{(\frac{2x_{\text{max}}}{2^q})^2 + 12}
\]

Let normalized signal power be denoted as 'P'.

\[
\frac{S}{N} = \frac{P}{4x_{\text{max}}^2 \times \frac{1}{12}} = \frac{3P}{x_{\text{max}}^2 \cdot 2^{2q}}
\]
Non-Uniform Quantization:

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

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

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

\[ Z(x) = \left( \text{Sgn } x \right) \frac{\ln(1 + \mu |x|)}{\ln(1 + \mu)} \quad |x| \leq 1 \quad \ldots \quad (1) \]

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

![Graph showing signal to noise ratio (SNR) with and without companding](image)

**Fig. 11** PCM performance with μ-law companding
It can be observed from above figure that signal to noise ratio of PCM remains almost constant with companding.

**A-Law for Companding**

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

\[
Z(x) = \begin{cases} 
\frac{A|x|}{1+lnA} & \text{for } 0 \leq |x| \leq \frac{1}{A} \\
\frac{1+ln(A|x|)}{1+lnA} & \text{for } \frac{1}{A} \leq |x| \leq 1 
\end{cases} \quad \ldots (2)
\]

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

**Signal to Noise Ratio of Companded PCM**

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

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

Here \( q = 2^\mu \) is number of quantization levels.

---

**Differential Pulse Code Modulation (DPCM):**

Redundant Information in PCM:

The samples of a signal are highly correlated with each other. This is because any signal does not change fast. That is its value from present sample to next sample does not differ by large amount. The adjacent samples of the signal carry the same information with little difference. When these samples are encoded by standard PCM system, the resulting encoded signal contains redundant information.
Fig. shows a continuous time signal \( x(t) \) by dotted line. This signal is sampled by flat top sampling at intervals \( T_s, 2T_s, 3T_s \ldots nT_s \). The sampling frequency is selected to be higher than nyquist rate. The samples are encoded by using 3 bit (7 levels) PCM. The sample is quantized to the nearest digital level as shown by small circles in the diagram. The encoded binary value of each sample is written on the top of the samples. We can see from Fig. that the samples taken at \( 4T_s, 5T_s \) and \( 6T_s \) are encoded to same value of (110). This information can be carried only by one sample. But three samples are carrying the same information means it is redundant. Consider another example of samples taken at \( 9T_s \) and \( 10T_s \). The difference between these samples is only due to last bit and first two bits are redundant, since they do not change.

Fig. Redundant information in PCM
**Principle of DPCM**

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

**DPCM Transmitter**

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

\[
e(nT_s) = x(nT_s) - \hat{x}(nT_s)
\]

\[\text{(1)}\]

![DPCM Transmitter Diagram](image)

Fig. Differential pulse code modulation transmitter

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

The quantizer output can be written as,

\[
e_q(nT_s) = e(nT_s) + q(nT_s) \tag{2}
\]

Here \( q(nT_s) \) is the quantization error. As shown in Fig. 1, the prediction filter input \( x_q(nT_s) \) is obtained by sum \( \hat{x}(nT_s) \) and quantizer output i.e.,

\[
x_q(nT_s) = \hat{x}(nT_s) + e_q(nT_s) \tag{3}
\]

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

\[
x_q(nT_s) = \hat{x}(nT_s) + e(nT_s) + q(nT_s) \tag{4}
\]

Equation 1 is written as,

\[
e(nT_s) = x(nT_s) - \hat{x}(nT_s)
\]

\[∴ e(nT_s) + \hat{x}(nT_s) = x(nT_s) \tag{5}\]

\[∴\text{Putting the value of } e(nT_s) + \hat{x}(nT_s) \text{ from above equation into equation } 4 \text{ we get},
\]

\[
x_q(nT_s) = x(nT_s) + q(nT_s) \tag{6}
\]

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

Reconstruction of DPCM Signal

Fig. 1 shows the block diagram of DPCM receiver.

![DPCM Receiver Diagram](image)

Fig. 1 DPCM receiver

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

**Line Coding:**

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

The waveform pattern of voltage or current used to represent the 1s and 0s of a digital signal on a transmission link is called line encoding. The common types of line coding are unipolar, polar, bipolar and Manchester encoding.

Line codes are used commonly in computer communication networks over short distances.
Fig. 3.2.1 Various digital PAM signals formats
(a) Unipolar RZ (b) Unipolar NRZ (c) Polar RZ (d) Polar NRZ (e) Bipolar NRZ (f) Split phase manchester (g) Polar quaternary NRZ
**Unipolar RZ (Return to zero):**

In unipolar format the wave form the wave form have a single polarity.

In the unipolar RZ form, the waveform has zero value when symbol ‘0’ is transmitted and waveform has ‘A’ volts when ‘1’ is transmitted. In RZ form, the ‘A’ volts is present for $T_b/2$ period if symbol ‘1’ is transmitted and for remaining $T_b/2$ waveform returns to zero value.

**Unipolar NRZ (Non Return to zero):**

In the unipolar NRZ, when symbol ‘1’ is to transmitted, the signal has ‘A’ volts for full duration. When symbol ‘0’ is to be transmitted, the signal has zero volts (no signal) for complete symbol duration.

**Polar RZ (Return to zero):**

In the polar form the representation in bipolar.

In the Polar RZ format, symbol ‘1’ is represented by positive voltage polarity and symbol ‘0’ is represented by negative polarity. Since this is RZ format, the pulse is transmitted only for half duration.

**Polar NRZ (Non Return to zero):**

In the Polar NRZ format, symbol ‘1’ is represented by positive voltage polarity and symbol ‘0’ is represented by negative polarity. These polarities are maintained over the complete pulse duration.

**Bipolar NRZ:**

In this format successive 1’s are represented by pulses with alternate polarity and 0’s are represented by no pulses.

**Split Phase Manchester:**

Here if symbol ‘1’ is to be transmitted, then a positive half interval pulse is followed by a negative half interval pulse. If symbol ‘0’ is to be transmitted, then a negative half interval pulse is followed by a positive half interval pulse.

**Polar Quaternary NRZ:**

This format is derived to reduce the signalling rate ‘r’. The message bits are grouped in the blocks of two. Therefore there are four possible combinations 00, 01, 10 and 11. To these four combinations, four amplitude levels are assigned.

<table>
<thead>
<tr>
<th>Message Combination</th>
<th>$X(t)=a_n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>-3A/2</td>
</tr>
<tr>
<td>01</td>
<td>-A/2</td>
</tr>
<tr>
<td>10</td>
<td>A/2</td>
</tr>
<tr>
<td>11</td>
<td>3A/2</td>
</tr>
</tbody>
</table>
Adaptive Differential pulse Code Modulation (ADPCM):

**Principle:** The ADPCM uses adaptive quantizer that has time varying step size. This step size depends upon amplitude level and spectrum of the speech signal.

**Implementation:**
- The adaptive quantization can be implemented with forward estimation or backward estimation. In forward estimation, unquantized sample of the input signal are used to obtain step size.
- In backward estimation samples of quantizer output are used to obtain the step size.
- Fig (a) shows the block diagram of ADPCM encoder. Observe that the above block diagram is similar to that of DPCM transmitter. There is an addition of one block adaptive prediction.
- The Adaptive predictor generates step size from \( x_q(nT_s) \) and \( e_q(nT_s) \). The prediction filter accordingly generates an estimate of input signal.

![ADPCM Encoder](image)

![ADPCM Decoder](image)

**Advantages:**
- The bit rate required by PCM is reduced to half of its earlier value.
- Because of backward estimation the problems of level estimation, delay and requirement of buffer are eliminated.
- Efficient for speech coding at low bit rates.

**Disadvantages:**
- Secure transmission over radio channels.
- Voice coding at 32 kbps.
Time Division Multiplexing:

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

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

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

The TDM system is highly sensitive to dispersion in the common channel, that is, to variations of amplitude with frequency or lack of proportionality of phase with frequency. Accordingly, accurate equalization of both magnitude and phase responses of the channel is necessary to ensure a satisfactory operation of the system;
TDM is immune to nonlinearities in the channel as a source of crosstalk. The reason for this behaviour is that different message signals are not simultaneously applied to the channel.

**Introduction to Delta Modulation**

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

**Delta Modulation**

**Operating Principle of DM**

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

\[
e(nT_s) = x(nT_s) - \hat{x}(nT_s)
\] ... (1)

Here, \( e(nT_s) \) = Error at present sample  
\( x(nT_s) \) = Sampled signal of \( x(t) \)  
\( \hat{x}(nT_s) \) = Last sample approximation of the staircase waveform.

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

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

\( = \) Last sample approximation of staircase waveform.

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

\[
b(nT_s) = \delta \text{ sgn} [e(nT_s)]
\] ... (3)

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

\[
b(nT_s) = +\delta \quad \text{if} \quad x(nT_s) \geq \hat{x}(nT_s) \\
= -\delta \quad \text{if} \quad x(nT_s) < \hat{x}(nT_s)
\] ... (4)

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

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

\( T_s = \) Sampling interval.
**DM Transmitter**

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

The summer in the accumulator adds quantizer output (±δ) with the previous sample approximation. This gives present sample approximation, i.e.,

\[ u(nT_s) = u(nT_s - T_s) \pm \delta \]  

or

\[ = u[(n-1)T_s] + b(nT_s) \]  

... (5)

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

---

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

**DM Receiver**

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

**Advantages and Disadvantages of Delta Modulation**

**Advantages of Delta Modulation**

The delta modulation has following advantages over PCM,

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

**Disadvantages of Delta Modulation**

The delta modulation has two drawbacks:
Slope Overload Distortion (Startup Error)

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

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

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

Granular Noise (Hunting)

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

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

Adaptive Delta Modulation

Operating Principle

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

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

Transmitter and Receiver

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

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

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

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

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

Condition for Slope overload distortion occurrence:

Slope overload distortion will occur if

\[ A_m > \frac{\delta}{2\pi f_m T_s} \]

where \( T_s \) is the sampling period.

Let the sine wave be represented as,

\[ x(t) = A_m \sin (2\pi f_m t) \]

Slope of \( x(t) \) will be maximum when derivative of \( x(t) \) with respect to \( t \) will be maximum. The maximum slope of delta modulator is given

\[
\text{Max. slope} = \frac{\text{Step size}}{\text{Sampling period}} = \frac{\delta}{T_s}
\]

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

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

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

\[
\max |A_m 2\pi f_m \cos (2\pi f_m t)| > \frac{\delta}{T_s}
\]

\[
A_m 2\pi f_m > \frac{\delta}{T_s}
\]

or

\[ A_m > \frac{\delta}{2\pi f_m T_s} \]

\[ \text{..........(2)} \]
Expression for Signal to Quantization Noise power ratio for Delta Modulation:

To obtain signal power:

\[ A_m \leq \frac{\delta}{2\sqrt{f_m T_s}} \]

Here \( A_m \) is peak amplitude of sinusoided signal
\( \delta \) is the step size
\( f_m \) is the signal frequency and
\( T_s \) is the sampling period.

From above equation, the maximum signal amplitude will be,

\[ A_m = \frac{\delta}{2\sqrt{f_m T_s}} \]

...................(1)

Signal power is given as,

\[ P = \frac{V^2}{R} \]

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

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

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

\[ P = \frac{A_m^2}{2} \]

Putting for \( A_m \) from equation 1

\[ P = \frac{\delta^2}{8\pi^2 f_m^2 T_s^2} \]

...................(2)

This is an expression for signal power in delta modulation.

(ii) To obtain noise power

\[ f_e(e) \]

\[ \frac{1}{\delta - (-\delta)} = \frac{1}{2\delta} \]

\[ -\delta \quad \delta \]

\[ e \]

Fig. Uniform distribution of quantization error
\[ f_\varepsilon (\varepsilon) = \begin{cases} 
0 & \text{for } \varepsilon < \delta \\
\frac{1}{2\delta} & \text{for } -\delta < \varepsilon < \delta \\
0 & \text{for } \varepsilon > \delta 
\end{cases} \] 

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

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

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

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

From equation 3,

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

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

\[ = \frac{1}{2\delta} \left[ \frac{\delta^3}{3} + \frac{\delta^3}{3} \right] = \frac{\delta^2}{3} \]

\[ \text{Hence noise power will be,} \]

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

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

\[ \text{noise power} = \frac{\delta^2}{3} \]

\[ \text{.............(5)} \]
This noise power is uniformly distributed over \(-f_s\) to \(f_s\) range. This is illustrated in Fig.

At the output of delta modulator receiver there is lowpass reconstruction filter whose cutoff frequency is \(W\). This cutoff frequency is equal to highest signal frequency. The reconstruction filter passes part of the noise power at the output as Fig.

From the geometry of Fig., output noise power will be,

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

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

\[
\text{Output noise power} = \frac{WT_s \delta^2}{3}
\]

\[(6)\]

(iii) To obtain signal to noise power ratio

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

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

From equation 2. and equation 6

\[
\frac{S}{N} = \frac{8\pi^2f_n^2T_s^2}{WT_s \delta^2}
\]

\[
\frac{S}{N} = \frac{3}{8\pi^2f_n^2T_s^3}
\]

\[(7)\]

This is an expression for signal to noise power ratio in delta modulation.
Comparision of PCM, DM, ADM and DPCM:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Number of bits</td>
<td>It can use 4, 8 or 16 bits per sample.</td>
<td>It uses only one bit for one sample.</td>
<td>Only one bit is used to encode one sample.</td>
<td>Bits can be more than one but are less than PCM.</td>
</tr>
<tr>
<td>2</td>
<td>Levels, step size</td>
<td>The number of levels depend on number of bits. Level size is fixed.</td>
<td>Step size is fixed and cannot be varied.</td>
<td>According to the signal variation, step size varies (Adapted).</td>
<td>Fixed number of levels are used.</td>
</tr>
<tr>
<td>3</td>
<td>Quantization error and distortion</td>
<td>Quantization error depends on number of levels used.</td>
<td>Slope overload distortion and granular noise is present.</td>
<td>Quantization error is present but other errors are absent.</td>
<td>Slope overload distortion and quantization noise is present.</td>
</tr>
<tr>
<td>4</td>
<td>Bandwidth of transmission channel</td>
<td>Highest bandwidth is required since number of bits are high.</td>
<td>Lowest bandwidth is required.</td>
<td>Lowest bandwidth is required.</td>
<td>Bandwidth required is lower than PCM.</td>
</tr>
<tr>
<td>5</td>
<td>Feedback</td>
<td>There is no feedback in transmitter or receiver.</td>
<td>Feedback exists in transmitter.</td>
<td>Feedback exists.</td>
<td>Feedback exists.</td>
</tr>
<tr>
<td>7</td>
<td>Signal to noise ratio</td>
<td>Good.</td>
<td>Poor.</td>
<td>Better than DM.</td>
<td>Fair.</td>
</tr>
<tr>
<td>8</td>
<td>Area of applications</td>
<td>Audio and video Telephony</td>
<td>Speech and images.</td>
<td>Speech and images.</td>
<td>Speech and video.</td>
</tr>
</tbody>
</table>

Next part shows the comparison for voice encoding.

<table>
<thead>
<tr>
<th>Sr.No.</th>
<th>Parameter</th>
<th>PCM</th>
<th>DM</th>
<th>ADM</th>
<th>DPCM</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>Sampling rate kHz</td>
<td>8</td>
<td>64-128</td>
<td>48-54</td>
<td>8</td>
</tr>
<tr>
<td>10</td>
<td>Bits/sample</td>
<td>7 - 8</td>
<td>1</td>
<td>1</td>
<td>4-6</td>
</tr>
<tr>
<td>11</td>
<td>Bit rate</td>
<td>56-64</td>
<td>64-128</td>
<td>46-54</td>
<td>32-48</td>
</tr>
</tbody>
</table>

For all Problems please refer given notebook
UNIT 2

BASEBAND PULSE TRANSMISSION
INTRODUCTION

In signal processing, a matched filter is obtained by correlating a known signal, or template, with an unknown signal to detect the presence of the template in the unknown signal.\[1\][2] This is equivalent to convolving the unknown signal with a conjugated time-reversed version of the template. The matched filter is the optimal linear filter for maximizing the signal to noise ratio (SNR) in the presence of additive stochastic noise. Matched filters are commonly used in radar, in which a known signal is sent out, and the reflected signal is examined for common elements of the outgoing signal. Pulse compression is an example of matched filtering. It is so called because impulse response is matched to input pulse signals. Two-dimensional matched filters are commonly used in image processing, e.g., to improve SNR for X-ray. Matched filtering is a demodulation technique with LTI (linear time invariant) filters to maximize SNR.\[3\] It was originally also known as a North filter.

MATCHED FILTER

Science each of the orthonormal basic functions are $\Phi_1(t), \Phi_2(t), \ldots, \Phi_M(t)$ is assumed to be zero outside the interval $0 < t < T$. We can design a linear filter with impulse response $h(t)$, with the received signal $x(t)$ the fitter output is given by the convolution integral

$$y(t) = \int_{-\infty}^{\infty} x(t) \ast h(t) \, dt$$

where $x_j$ is the $j^{th}$ correlator output produced by the received signal $x(t)$.

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

For a matched filter operating in real time to be physically realizable, it must be causal. For causal system
**Φ(t)** = input signal  
**h(t)** = impulse response  
**W(t)** = white noise

The impulse response of the matched filter is time-reversed and delayed version of the input signal.

**MATCHED FILTER PROPERTIES**

**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.

**PROPERTY 2**
The output signal of a Matched Filter is proportional to a shifted version of the autocorrelation function of the input signal to which the filter is matched.

**PROPERTY 3**
The output Signal to Noise Ratio of a Matched filter depends only on the ratio of the signal energy to the power spectral density of the white noise at the filter input.

**PROPERTY 4**

The Matched Filtering operation may be separated into two matching conditions; namely spectral phase matching that produces the desired output peak at time $T$, and the spectral amplitude matching that gives this peak value its optimum signal to noise density ratio.

**MATCHED FILTER FOR RECTANGULAR PULSE**

- **Matched filter for causal rectangular pulse shape**
  
  Impulse response is causal rectangular pulse of same duration

- **Convolve input with rectangular pulse of duration $T$ sec and sample result at $T$ sec is same as**
  
  First, integrate for $T$ sec
  
  Second, sample at symbol period $T$ sec
  
  Third, reset integration for next time period

- **Integrate and dump circuit**

  It is well-known that for a radar waveform $x(t)$ in additive white noise of power $\sigma^2$, the optimum receiver frequency response, in the sense of maximizing the signal-to-noise ratio (SNR) in the receiver output signal $y(t)$ at a particular instant $T$, is the *matched filter* impulse response. The corresponding frequency response is $\beta$. The constant $\alpha$ is arbitrary; it affects only the overall gain, but has no impact on the SNR achieved or the shape of the receiver frequency response. Henceforth we will assume $\alpha$ has been chosen to make the peak gain of equal to 1.

  The definition of SNR is the square of the peak signal voltage at $t = T$, divided by the output noise power obtained by integrating the filtered noise power spectral density over all frequency:
When the matched filter is used, the peak SNR achieved can be shown to be , where $E$ is the signal energy

If $x(t)$ is a simple rectangular pulse of unit amplitude and duration $\tau$, $E = \tau$. The problem of interest here is what happens when the signal is a rectangular pulse, but the receiver filter is, as may be more likely, a bandpass filter (BPF) with cutoff frequency $\pm B$ Hz. As the BPF cutoff $B$ increases from zero, noise is added linearly to the output. However, signal energy is not added linearly. As $B$ increases, the energy in the mainlobe of the pulse’s sinc-shaped spectrum increases signal output energy fairly rapidly, but after a point, only the sinc sidelobes are passed to the output, and the rate of signal energy growth will slow. Consequently, there is some choice of $B$ that produces the maximum SNR at the output. A narrower cutoff filters out too much signal energy; a wider cutoff allows too much noise energy to pass. The goal of this memo is to determine this optimum BPF cutoff $B_{\text{opt}}$ analytically, and to confirm the results via simulation.

**ERROR RATE DUE TO NOISE**

- To proceed with the analysis, consider a binary PCM system based on polar nonreturn-to-zero (NRZ) signaling.
- Symbols 1 and 0 are represented by positive and negative rectangular pulses of equal amplitude and equal duration.
- The channel noise is modeled as additive white Gaussian noise $w(t)$ of zero mean and power spectral density $N_0/2$; the Gaussian assumption is needed for later calculations. In the signaling interval $0 < t < T_b$ the received signal written as:

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

where $T_b$ is the bit duration, and $A$ is the transmitted pulse amplitude.

- The receiver has acquired knowledge of the starting and ending times of each transmitted pulse;

- Given the noisy signal $x(t)$, the receiver is required to make a decision in each signaling interval as to whether the transmitted symbol is a 1 or a 0.
The structure of the receiver used to perform this decision-making process is shown in Figure. It consists of a matched filter followed by a sampler, and then finally a decision device.

![Diagram of receiver](image)

**FIGURE 3.3 Receiver for baseband transmission of binary-encoded PCM wave using polar NRZ signaling.**

The filter is matched to a rectangular pulse of amplitude $A$ and duration $T_b$, exploiting the bit-timing information available to the receiver. The resulting matched filter output is sampled at the end of each signaling interval.

The presence of channel noise $w(t)$ adds randomness to the matched filter output.

The sample value $y$ is compared to a preset threshold $\lambda$ in the decision device. If the threshold is exceeded, the receiver makes a decision in favor of symbol 1; if not, a decision is made in favor of symbol 0.

There are two possible kinds of error to be considered:
1. Symbol 1 is chosen when a 0 was actually transmitted; we refer to this error as an *error of the first kind*.
2. Symbol 0 is chosen when a 1 was actually transmitted; we refer to this error as an *error of the second kind*.

**INTER SYMBOL INTERFERENCE**

Generally, digital data is represented by electrical pulse, communication channel is always band limited. Such a channel disperses or spreads a pulse carrying digitized samples passing through it. When the channel bandwidth is greater than bandwidth of pulse, spreading of pulse is very less. But when channel bandwidth is close to signal bandwidth, i.e. if we transmit digital data which demands more bandwidth which exceeds channel bandwidth, spreading will occur and
cause signal pulses to overlap. This overlapping is called **Inter Symbol Interference**. In short it is called **ISI**. Similar to interference caused by other sources, ISI causes degradations of signal if left uncontrolled.

This problem of ISI exists strongly in Telephone channels like coaxial cables and optical fibers. In this chapter main objective is to study the effect of ISI, when digital data is transmitted through band limited channel and solution to overcome the degradation of waveform by properly shaping pulse.

The effect of sequence of pulses transmitted through channel is shown in fig. The Spreading of pulse is greater than symbol duration, as a result adjacent pulses interfere. i.e. pulses get completely smeared, tail of smeared pulse enter into adjacent symbol intervals making it difficult to decide actual transmitted pulse.

First let us have look at different formats of transmitting digital data. In base band transmission best way is to map digits or symbols into pulse waveform. This waveform is generally termed as **Line codes**.

**NYQUIST ISI CRITERION**

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

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

When consecutive symbols are transmitted over a channel by a linear modulation (such as ASK, QAM, etc.), the impulse response (or equivalently the frequency response) of the channel causes a transmitted symbol to be spread in the time domain. This causes intersymbol interference because the previously transmitted symbols affect the currently received symbol,
thus reducing tolerance for noise. The Nyquist theorem relates this time-domain condition to
an equivalent frequency-domain condition.

The Nyquist criterion is closely related to the Nyquist-Shannon sampling theorem, with only a
differing point of view.

This criterion can be intuitively understood in the following way: frequency-shifted replicas of
H(f) must add up to a constant value.

In practice this criterion is applied to baseband filtering by regarding the symbol sequence as
weighted impulses (Dirac delta function). When the baseband filters in the communication
system satisfy the Nyquist criterion, symbols can be transmitted over a channel with flat
response within a limited frequency band, without ISI. Examples of such baseband filters are
the raised-cosine filter, or the sinc filter as the ideal case.

**Derivation**

To derive the criterion, we first express the received signal in terms of the transmitted symbol
and the channel response. Let the function h(t) be the channel impulse response, x[n] the
symbols to be sent, with a symbol period of Ts; the received signal y(t) will be in the form
(where noise has been ignored for simplicity):

only one transmitted symbol has an effect on the received y[k] at sampling instants, thus
removing any ISI. This is the time-domain condition for an ISI-free channel. Now we find a
frequency-domain equivalent for it. We start by expressing this condition in continuous time:

This is the Nyquist ISI criterion and, if a channel response satisfies it, then there is no ISI
between the different samples.

**RAISED COSINE SPECTRUM**

We may overcome the practical difficulties encountered with the ideal Nyquist channel by
extending the bandwidth from the minimum value $W = R_b/2$ to an adjustable value between
W and 2 W. We now specify the frequency function $P(f)$ to satisfy a condition more elaborate
than that for the ideal Nyquist channel; specifically, we retain three terms and restrict the
frequency band of interest to $[-W, W]$, as shown by
We may devise several band-limited functions to satisfy (1). A particular form of \( P(f) \) that embodies many desirable features is provided by a raised cosine spectrum. This frequency characteristic consists of a flat portion and a rolloff portion that has a sinusoidal form, as follows:

\[
P(f) = \begin{cases} 
\frac{1}{2W} & \text{for } 0 \leq |f| < f_1 \\
\frac{1}{4W} \left( 1 - \sin \frac{\pi (|f| - W)}{2W - 2f_1} \right) & \text{for } f_1 \leq |f| < 2W - f_1 \\
0 & \text{for } |f| \geq 2W - f_1 
\end{cases}
\]

The frequency parameter \( f_1 \) and bandwidth \( W \) are related by

\[
\alpha = 1 - \frac{f_1}{W}
\]

The parameter alpha is called the rolloff factor; it indicates the excess bandwidth over the ideal solution, \( W \). Specifically, the transmission bandwidth \( B_T \) is defined by \( 2W - f_1 = W(1 + \alpha) \).

The frequency response \( P(f) \), normalized by multiplying it by 2 \( W \), is shown plotted in Fig. 1 for three values of alpha, namely, 0, 0.5, and 1. We see that for alpha = 0.5 or 1, the function \( P(f) \) cuts off gradually as compared with the ideal Nyquist channel (i.e., alpha = 0) and is therefore easier to implement in practice. Also the function \( P(f) \) exhibits odd symmetry with respect to the Nyquist bandwidth \( W \), making it possible to satisfy the condition of (1). The time response \( p(t) \) is the inverse Fourier transform of the function \( P(f) \). Hence, using the \( P(f) \) defined in (2), we obtain the result

\[
p(t) = \text{sinc}(2Wt) \left( \frac{\cos 2\pi \alpha W t}{1 - 16\alpha^2 W^2 t^2} \right)
\]
which is shown plotted in Fig. 2 for alpha = 0, 0.5, and 1. The function p(t) consists of the product of two factors: the factor \( \text{rm sinc}(2 W t) \) characterizing the ideal Nyquist channel and a second factor that decreases as 1/\( \vert t \vert^2 \) for large \( \vert t \vert \). The first factor ensures zero crossings of p(t) at the desired sampling instants of time \( t = iT \) with \( i \) an integer (positive and negative). The second factor reduces the tails of the pulse considerably below that obtained from the ideal Nyquist channel, so that the transmission of binary waves using such pulses is relatively insensitive to sampling time errors. In fact, for alpha = 1, we have the most gradual rolloff in that the amplitudes of the oscillatory tails of p(t) are smallest. Thus, the amount of intersymbol interference resulting from timing error decreases as the rolloff factor alpha is increased from zero to unity.

![Graph showing p(t) for alpha = 0, 0.5, and 1](image)

The special case with alpha = 1 (i.e., \( f_1 = 0 \)) is known as the full-cosine rolloff characteristic, for which the frequency response of (2) simplifies to

\[
P(f) = \begin{cases} 
\frac{1}{4W} \left( 1 + \cos \frac{\pi f}{2W} \right) & \text{for } 0 < |f| < 2W \\
0 & \text{if } |f| \geq 2W
\end{cases}
\]

Correspondingly, the time response p(t) simplifies to
The time response exhibits two interesting properties:

These two properties are extremely useful in extracting a timing signal from the received signal for the purpose of synchronization. However, the price paid for this desirable property is the use of a channel bandwidth double that required for the ideal Nyquist channel corresponding to $\alpha = 0$.

\[ p(t) = \frac{\text{sinc}(4Wt)}{1 - 16W^2t^2} \]

CORRELATIVE CODING – DUOBINARY SIGNALING

The condition for zero ISI (Inter Symbol Interference) is
\[ p(nT) = \begin{cases} 1, & n=0 \\ 0, & n\neq 0 \end{cases} \]

which states that when sampling a particular symbol (at time instant \( nT=0 \)), the effect of all other symbols on the current sampled symbol is zero.

As discussed in the previous article, one of the practical ways to mitigate ISI is to use partial response signaling technique (otherwise called as “correlative coding”). In partial response signaling, the requirement of zero ISI condition is relaxed as a controlled amount of ISI is introduced in the transmitted signal and is counteracted in the receiver side.

By relaxing the zero ISI condition, the above equation can be modified as,

\[ p(nT) = \begin{cases} 1, & n=0,1 \\ 0, & \text{otherwise} \end{cases} \]

which states that the ISI is limited to two adjacent samples. Here we introduce a controlled or “deterministic” amount of ISI and hence its effect can be removed upon signal detection at the receiver.

**Duobinary Signaling:**

The following figure shows the duobinary signaling scheme (click to enlarge).
Encoding Process:

1) $a_n = \text{binary input bit; } a_n \in \{0,1\}$. 
2) $b_n = \text{NRZ polar output of Level converter in the precoder and is given by,}$  

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

3) $y_n$ can be represented as

$$y_n = b_n + b_{n-1} = \begin{cases} 2d, & \text{if } a_n = a_{n-1} = 1 \\ 0, & \text{if } a_n \neq a_{n-1} \\ -2d, & \text{if } a_n = a_{n-1} = 0 \end{cases}$$

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

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

$$h(t) = \text{sinc}(t/T) + \text{sinc}(t-T/T)$$
Decoding Process:

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

\[ b^n = y_n - b^{n-1} \]

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

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

- If \( y_n < d \), then \( a^n = 1 \)
- If \( y_n > d \), then \( a^n = 0 \)
- If \( y_n = 0 \), randomly guess \( a^n \)

CORRELATIVE CODING – MODIFIED DUOBINARY SIGNALING

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

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

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

In the case of modified duobinary signaling, the above equation is modified as

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

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

**Modified Duobinary Signaling:**

The following figure shows the modified duobinary signaling scheme (click to enlarge).

![Modified DuoBinary Signaling](image)

**Encoding Process:**

1) \( a_n \) = binary input bit; \( a_n \in \{0,1\} \).
2) \( b_n = \) NRZ polar output of Level converter in the precoder and is given by,
bn={−d, if ak=0
      +d, if ak=1

where \( a_k \) is the precoded output (before level converter).

3) \( y_n \) can be represented as

\[
y_n = b_n + b_{n-2} =\begin{cases} 
2d, & \text{if } ak = ak-2 = 1 \\
0, & \text{if } ak \neq ak-2 \\
-2d, & \text{if } ak = ak-2 = 0
\end{cases}
\]

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

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

\[
h(t) = \text{sinc}(t/T) - \text{sinc}(t - 2T/T)
\]

**Decoding Process:**

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

\[ b^n = y^n - b^{n-2} \]

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

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

\[
\begin{align*}
\text{If } y_n < d, \quad & a^n = 0 \\
\text{If } y_n > d, \quad & a^n = 1 \\
\text{If } y_n = 0, \quad & \text{randomly guess } a^n
\end{align*}
\]

**PARTIAL RESPONSE SIGNALLING**

Partial response signalling (PRS), also known as correlative coding, was introduced for the first time in 1960s for high data rate communication [lender, 1960]. From a practical point of view, the background of this technique is related to the Nyquist criterion.

Assume a Pulse Amplitude Modulation (PAM), according to the Nyquist criterion, the highest possible transmission rate without Inter-symbol-interference (ISI) at the receiver over a channel with a bandwidth of \( W \) (Hz) is \( 2W \) symbols/sec.

**BASEBAND M-ARY PAM TRANSMISSION**

Up to now for binary systems the pulses have two possible amplitude levels. In a baseband M-ary PAM system, the pulse amplitude modulator produces \( M \) possible amplitude levels with \( M > 2 \). In an M-ary system, the information source emits a sequence of symbols from an alphabet that consists of \( M \) symbols. Each amplitude level at the PAM modulator output corresponds to a distinct symbol. The symbol duration \( T \) is also called as the signaling rate of the system, which is expressed as symbols per second or bauds.
Let’s consider the following quaternary (M=4) system. The symbol rate is 1/(2Tb), since each symbol consists of two bits.

The symbol duration $T$ of the $M$-ary system is related to the bit duration $T_b$ of the equivalent binary PAM system as $T = T_b \log_2 M$. For a given channel bandwidth, using $M$-ary PAM system, $\log_2 M$ times more information is transmitted than binary PAM system. The price we paid is the increased bit error rate compared to binary PAM system. To achieve the same probability of error as the binary PAM system, the transmit power in $M$-ary PAM system must be increased.

For $M$ much larger than 2 and an average probability of symbol error small compared to 1, the transmitted power must be increased by a factor of $M^2/\log_2 M$ compared to binary PAM system. The $M$-ary PAM transmitter and receiver is similar to the binary PAM transmitter and receiver. In transmitter, the $M$-ary pulse train is shaped by a transmit filter and transmitted through a channel which corrupts the signal with noise and ISI.

The received signal is passed through a receive filter and sampled at an appropriate rate in synchronism with the transmitter. Each sample is compared with preset threshold values and a decision is made as to which symbol was transmitted. Obviously, in $M$-ary system there are $M-1$ threshold levels which makes the system complicated.

**EYE DIAGRAMS**

The quality of digital transmission systems are evaluated using the bit error rate. Degradation of quality occurs in each process modulation, transmission, and detection. The eye pattern is experimental method that contains all the information concerning the degradation of quality. Therefore, careful analysis of the eye pattern is important in analyzing the degradation mechanism.

Eye patterns can be observed using an oscilloscope. The received wave is applied to the vertical deflection plates of an oscilloscope and the saw tooth wave at a rate equal to transmitted symbol rate is applied to the horizontal deflection plates, resulting display is eye pattern as it
resemble human eye.

- The interior region of eye pattern is called eye opening

![Eye pattern diagram](image)

We get superposition of successive symbol intervals to produce eye pattern as shown below.

- The width of the eye opening defines the time interval over which the received wave can be sampled without error from ISI.
- Optimum sampling time corresponds to the maximum eye opening.
- The height of the eye opening at a specified sampling time is a measure of the margin over channel noise. The sensitivity of the system to timing error is determined by the rate of closure of the eye as the sampling time is varied. Any non linear transmission distortion would reveal itself in an asymmetric or squinted eye.
- When the effect of ISI is excessive, traces from the upper portion of the eye pattern cross traces from lower portion with the result that the eye is completely closed.
Raised Cosine Eye Diagram

- The larger $\alpha$, the wider the opening.
- The larger $\alpha$, the larger bandwidth $(1+\alpha)/T_b$
- But smaller $\alpha$ will lead to larger errors if not sampled at the best sampling time which occurs at the center of the eye.

Example of eye pattern:
Binary-PAM Perfect channel (no noise and no ISI)

Example of eye pattern: Binary-PAM with noise no ISI
UNIT –III
SIGNAL SPACE ANALYSIS
INTRODUCTION

Space analysis provides a mathematically elegant and highly insightful tool for the study of digital signal transmission. Signal space analysis permits a general geometric framework for the interpretation of digital signaling that includes both baseband and bandpass signaling schemes. Furthermore, it provides an intuitive insight into the error performance and spectral efficiency characteristics of the various digital signaling schemes. Before introducing the Signal Space analysis technique, a brief review of digital transmission is necessary.

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

The AWGN channel is one of the simplest mathematical models for various physical communications channels. The received signal $r(t)$ is given by

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

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

The key to analyzing and understanding the performance of digital transmission is the realization that signals used in communications can be expressed and visualized graphically. Thus, we need to understand signal space concepts as applied to digital communications. Two entirely different signal sets can have the same geometric representation. The underlying geometry will determine the performance and the receiver structure.

GRAM SCHMITT ORTHOGONALIZATION (GSO) PROCEDURE

Suppose we are given a signal set

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

Find the orthogonal basis functions for this signal set

$$\{\phi_1(t), \ldots, \phi_K(t)\}$$
where $K \leq M$

**Step 1: Construct the First Basis Function**

Compute the energy in signal 1:

$$E_1 = \int_{-\infty}^{\infty} s_1^2(t) \, dt$$

The first basis function is just a normalized version of $s_1(t)$

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

$$s_1(t) = s_{11} \phi_1(t) = \sqrt{E_1} \phi_1(t)$$

$$s_{11} = \int_{-\infty}^{\infty} s_1(t) \phi_1(t) \, dt = \sqrt{E_1}$$

**Step 2: Construct the Second Basis Function**

Compute correlation between signal 2 and basic function 1

$$s_{21} = \int_{-\infty}^{\infty} s_2(t) \phi_1(t) \, dt$$

Subtract off the correlation portion

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

Compute the energy in the remaining portion

$$E_{g_2} = \int_{-\infty}^{\infty} [g_2(t)]^2 \, dt$$

Normalize the remaining portion

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

$$s_{22} = \int_{-\infty}^{\infty} s_2(t) \phi_2(t) \, dt = \sqrt{E_{g_2}}$$

**Step 3: Construct Successive Basis Functions**

For signal $s_k(t)$, compute
Energy of $k$-th basis function:

\[ s_{ki} = \int_{-\infty}^{\infty} s_k(t) \phi_i(t) \, dt \]

Energy of $k$-th basis function:

\[ E_{g_k} = \int_{-\infty}^{\infty} [g_k(t)]^2 \, dt \]

\[ \phi_k(t) = \frac{1}{\sqrt{E_{g_k}}} g_k(t) \]

In general

\[ s_{kk} = \int_{-\infty}^{\infty} s_k(t) \phi_k(t) \, dt = \sqrt{E_{g_k}} \]

**SUMMARY OF GSO PROCEDURE**

1st basis function is normalized version of the first signal Successive basis functions are found by removing portions of signals that are correlated to previous basis functions and normalizing the result. This procedure is repeated until all basis functions are found. If no new basis functions is added. The order in which signals are considered is arbitrary.

A signal set may have many different sets of basis functions. A change of basis functions is essentially a rotation of the signal points around the origin. The order in which signals are used in the GSO procedure affects the resulting basis functions. The choice of basis functions does not affect the performance of the modulation scheme.

**GEOMETRIC REPRESENTATION OF SIGNALS**

Let be

\[ \{\phi_1(t), \ldots, \phi_n(t)\}_{n \text{ signals}} \]

Consider a signal $x(t)$ and suppose that

\[ x(t) = \sum_{i=1}^{n} x_i \phi_i(t) \]

If every signal can be written as above ⇒

\[ \{\phi_1(t), \ldots, \phi_n(t)\} \]

**basis functions:**
• Signal set \(\{\phi_k(t)\}^n\) is an **orthogonal** set if

\[
\int_{-\infty}^{\infty} \phi_j(t) \phi_k(t) \, dt = \begin{cases} 0 & j \neq k \\ c_j & j = k \end{cases}
\]

• If \(c_j \equiv 1 \forall j \rightarrow \{\phi_k(t)\}\) is an **orthonormal** set.

• In this case,

\[
x_k = \int_{-\infty}^{\infty} x(t) \phi_k(t) \, dt
\]

\[
x(t) = \sum_{i=1}^{n} x_i \phi_i(t)
\]

\[
x = (x_1, x_2, \ldots, x_n)
\]

• Given the set of the orthonormal basis

\[
\{\phi_1(t), \ldots, \phi_n(t)\}
\]

• Let \(x(t)\) and \(y(t)\) be represented as

\[
x(t) = \sum_{i=1}^{n} x_i \phi_i(t) , \quad y(t) = \sum_{i=1}^{n} y_i \phi_i(t)
\]

with \(x = (x_1, x_2, \ldots, x_n)\), \(y = (y_1, y_2, \ldots, y_n)\)

• Then the inner product of \(x\) and \(y\) is

\[
x \cdot y = \int_{-\infty}^{\infty} x(t) y(t) \, dt
\]
• Consider a set of M signals (M-ary symbol) \( \{s_i(t), i = 1, 2, ..., M\} \) with finite energy. That is
  \[ \int_{-\infty}^{\infty} s_i^2(t) dt < \infty \]

• Then, we can express each of these waveforms as a weighted linear combination of orthonormal signals
  \[ s_i(t) = \sum_{j=1}^{N} s_{ij} \phi_j(t) \quad \text{for } i = 1, \ldots, M \]

where \( N \leq M \) is the dimension of the signal space and \( \{\phi_j(t)\}_1^N \) are called the orthonormal basis functions.

**Proof**

\[
\int_{-\infty}^{\infty} x(t)y(t) dt = \int_{-\infty}^{\infty} \left[ \sum_{i=1}^{n} x_i \phi_i(t) \right] \left[ \sum_{j=1}^{n} y_j \phi_j(t) \right] dt \\
= \sum_{k=1}^{n} x_k y_k \triangleq x \cdot y
\]

Since

\[
\int_{-\infty}^{\infty} \phi_i(t)\phi_j(t) dt = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases}
\]

\[ E_x = \text{Energy of } x(t) = \int_{-\infty}^{\infty} x^2(t) dt \]

\[ E_x = x \cdot x = \|x\|^2 \]

where \( N \leq M \) is the dimension of the signal space and are called the orthonormal basis functions.
The essence of geometric representation of signals is to represent any set of M energy signals \( s_i(t) \) as linear combinations of \( N \) orthonormal basis functions, where \( N \leq M \).

That is to say, given a set of real-valued energy signals \( s_1(t), s_2(t), \ldots, s_M(t) \), each of duration \( T \) seconds, we write

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

Where the coefficients of the expansion are defined by:

\[
a_{ij} = \int_{0}^{T} s_i(t) \phi_j(t) \, dt, \quad \begin{cases} i = 1, 2, \ldots, M \\ j = 1, 2, \ldots, N \end{cases}
\]

The real-valued basis function are orthonormal

\[
\int_{0}^{T} \phi_i(t) \phi_j(t) \, dt = \delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}
\]

Signal Vector: We may state that each signal is completely determined by the vector of its coefficients

\[
s_i = \begin{bmatrix} a_{i1} \\ a_{i2} \\ \vdots \\ a_{iM} \end{bmatrix}, \quad i = 1, 2, \ldots, M
\]

Signal Space: The N-Dimensional Euclidean space is called the signal space

Example
\[\begin{align*}
N &= 2 \\
M &= 3
\end{align*}\]
Perhaps the most important, and certainly the most analyzed, digital communication channel is the AWGN channel shown in Figure. This channel passes the sum of the modulated signal $x(t)$ and an uncorrelated Gaussian noise $n(t)$ to the output. The Gaussian noise is assumed to be uncorrelated with itself (or “white”) for any non-zero time offset $\tau$, that is

$$E[n(t)n(t-\tau)] = \frac{N_0}{2} \delta(\tau)$$

and zero mean, $E[n(t)] = 0$. With these definitions, the Gaussian noise is also strict sense stationary. The analysis of the AWGN channel is a foundation for the analysis of more complicated channel models in later chapters.

The assumption of white Gaussian noise is valid in the very common situation where the noise is predominantly determined by front-end analog receiver thermal noise.
In the absence of additive noise in Figure $y(t) = x(t)$, and the demodulation process would exactly recover the transmitted signal. This section shows that for the AWGN channel, this demodulation process provides sufficient information to determine optimally the transmitted signal. The resulting components $y_i = \langle y(t), \phi_i(t) \rangle$, $l = 1, ..., N$ comprise a vector channel output, $y = [y_1, ..., y_N]^T$ that is equivalent for detection purposes to $y(t)$. The analysis can thus convert the continuous channel $y(t) = x(t) + n(t)$ to a discrete vector channel model,

$$y = x + n,$$

where $n = [n_1, n_2, ..., n_N]$ and $n_l = h n(t), \phi_l(t)$. The vector channel output is the sum of the vector equivalent of the modulated signal and the vector equivalent of the demodulated noise. Nevertheless, the exact noise sample function may not be reconstructed from $n$.

There may exist a component of $n(t)$ that is orthogonal to the space spanned by the basis functions $\{\phi_1(t), ..., \phi_N(t)\}$. This unrepresented noise component is

$$n^\sim(t) = n(t) - n^\hat{}(t) = y(t) - y^\hat{}(t).$$

The development of the MAP detector could have replaced $y$ by $y(t)$ everywhere and the development would have proceeded identically with the tacit inclusion of the time variable $t$ in the probability densities (and also assuming stationarity of $y(t)$ as a random process). The Theorem of Irrelevance would hold with $[y_1 \ y_2]$ replaced by $[\sim y(t) \sim n(s)]$, as long as the relation holds for any pair of time instants $t$ and $s$. In a non-mathematical sense, the unrepresented noise is useless to the receiver, so there is nothing of value lost in the vector demodulator, even though some of the channel output noise is not represented.

The following algebra demonstrates that $\sim n(s)$ is irrelevant:

First,
Coherent Detection (of A Known Deterministic Signal) in Independent, Identically Distributed (I.I.D.) of signals in noise

\[
E[\hat{n}(s) \cdot \hat{n}(t)] = E \left[ \hat{n}(s) \cdot \sum_{l=1}^{N} n_l \varphi_l(t) \right] = \sum_{l=1}^{N} \varphi_l(t) E[\hat{n}(s) \cdot n_l]
\]

\[
E[\hat{n}(s) \cdot n_l] = E[(n(s) - \hat{n}(s)) \cdot n_l]
\]

\[
= E \left[ \int_{-\infty}^{\infty} n(s) \varphi_l(\tau) n(\tau) d\tau \right] - E \left[ \sum_{k=1}^{N} n_k n_l \varphi_k(s) \right]
\]

\[
= \frac{N_0}{2} \int_{-\infty}^{\infty} \delta(s - \tau) \varphi_l(\tau) d\tau = \frac{N_0}{2} \varphi_l(s)
\]

\[
= \frac{N_0}{2} [\varphi_l(s) - \varphi_l(s)] = 0
\]

Second,

\[
P_{x|\hat{y}(t),\hat{n}(s)} = \frac{P_{x,\hat{y}(t),\hat{n}(s)}}{P_{\hat{y}(t),\hat{n}(s)}} = \frac{P_{x,\hat{y}(t)} \cdot P_{\hat{n}(s)}}{P_{\hat{y}(t)} \cdot P_{\hat{n}(s)}} = \frac{P_{x,\hat{y}(t)}}{P_{\hat{y}(t)}} = P_{x|\hat{y}(t)}
\]

Coherent Detection (of A Known Deterministic Signal) in Independent, Identically Distributed (I.I.D.) of signals in noise

\[ \mathcal{H}_0 : \quad x[n] = w[n], \quad n = 1, 2, \ldots, N \quad \text{versus} \]

\[ \mathcal{H}_1 : \quad x[n] = s[n] + w[n], \quad n = 1, 2, \ldots, N \]

where

\[ s[n] \] is a known deterministic signal and \n\[ w[n] \] is i.i.d. noise with exactly known probability density or mass function (pdf/pmf).
The scenario where the signal $s[n]$, $n = 1, 2, \ldots, N$ is exactly known to the designer is sometimes referred to as the coherent-detection scenario.

The likelihood ratio for this problem is

$$\Lambda(x) = \frac{\prod_{n=1}^{N} p_w(x[n] - s[n])}{\prod_{n=1}^{N} p_w(x[n])}$$

and its logarithm is

$$\log \Lambda(x) = \sum_{n=0}^{N-1} \log \left[ \frac{p_w(x[n] - s[n])}{p_w(x[n])} \right]$$

which needs to be compared with a threshold $\gamma$ (say). Here is a schematic of our coherent detector:

If the noise $w[n]$ i.i.d. $\mathcal{N}(0, \sigma^2)$ (i.e. additive white Gaussian noise, AWGN) and noise variance $\sigma^2$ is known, the likelihood-ratio test reduces to (upon taking log and scaling by $\sigma^2$):

$$\sigma^2 \log \Lambda(x) = \sum_{n=1}^{N} (x[n]s[n] - s^2[n]/2) \overset{H_1}{\geq} \gamma \text{ (a threshold)}$$ (1)
CORRELATION RECEIVER

The requirement for a large number of velocity channels has favored the use of cross-correlation receivers. The principle on which the cross-correlation receiver operates is that, for two random time-varying signals, $V_1(t)$ and $V_2(t)$, the cross-correlation function

$$\sigma(\tau) = \int V_1(t)V_2(t-\tau)dt$$

is the Fourier transform of the visibility spectrum $V_1(\omega) V_2(\omega)$ of the two signals. Here the signals $V_1(t)$ and $V_2(t)$ are the voltages from the two telescopes forming the interferometer, and $V_1(\omega) V_2(\omega)$ is the cross-correlated spectrum at an angular frequency $\omega$. The cross-correlation function is sampled over a range of delays $\pm \Delta T$, $\pm 2\Delta T$, $\ldots$ to a maximum delay $\pm T$ sec, and the visibility spectrum is obtained as the Fourier transform of the sampled cross-correlation function.

The maximum delay $T$ determines the resolution of the synthesized frequency channel, $1/2T$ Hz, and the sampling interval $\Delta T$ produces a grating response in frequency at an interval $1/2\Delta T$. The exact shape of the equivalent frequency filter is the Fourier transform of the weighting applied to the cross-correlation function. If the latter is transformed with equal weight applied to each delay, then the equivalent frequency filters have a $\sin \theta / \theta$ response, with a half-width of $1.2 / 2T$ Hz and $22\%$ side lobes. Both positive and negative delays must be sampled to determine the amplitude and phase of the interferometer, and the correlation receiver is equivalent to a bank of $T / \Delta T$ adjacent frequency filters at intervals of $1 / T$ Hz.

The cross-correlation may be achieved in practice either in an analogue device using physical delay steps or in a digital correlator. In the latter, simplified logic results if one-bit sampling of the correlation function is employed (so that only the sign of the sampled correlation function is recorded). This results in some loss in signal-to-noise ratio, but the visibility spectrum may be fully restored (Weinreb, 1963) through the Van Flyck correction. While increasing the complexity of the data processing, as an extra Fourier transform must be computed, the correlation receiver has a number of advantages over a conventional filter bank receiver in that the relative sensitivity of the frequency channels is easily calibrated. A digital correlator has good stability essential for a good synthesis, and the additional

and is known as the _correlator detector_ or simply _correlator_.
advantage that the bandwidth can be changed by simply changing the clock rate which
determines the sampling interval.

Correlation Demodulator

- Consider each demodulator output

\[
\begin{align*}
\mathbf{f}_k(t) & = \int_0^T r(t) f_k(t) dt \\
& = \int_0^T s_m(t) f_k(t) dt \\
& \quad + \int_0^T n(t) f_k(t) dt \\
& = s_{mk} + n_k
\end{align*}
\]
Correlator outputs

\[ E(n_k n_m) = \int_{0}^{T} \int_{0}^{T} E[n(t)n(\tau)] f_k(t) f_m(\tau) dt d\tau \]
\[ = \frac{1}{2} N_0 \int_{0}^{T} f_k(t) f_m(\tau) dt \]
\[ = \begin{cases} 
\frac{1}{2} N_0 & m = k \\
0 & m \neq k 
\end{cases} \]

**EQUIVALENCE OF CORRELATION N MATCHED FILTER RECEIVER**

- Use filters whose impulse response is the orthonormal basis of signal
- Can show this is *exactly equivalent* to the correlation demodulator
- We find that this Demodulator Maximizes the SNR
- Essentially show that any other function than \( f_d(t) \) decreases SNR as is not as well correlated to components of \( r(t) \)
PROBABILITY OF ERROR IN TRANSMISSION

In a binary PCM system, binary digits may be represented by two pulse levels. If these levels are chosen to be 0 and A, the signal is termed an on-off binary signal. If the level switches between A=2 and A=2 it is called a polar binary signal.

Suppose we are transmitting digital information, and decide to do this using two-level pulses each with period $T$.
The binary digit 0 is represented by a signal of level 0 for the duration $T$ of the transmission, and the digit 1 is represented by the signal level $A_T$.

In what follows we do not consider modulating the signal — it is transmitted at baseband. In the event of a noisy Gaussian channel (with high bandwidth) the signal at the receiver may look as follows:

Here the binary levels at the receiver are nominally 0 (signal absent) and $A$ (signal present) upon receipt of a 0 or 1 digit respectively.

The function of a receiver is to distinguish the digit 0 from the digit 1. The most important performance characteristic of the receiver is the probability that an error will be made in such a determination.

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

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

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

$$f(t) = \begin{cases} 
0 & \text{symbol 0 transmitted (signal absent)} \\
1 & \text{symbol 1 transmitted (signal present).}
\end{cases}$$

In what follows, it is assumed that the transmitter and the receiver are synchronised, so the receiver has perfect knowledge of the arrival times of sequences of pulses. The means of achieving this synchronisation is not considered here. This means that without loss of generality we can always assume that the bit to be received lies in the interval 0 to $T$. 
SIGNAL CONSTELLATION DIAGRAMS

A graphical representation of the complex envelope of each possible signal. The x-axis represents the in-phase component and the y-axis represents the quadrature component of the complex envelope. The distance between signals on a constellation diagram relates to how different the modulation waveforms are and how well a receiver can differentiate between them when random noise is present.

Two key performance measures of a modulation scheme are power efficiency and bandwidth efficiency.

Power efficiency is a measure of how favorably the tradeoff between fidelity and signal power is made, and is expressed as the ratio of the signal energy per bit \( E_b \) to the noise PSD \( N_0 \) required to achieve a given probability of error (say \( 10^{-5} \)):

Bandwidth efficiency describes the ability of a modulation scheme to accommodate data within a limited bandwidth, in general, it is defined as the ratio of the data bit rate \( R \) to the required RF bandwidth \( B \):

Channel capacity gives an upper bound of achievable bandwidth efficiency:

\[
\eta_{B_{max}} = \frac{C}{B} = \log_2 \left(1 + N^5\right)
\]

A constellation diagram is a representation of a signal modulated by a digital modulation scheme such as quadrature amplitude modulation or phase-shift keying. It displays the signal as a two-dimensional X-Y plane scatter diagram in the complex plane at symbol sampling instants. In a more abstract sense, it represents the possible symbols that may be selected by a given modulation scheme as points in the complex plane. Measured constellation diagrams can be used to recognize the type of interference and distortion in a signal.
A constellation diagram for Gray encoded 8-PSK.

By representing a transmitted symbol as a complex number and modulating a cosine and sine carrier signal with the real and imaginary parts (respectively), the symbol can be sent with two carriers on the same frequency. They are often referred to as *quadrature carriers*. A coherent detector is able to independently demodulate these carriers. This principle of using two independently modulated carriers is the foundation of quadrature modulation. In pure phase modulation, the phase of the modulating symbol is the phase of the carrier itself and this is the best representation of the modulated signal.

As the symbols are represented as complex numbers, they can be visualized as points on the complex plane. The real and imaginary axes are often called the *in phase*, or I-axis, and the *quadrature*, or Q-axis, respectively. Plotting several symbols in a scatter diagram produces the constellation diagram. The points on a constellation diagram are called *constellation points*. They are a set of *modulation symbols* which compose the *modulation alphabet*.

Also a diagram of the ideal positions, signal space diagram, in a modulation scheme can be called a constellation diagram. In this sense the constellation is not a scatter diagram but a representation of the scheme itself. The example shown here is for 8-PSK, which has also been given a Gray coded bit assignment.
2.1 INTRODUCTION

Referring to Equation (2.1), if the information signal is digital and the amplitude (V) of the carrier is varied proportional to the information signal, a digitally modulated signal called amplitude shift keying (ASK) is produced.

If the frequency (f) is varied proportional to the information signal, frequency shift keying (FSK) is produced, and if the phase of the carrier (θ) is varied proportional to the information signal, phase shift keying (PSK) is produced.

If both the amplitude and the phase are varied proportional to the information signal, quadrature amplitude modulation (QAM) results. ASK, FSK, PSK, and QAM are all forms of digital modulation:

![Figure 2-1 simplified block diagram for a digital modulation system](image)

\[ v(t) = V \sin (2\pi \cdot ft + \theta) \]  

(2.1)

Figure 2-1 shows a simplified block diagram for a digital modulation system.
In the transmitter, the precoder performs level conversion and then encodes the incoming data into groups of bits that modulate an analog carrier.

The modulated carrier is shaped (filtered), amplified, and then transmitted through the transmission medium to the receiver.

The transmission medium can be a metallic cable, optical fiber cable, Earth's atmosphere, or a combination of two or more types of transmission systems.

In the receiver, the incoming signals are filtered, amplified, and then applied to the demodulator and decoder circuits, which extracts the original source information from the modulated carrier.

The clock and carrier recovery circuits recover the analog carrier and digital timing (clock) signals from the incoming modulated wave since they are necessary to perform the de-modulation process.

FIGURE 2-1 Simplified block diagram of a digital radio system.
2-2 INFORMATION CAPACITY, BITS, BIT RATE, BAUD, AND MARY ENCODING

2-2-1 Information Capacity, Bits, and Bit Rate

\[ I \propto B \times t \quad (2.2) \]

where \( I \) = information capacity (bits per second) \( B \) = bandwidth (hertz) \( t \) = transmission time (seconds)

From Equation 2-2, it can be seen that information capacity is a linear function of bandwidth and transmission time and is directly proportional to both.

If either the bandwidth or the transmission time changes, a directly proportional change occurs in the information capacity.

The higher the signal-to-noise ratio, the better the performance and the higher the information capacity.

Mathematically stated, the Shannon limit for information capacity is

\[ I = B \log_2 \left( 1 + \frac{S}{N} \right) \quad (2.3) \]

or

\[ I = 3.32B \log_{10} \left( 1 + \frac{S}{N} \right) \quad (2.4) \]

where \( I \) = information capacity (bps) \( B \) = bandwidth (hertz) \( \frac{S}{N} \) = signal-to-noise power ratio (unitless)
For a standard telephone circuit with a signal-to-noise power ratio of 1000 (30 dB) and a bandwidth of 2.7 kHz, the Shannon limit for information capacity is

\[ I = (3.32)(2700) \log_{10} (1 + 1000) = 26.9 \text{ kbps} \]

Shannon’s formula is often misunderstood. The results of the preceding example indicate that 26.9 kbps can be propagated through a 2.7-kHz communications channel. This may be true, but it cannot be done with a binary system. To achieve an information transmission rate of 26.9 kbps through a 2.7-kHz channel, each symbol transmitted must contain more than one bit.

**2-2-2 M-ary Encoding**

*M-ary* is a term derived from the word *binary*.

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

For example, a digital signal with four possible conditions (voltage levels, frequencies, phases, and so on) is an M-ary system where \( M = 4 \). If there are eight possible conditions, \( M = 8 \) and so forth.

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

\[ N \otimes \log_2 M \] (2.5)

where

- \( N \) = number of bits necessary
- \( M \) = number of conditions, levels, or combinations
possible with $N$ bits

Equation 2-5 can be simplified and rearranged to express the number of conditions possible with $N$ bits as

$$2^N = M$$ (2.6)

For example, with one bit, only $2^1 = 2$ conditions are possible. With two bits, $2^2 = 4$ conditions are possible, with three bits, $2^3 = 8$ conditions are possible, and so on.

### 2-2-3 Baud and Minimum Bandwidth

Baud refers to the rate of change of a signal on the transmission medium after encoding and modulation have occurred.

Hence, baud is a unit of transmission rate, modulation rate, or symbol rate and, therefore, the terms symbols per second and baud are often used interchangeably.

Mathematically, baud is the reciprocal of the time of one output signaling element, and a signaling element may represent several information bits. Baud is expressed as

$$\text{baud} = \frac{1}{t_s},$$ (2.7)

where
- $\text{baud} =$ symbol rate (baud per second)
- $t_s =$ time of one signaling element (seconds)
The minimum theoretical bandwidth necessary to propagate a signal is called the minimum Nyquist bandwidth or sometimes the minimum Nyquist frequency.

Thus, \( f_b = 2B \), where \( f_b \) is the bit rate in bps and \( B \) is the ideal Nyquist bandwidth.

The relationship between bandwidth and bit rate also applies to the opposite situation. For a given bandwidth \( (B) \), the highest theoretical bit rate is \( 2B \).

For example, a standard telephone circuit has a bandwidth of approximately 2700 Hz, which has the capacity to propagate 5400 bps through it. However, if more than two levels are used for signaling (higher-than-binary encoding), more than one bit may be transmitted at a time, and it is possible to propagate a bit rate that exceeds \( 2B \).

Using multilevel signaling, the Nyquist formulation for channel capacity is

\[
f_b = B \log_2 M
\]  

(2.8)

where

\( f_b \) = channel capacity (bps)

\( B \) = minimum Nyquist bandwidth (hertz)

\( M \) = number of discrete signal or voltage levels

Equation 2.8 can be rearranged to solve for the minimum bandwidth necessary to pass \( M \)-ary digitally modulated carriers

\[
B = \frac{f_b}{\log_2 M}
\]  

(2.9)
If \( N \) is substituted for \( \log_2 M \), Equation 2.9 reduces to

\[
B = \frac{f}{b} N
\]  
(2.10)

where \( N \) is the number of bits encoded into each signaling element.

In addition, since baud is the encoded rate of change, it also equals the bit rate divided by the number of bits encoded into one signaling element. Thus,

\[
B_{\text{Baud}} = \frac{f}{b} N
\]  
(2.11)

By comparing Equation 2.10 with Equation 2.11 the baud and the ideal minimum Nyquist bandwidth have the same value and are equal to the bit rate divided by the number of bits encoded.

2-3 AMPLITUDE-SHIFT KEYING

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

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

\[ v_{\text{ask}}(t) = \left[ 1 + v_m(t) \right] \left[ \frac{A}{2} \cos(\omega_c t) \right] \] (2.12)

where

- \( v_{\text{ask}}(t) \) = amplitude-shift keying wave
- \( v_m(t) \) = digital information (modulating) signal (volts) \( A/2 \) = unmodulated carrier amplitude (volts)
- \( \omega_c \) = analog carrier radian frequency (radians per second, \( 2\pi f_c t \))

In Equation 2.12, the modulating signal \([v_m(t)]\) is a normalized binary waveform, where +1 V = logic 1 and -1 V = logic 0. Therefore, for a logic 1 input, \( v_m(t) = +1 \) V, Equation 2.12 reduces to

\[ v_{\text{ask}}(t) = \left[ 1 + 1 \right] \left[ \frac{A}{2} \cos(\omega_c t) \right] \]

\[ = A \cos(\omega_c t) \]

and for a logic 0 input, \( v_m(t) = -1 \) V, Equation 2.12 reduces to

\[ v_{\text{ask}}(t) = \left[ 1 - 1 \right] \left[ \frac{A}{2} \cos(\omega_c t) \right] \]

Thus, the modulated wave \( v_{\text{ask}}(t) \), is either \( A \cos(\omega_c t) \) or 0. Hence, the carrier is either "on" or "off," which is why amplitude-shift keying is sometimes referred to as on-off keying (OOK).

8
Figure 2-2 shows the input and output waveforms from an ASK modulator.

From the figure, it can be seen that for every change in the input binary data stream, there is one change in the ASK waveform, and the time of one bit \((t_b)\) equals the time of one analog signaling element \((t_s)\).

\[
B = f_b / 1 = f_b \quad \text{baud} = f_b / 1 = f_b
\]

The entire time the binary input is high, the output is a constant-amplitude, constant-frequency signal, and for the entire time the binary input is low, the carrier is off.

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

**Example 2-1**

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

For ASK, $N = 1$, and the baud and minimum bandwidth are determined from Equations 2.11 and 2.10, respectively:

$$B = \frac{10,000}{1} = 10,000$$

baud = $\frac{10,000}{1} = 10,000$

The use of amplitude-modulated analog carriers to transport digital information is a relatively low-quality, low-cost type of digital modulation and, therefore, is seldom used except for very low-speed telemetry circuits.

2-4 FREQUENCY-SHIFT KEYING

FSK is a form of constant-amplitude angle modulation similar to standard frequency modulation (FM) except the modulating signal is a binary signal that varies between two discrete voltage levels rather than a continuously changing analog waveform.

Consequently, FSK is sometimes called binary FSK (BFSK). The general expression for FSK is

$$v_{fsk}(t) = V_c \cos[2\pi f_c + \nu_m(t) \Delta f t]$$  \hspace{1cm} (2.13)

where

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

$V_c = \text{peak analog carrier amplitude (volts)}$ $f_c = \text{analog carrier center frequency (hertz)}$

$\nu_m(t) = \text{peak change (shift) in the analog carrier frequency}$
From Equation 2.13, it can be seen that the peak shift in the carrier frequency \( f \) is proportional to the amplitude of the binary input signal \( v_m(t) \), and the direction of the shift is determined by the polarity.

The modulating signal is a normalized binary waveform where a logic 1 = +1 V and a logic 0 = -1 V. Thus, for a logic 1 input, \( v_m(t) = +1 \), Equation 2.13 can be rewritten as

\[
v_{f_{sk}}(t) = V_c \cos[2\pi(f_c + \Delta f)t]
\]

For a logic 0 input, \( v_m(t) = -1 \), Equation 2.13 becomes

\[
v_{f_{sk}}(t) = V_c \cos[2\pi(f_c - \Delta f)t]
\]

With binary FSK, the carrier center frequency \( f_c \) is shifted (deviated) up and down in the frequency domain by the binary input signal as shown in Figure 2-3.

\[f_s \quad f_c \quad f_m\]

\[\Delta f \quad +\Delta f\]

\[
\text{Logic 0} \quad \text{Binary input signal} \\
\text{Logic 1}
\]

FIGURE 2-3 FSK in the frequency domain
As the binary input signal changes from a logic 0 to a logic 1 and vice versa, the output frequency shifts between two frequencies: a mark, or logic 1 frequency \( f_m \), and a space, or logic 0 frequency \( f_s \). The mark and space frequencies are separated from the carrier frequency by the peak frequency deviation \( f \) and from each other by \( 2f \).

Frequency deviation is illustrated in Figure 2-3 and expressed mathematically as

\[
f = \frac{|f_m - f_s|}{2}
\]  

(2.14)

where 
- \( f \) = frequency deviation (hertz)
- \( |f_m - f_s| \) = absolute difference between the mark and space frequencies (hertz)

Figure 2-4a shows in the time domain the binary input to an FSK modulator and the corresponding FSK output.

When the binary input \( (f_b) \) changes from a logic 1 to a logic 0 and vice versa, the FSK output frequency shifts from a mark \( f_m \) to a space \( f_s \) frequency and vice versa.

In Figure 2-4a, the mark frequency is the higher frequency \( (f_c + f) \) and the space frequency is the lower frequency \( (f_c - f) \), although this relationship could be just the opposite.

Figure 2-4b shows the truth table for a binary FSK modulator. The truth table shows the input and output possibilities for a given digital modulation scheme.
2-4-1 FSK Bit Rate, Baud, and Bandwidth

In Figure 2-4a, it can be seen that the time of one bit \( t_b \) is the same as the time the FSK output is a mark of space frequency \( t_s \). Thus, the bit time equals the time of an FSK signaling element, and the bit rate equals the baud.

The baud for binary FSK can also be determined by substituting \( N = 1 \) in Equation 2.11:

\[
\text{baud} = \frac{f_b}{1} = f_b
\]

The minimum bandwidth for FSK is given as

\[
B = |(f_s - f_b) - (f_m - f_b)|
\]

\[
= |f_s - f_m| + 2f_b
\]

and since \( |f_s - f_m| \) equals 2 \( f \), the minimum bandwidth can be approximated as

\[
B = 2( f + f_b )
\]

(2.15)
where

\[ B = \text{minimum Nyquist bandwidth (hertz)} \]
\[ f = \text{frequency deviation } |f_m - f_s| \text{ (hertz)} \]
\[ f_b = \text{input bit rate (bps)} \]

Example 2-2

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

Solution

a. The peak frequency deviation is determined from Equation 2.14:

\[ f = |49 \text{kHz} - 51 \text{kHz}| / 2 = 1 \text{ kHz} \]

b. The minimum bandwidth is determined from Equation 2.15:

\[ B = 2(1000 + 2000) = 6 \text{ kHz} \]

c. For FSK, \( N = 1 \), and the baud is determined from Equation 2.11 as

\[ \text{baud} = 2000 / 1 = 2000 \]

Bessel functions can also be used to determine the approximate bandwidth for an FSK wave. As shown in Figure 2-5, the fastest rate of change (highest fundamental frequency) in a non-return-to-zero (NRZ) binary signal occurs when alternating 1s and 0s are occurring (i.e., a square wave).
Since it takes a high and a low to produce a cycle, the highest fundamental frequency present in a square wave equals the repetition rate of the square wave, which with a binary signal is equal to half the bit rate. Therefore,

\[ f_a = \frac{f_b}{2} \]  

(2.16)

where

- \( f_a \) = highest fundamental frequency of the binary input signal (hertz)
- \( f_b \) = input bit rate (bps)

The formula used for modulation index in FM is also valid for FSK; thus,

\[ h = \frac{f}{f_a} \quad \text{(unitless)} \]  

(2.17)

where

- \( h \) = FM modulation index called the h-factor in FSK
- \( f_o \) = fundamental frequency of the binary modulating signal (hertz)
\[ f = \text{peak frequency deviation (hertz)} \]

The peak frequency deviation in FSK is constant and always at its maximum value, and the highest fundamental frequency is equal to half the incoming bit rate. Thus,

\[
h = \frac{|f_m - f_s|}{2} \frac{f_b}{2}
\]

or

\[
h = \frac{|f_m - f_s|}{f_b}
\]

(2.18)

where
\( h = \text{h-factor (unitless)} \)
\( f_m = \text{mark frequency (hertz)} \)
\( f_s = \text{space frequency (hertz)} \)
\( f_b = \text{bit rate (bits per second)} \)

**Example 2-3**

Using a Bessel table, determine the minimum bandwidth for the same FSK signal described in Example 2-1 with a mark frequency of 49 kHz, a space frequency of 51 kHz, and an input bit rate of 2 kbps.

Solution The modulation index is found by substituting into Equation 2.17:

\[
h = \frac{|49 \text{ kHz} - 51 \text{ kHz}|}{2 \text{ kbps}} = 2
\]

From a Bessel table, three sets of significant sidebands are produced for a modulation index of one. Therefore, the bandwidth can be determined as follows:

\[ B = 2(3 \times 1000) \]
The bandwidth determined in Example 2-3 using the Bessel table is identical to the bandwidth determined in Example 2-2.

<table>
<thead>
<tr>
<th>Modulation Index</th>
<th>Carrier</th>
<th>Side Frequency Pairs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>m</td>
<td>x_m</td>
</tr>
<tr>
<td>0.00</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>0.25</td>
<td>0.08</td>
<td>0.12</td>
</tr>
<tr>
<td>0.50</td>
<td>0.17</td>
<td>0.22</td>
</tr>
<tr>
<td>1.00</td>
<td>0.34</td>
<td>0.41</td>
</tr>
<tr>
<td>2.00</td>
<td>0.58</td>
<td>0.65</td>
</tr>
<tr>
<td>2.50</td>
<td>0.82</td>
<td>0.89</td>
</tr>
<tr>
<td>3.00</td>
<td>0.91</td>
<td>0.97</td>
</tr>
<tr>
<td>4.00</td>
<td>1.10</td>
<td>1.14</td>
</tr>
<tr>
<td>5.00</td>
<td>1.22</td>
<td>1.26</td>
</tr>
<tr>
<td>6.00</td>
<td>1.30</td>
<td>1.34</td>
</tr>
<tr>
<td>7.00</td>
<td>1.38</td>
<td>1.41</td>
</tr>
<tr>
<td>8.00</td>
<td>1.44</td>
<td>1.47</td>
</tr>
<tr>
<td>9.00</td>
<td>1.50</td>
<td>1.52</td>
</tr>
<tr>
<td>10.00</td>
<td>1.56</td>
<td>1.58</td>
</tr>
</tbody>
</table>

2-4-2 FSK Transmitter

Figure 2-6 shows a simplified binary FSK modulator, which is very similar to a conventional FM modulator and is very often a voltage-controlled oscillator (VCO).

The center frequency (f_c) is chosen such that it falls halfway between the mark and space frequencies.

![FIGURE 2-6 FSK modulator](image-url)
A logic 1 input shifts the VCO output to the mark frequency, and a logic 0 input shifts the VCO output to the space frequency.

Consequently, as the binary input signal changes back and forth between logic 1 and logic 0 conditions, the VCO output shifts or deviates back and forth between the mark and space frequencies.

\[ f = v_{m}(t)k_I \]  

\( v_{m}(t) = \) peak binary modulating-signal voltage (volts)  
\( k_I = \) deviation sensitivity (hertz per volt).

FIGURE 2-6 FSK modulator
2-4-3 FSK Receiver

FSK demodulation is quite simple with a circuit such as the one shown in Figure 2-7.

![Diagram of Noncoherent FSK demodulator](image)

FIGURE 2-7 Noncoherent FSK demodulator

The FSK input signal is simultaneously applied to the inputs of both bandpass filters (BPFs) through a power splitter.

The respective filter passes only the mark or only the space frequency on to its respective envelope detector.

The envelope detectors, in turn, indicate the total power in each passband, and the comparator responds to the largest of the two powers.

This type of FSK detection is referred to as noncoherent detection.
Figure 2-8 shows the block diagram for a coherent FSK receiver.

The incoming FSK signal is multiplied by a recovered carrier signal that has the exact same frequency and phase as the transmitter reference.

However, the two transmitted frequencies (the mark and space frequencies) are not generally continuous; it is not practical to reproduce a local reference that is coherent with both of them. Consequently, coherent FSK detection is seldom used.

![Coherent FSK demodulator](image)

FIGURE 2-8 Coherent FSK demodulator

The most common circuit used for demodulating binary FSK signals is the *phaselocked loop* (PLL), which is shown in block diagram form in Figure 2-9.

![PLL-FSK demodulator](image)

FIGURE 2-9 PLL-FSK demodulator

As the input to the PLL shifts between the mark and space frequencies, the *dc error voltage* at the output of the phase...
comparator follows the frequency shift.

Because there are only two input frequencies (mark and space), there are also only two output error voltages. One represents a logic 1 and the other a logic 0.

Binary FSK has a poorer error performance than PSK or QAM and, consequently, is seldom used for high-performance digital radio systems.

Its use is restricted to low-performance, low-cost, asynchronous data modems that are used for data communications over analog, voice-band telephone lines.

2-4-4 Continuous-Phase Frequency-Shift Keying

Continuous-phase frequency-shift keying (CP-FSK) is binary FSK except the mark and space frequencies are synchronized with the input binary bit rate.

With CP-FSK, the mark and space frequencies are selected such that they are separated from the center frequency by an exact multiple of one-half the bit rate ($f_m$ and $f_s = n[f_b / 2]$, where $n = \text{any integer}$).

This ensures a smooth phase transition in the analog output signal when it changes from a mark to a space frequency or vice versa.

Figure 2-10 shows a noncontinuous FSK waveform. It can be seen that when the input changes from a logic 1 to a logic 0 and vice versa, there is an abrupt phase discontinuity in the analog signal. When this occurs, the demodulator has trouble following the frequency shift; consequently, an error may occur.
FIGURE 2-10 Noncontinuous FSK waveform

Figure 2-11 shows a continuous phase FSK waveform.

FIGURE 2-11 Continuous-phase MSK waveform

Notice that when the output frequency changes, it is a smooth, continuous transition. Consequently, there are no phase discontinuities.

CP-FSK has a better bit-error performance than conventional binary FSK for a given signal-to-noise ratio.

The disadvantage of CP-FSK is that it requires synchro-nization circuits and is, therefore, more expensive to implement.
2-5 PHASE-SHIFT KEYING

*Phase-shift keying* (PSK) is another form of *angle-modulated, constant-amplitude* digital modulation.

2-5-1 Binary Phase-Shift Keying

The simplest form of PSK is *binary phase-shift keying* (BPSK), where $N = 1$ and $M = 2$.

Therefore, with BPSK, two phases ($2^1 = 2$) are possible for the carrier. One phase represents a logic 1, and the other phase represents a logic 0. As the input digital signal changes state (i.e., from a 1 to a 0 or from a 0 to a 1), the phase of the output carrier shifts between two angles that are separated by 180°.

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

![FIGURE 2-12 BPSK transmitter](image-url)
2-5-1-1 BPSK transmitter.

Figure 2-12 shows a simplified block diagram of a BPSK transmitter.

The balanced modulator acts as a phase reversing switch. Depending on the logic condition of the digital input, the carrier is transferred to the output either in phase or 180° out of phase with the reference carrier oscillator.

Figure 2-13 shows the schematic diagram of a balanced ring modulator.

The balanced modulator has two inputs: a carrier that is in phase with the reference oscillator and the binary digital data.

For the balanced modulator to operate properly, the digital input voltage must be much greater than the peak carrier voltage.

This ensures that the digital input controls the on/off state of diodes D1 to D4. If the binary input is a logic 1 (positive voltage), diodes D1 and D2 are forward biased and on, while diodes D3 and D4 are reverse biased and off (Figure 2-13b). With the polarities shown, the carrier voltage is developed across transformer T2 in phase with the carrier voltage across T1. Consequently, the output signal is in phase with the reference oscillator.

If the binary input is a logic 0 (negative voltage), diodes D1 and D2 are reverse biased and off, while diodes D3 and D4 are forward biased and on (Figure 9-13c). As a result, the carrier voltage is developed across transformer T2 180° out of phase with the carrier voltage across T1.
FIGURE 9-13 (a) Balanced ring modulator; (b) logic 1 input; (c) logic 0 input
FIGURE 2-14 BPSK modulator: (a) truth table; (b) phasor diagram; (c) constellation diagram

2-5-1-2 Bandwidth considerations of BPSK.

In a BPSK modulator, the carrier input signal is multiplied by the binary data.

If +1 V is assigned to a logic 1 and -1 V is assigned to a logic 0, the input carrier (\(\sin \omega c t\)) is multiplied by either a + or -1.

The output signal is either +1 \(\sin \omega c t\) or -1 \(\sin \omega c t\) the first represents a signal that is in phase with the reference oscillator, the latter a signal that is 180° out of phase with the reference oscillator.
Each time the input logic condition changes, the output phase changes.

Mathematically, the output of a BPSK modulator is proportional to

\[
\text{BPSK output} = [\sin (2\pi f_a t)] \times [\sin (2\pi f_c t)]
\]  (2.20)

where
\[ f_a = \text{maximum fundamental frequency of binary input (hertz)} \]
\[ f_c = \text{reference carrier frequency (hertz)} \]

Solving for the trig identity for the product of two sine functions,

\[
0.5\cos[2\pi(f_c - f_a)t] - 0.5\cos[2\pi(f_c + f_a)t]
\]

Thus, the minimum double-sided Nyquist bandwidth \( B \) is

\[
f_c + f_a \quad \text{or} \quad \frac{f_c + f_a}{2f_a}
\]

and because \( f_a = f_b / 2 \), where \( f_b = \text{input bit rate} \), where \( B \) is the minimum double-sided Nyquist bandwidth.

Figure 2-15 shows the output phase-versus-time relationship for a BPSK waveform.

Logic 1 input produces an analog output signal with a 0° phase angle, and a logic 0 input produces an analog output signal with a 180° phase angle.

27
As the binary input shifts between a logic 1 and a logic 0 condition and vice versa, the phase of the BPSK waveform shifts between 0° and 180°, respectively.

BPSK signaling element \( (t_s) \) is equal to the time of one information bit \( (t_b) \), which indicates that the bit rate equals the baud.

![Diagram of BPSK signaling and phase-shift](image)

FIGURE 2-15 Output phase-versus-time relationship for a BPSK modulator

**Example 2-4**

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

Substituting into Equation 2-20 yields

\[
\text{output} = [\sin (2\pi f_a t)] \times [\sin (2\pi f_b t)]; \quad f_a = f_b / 2 = 5 \text{ MHz}
\]

\[
= [\sin 2\pi (5\text{MHz})t] \times [\sin 2\pi (70\text{MHz})t] \\
= 0.5\cos[2\pi (70\text{MHz} - 5\text{MHz})t] - 0.5\cos[2\pi (70\text{MHz} + 5\text{MHz})t]
\]

lower side frequency \quad \text{upper side frequency}\]

Minimum lower side frequency (LSF):

LSF = 70 MHz - 5 MHz = 65 MHz

Maximum upper side frequency (USF):

USF = 70 MHz + 5 MHz = 75 MHz

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

\[
B = 75 \text{ MHz} - 65 \text{ MHz} = 10 \text{ MHz}
\]

and the baud = \(f_b\) or 10 megabaud.
2-5-1-3 BPSK receiver.

Figure 2-16 shows the block diagram of a BPSK receiver.
The input signal maybe $+ \sin \omega_c t$ or $- \sin \omega_c t$.

The coherent carrier recovery circuit detects and regenerates a carrier signal that is both frequency and phase coherent with the original transmit carrier.

The balanced modulator is a product detector; the output is the product of the two inputs (the BPSK signal and the recovered carrier).

The low-pass filter (LPF) operates the recovered binary data from the complex demodulated signal.

FIGURE 2-16 Block diagram of a BPSK receiver
Mathematically, the demodulation process is as follows.

For a BPSK input signal of \( + \sin \omega_c t \) (logic 1), the output of the balanced modulator is

\[
\text{output} = (\sin \omega_c t)(\sin \omega_c t) = \sin^2 \omega_c t
\]

or

\[
\sin^2 \omega_c t = 0.5(1 - \cos 2\omega_c t) = 0.5 - 0.5 \cos 2\omega_c t
\]

filtered out

leaving

\[
\text{output} = + 0.5 \text{ V} = \text{logic 1}
\]

It can be seen that the output of the balanced modulator contains a positive voltage (+[1/2]V) and a cosine wave at twice the carrier frequency (2\( \omega_c t \)).

The LPF has a cutoff frequency much lower than 2\( \omega_c t \), and, thus, blocks the second harmonic of the carrier and passes only the positive constant component. A positive voltage represents a demodulated logic 1.

For a BPSK input signal of \( -\sin \omega_c t \) (logic 0), the output of the balanced modulator is

\[
\text{output} = (-\sin \omega_c t)(\sin \omega_c t) = \sin^2 \omega_c t
\]

or

\[
\sin^2 \omega_c t = -0.5(1 - \cos 2\omega_c t) = 0.5 + 0.5 \cos 2\omega_c t
\]

filtered out

\[31\]
leaving

\[ \text{output} = -0.5 \text{ V} = \text{logic 0} \]

The output of the balanced modulator contains a negative voltage (-[I/2]V) and a cosine wave at twice the carrier frequency (2\(\omega_c t\)).

Again, the LPF blocks the second harmonic of the carrier and passes only the negative constant component. A negative voltage represents a demodulated logic 0.

2-5-2 Quaternary Phase-Shift Keying

QPSK is an M-ary encoding scheme where \(N = 2\) and \(M = 4\) (hence, the name "quaternary" meaning "4"). A QPSK modulator is a binary (base 2) signal, to produce four different input combinations: 00, 01, 10, and 11.

Therefore, with QPSK, the binary input data are combined into groups of two bits, called *dibits*. In the modulator, each dibit code generates one of the four possible output phases (+45°, +135°, -45°, and -135°).

2-5-2-1 QPSK transmitter.

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

The I bit modulates a carrier that is in phase with the reference oscillator (hence the name "I" for "in phase" channel), and the
Q bit modulate, a carrier that is 90° out of phase.

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

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

FIGURE 2-17 QPSK modulator
Example 2-5
For the QPSK modulator shown in Figure 2-17, construct the truth table, phasor diagram, and constellation diagram.

Solution

For a binary data input of Q = 0 and I = 0, the two inputs to the I balanced modulator are -1 and sin $\omega_c t$, and the two inputs to the Q balanced modulator are -1 and cos $\omega_c t$.

Consequently, the outputs are

I balanced modulator = (-1)(sin $\omega_c t$) = -1 sin $\omega_c t$
Q balanced modulator = (-1)(cos $\omega_c t$) = -1 cos $\omega_c t$ and the output of the linear summer is

-1 cos $\omega_c t$ - 1 sin $\omega_c t = 1.414 \sin(\omega_c t - 135^\circ)$

For the remaining dibit codes (01, 10, and 11), the procedure is the same. The results are shown in Figure 2-18a.
In Figures 2-18b and c, it can be seen that with QPSK each of the four possible output phasors has exactly the same amplitude. Therefore, the binary information must be encoded entirely in the phase of the output signal.

Figure 2-18b, it can be seen that the angular separation between any two adjacent phasors in QPSK is 90°.

Therefore, a QPSK signal can undergo almost a+45° or -45° shift in phase during transmission and still retain the correct encoded information when demodulated at the receiver.

Figure 2-19 shows the output phase-versus-time relationship for a QPSK modulator.
FIGURE 2-19 Output phase-versus-time relationship for a PSK modulator.

2-5-2-2 Bandwidth considerations of QPSK

With QPSK, because the input data are divided into two channels, the bit rate in either the I or the Q channel is equal to one-half of the input data rate ($f_b/2$) (one-half of $f_b/2 = f_b/4$).

This relationship is shown in Figure 2-20.

FIGURE 2-20 Bandwidth considerations of a QPSK modulator

In Figure 2-20, it can be seen that the worse-case input condition to the I or Q balanced modulator is an alternative 1/0 pattern, which occurs when the binary input data have a 1100 repetitive pattern. One cycle of the fastest binary transition (a 1/0 sequence in the I or Q channel takes the same time as four input data bits.)
Consequently, the highest fundamental frequency at the input and fastest rate of change at the output of the balance modulators is equal to one-fourth of the binary input bit rate.

The output of the balanced modulators can be expressed mathematically as

\[
\text{output} = (\sin \omega_d t)(\sin \omega_c t) \tag{2.22}
\]

where

\[
\omega_d t = 2\pi \frac{f_b}{4} \quad \text{and} \quad \omega_c t = 2\pi f_c
\]

The output frequency spectrum extends from \(f_c + f_b / 4\) to \(f_c - f_b / 4\) and the minimum bandwidth \((f_N)\) is

\[
\left(f_c + \frac{f_b}{4}\right) - \left(f_c - \frac{f_b}{4}\right) = \frac{2f_b}{4} = \frac{f_b}{2}
\]

Example 2-6

For a QPSK modulator with an input data rate \(f_b\) equal to 10 Mbps and a carrier frequency 70 MHz, determine the minimum double-sided Nyquist bandwidth \((f_N)\) and the baud. Also, compare the results with those achieved with the BPSK modulator in
Example 2-4. Use the QPSK block diagram shown in Figure 2-17 as the modulator model.

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

\[ f_{bQ} = f_{b1} = \frac{f_b}{2} = \frac{10 \text{ Mbps}}{2} = 5 \text{ Mbps} \]

The highest fundamental frequency presented to either balanced modulator is

\[ f_a = \frac{f_{bQ}}{2} = \frac{5 \text{ Mbps}}{2} = 2.5 \text{ MHz} \]

The output wave from each balanced modulator is

\[ 0.5 \cos 2\pi(f_c - f_a)t - 0.5 \cos 2\pi(f_c + f_a)t \]

\[ 0.5 \cos 2\pi(70 - 2.5)\text{MHz}t - 0.5 \cos 2\pi(70 - 2.5)\text{MHz}t \]

\[ 0.5 \cos 2\pi(67.5)\text{MHz}t - 0.5 \cos 2\pi(72.5)\text{MHz}t \]

The minimum Nyquist bandwidth is

\[ B = (72.5 - 67.5)\text{MHz} = 5\text{MHz} \]

The symbol rate equals the bandwidth: thus,

symbol rate = 5 megabaud
The output spectrum is as follows:

![Output Spectrum Diagram]

It can be seen that for the same input bit rate the minimum bandwidth required to pass the output of the QPSK modulator is equal to one-half of that required for the BPSK modulator in Example 2-4. Also, the baud rate for the QPSK modulator is one-half that of the BPSK modulator.

The minimum bandwidth for the QPSK system described in Example 2-6 can also be determined by simply substituting into Equation 2-10:

\[ B = \frac{10 \text{ Mbps}}{2} = 5 \text{ MHz} \]

2-5-2-3 (QPSK receiver).

The block diagram of a QPSK receiver is shown in Figure 2-21. The power splitter directs the input QPSK signal to the I and Q product detectors and the carrier recovery circuit. The carrier recovery circuit reproduces the original transmit carrier oscillator signal. The recovered carrier must be frequency and phase coherent with the transmit reference carrier. The QPSK signal is demodulated in the I and Q product detectors, which generate the original I and Q data bits. The outputs of the product detectors are fed to the bit combining circuit, where they are converted from parallel I and Q data channels to a single binary output data stream.
The incoming QPSK signal may be any one of the four possible output phases shown in Figure 2-18. To illustrate the demodulation process, let the incoming QPSK signal be \(-\sin \omega_c t + \cos \omega_c t\). Mathematically, the demodulation process is as follows.

\[ I = (\sin \omega_c t + \cos \omega_c t)(\sin \omega_c t) \]

\[ = (-\sin \omega_c t)(\sin \omega_c t) + (\cos \omega_c t)(\sin \omega_c t) \]

\[ = -\sin^2 \omega_c t + (\cos \omega_c t)(\sin \omega_c t) \]

\[ = -\frac{1}{2}(1 - \cos 2\omega_c t) + \frac{1}{2}\sin(\omega_c + \omega_c)t + \frac{1}{2}\sin(\omega_c - \omega_c)t \]

\[ = -\frac{1}{2} + \frac{1}{2}\cos 2\omega_c t + \frac{1}{2}\sin 2\omega_c t + \frac{1}{2}\sin 0 \]

\[ = -\frac{1}{2} V \text{ (logic 0)} \quad (2.23) \]

FIGURE 2-21 QPSK receiver

The receive QPSK signal \((-\sin \omega_c t + \cos \omega_c t)\) is one of the inputs to the I product detector. The other input is the recovered carrier \((\sin \omega_c t)\). The output of the I product detector is

\[ I = (\sin \omega_c t + \cos \omega_c t)(\sin \omega_c t) \]

\[ = (-\sin \omega_c t)(\sin \omega_c t) + (\cos \omega_c t)(\sin \omega_c t) \]

\[ = -\sin^2 \omega_c t + (\cos \omega_c t)(\sin \omega_c t) \]

\[ = -\frac{1}{2}(1 - \cos 2\omega_c t) + \frac{1}{2}\sin(\omega_c + \omega_c)t + \frac{1}{2}\sin(\omega_c - \omega_c)t \]

\[ = -\frac{1}{2} + \frac{1}{2}\cos 2\omega_c t + \frac{1}{2}\sin 2\omega_c t + \frac{1}{2}\sin 0 \]

\[ = -\frac{1}{2} V \text{ (logic 0)} \]
Again, the receive QPSK signal \((-\sin \omega_c t + \cos \omega_c t)\) is one of the inputs to the Q product detector. The other input is the recovered carrier shifted \(90^\circ\) in phase \((\cos \omega_c t)\). The output of the Q product detector is

\[
Q = \left( -\sin \omega_c t + \cos \omega_c t \right) \left( \cos \omega_c t \right)
\]

\[
= \cos^2 \omega_c t - (\sin \omega_c t)(\cos \omega_c t)
\]

\[
= \frac{1}{2} (1 + \cos 2\omega_c t) - \frac{1}{2} \sin(\omega_c + \omega_c) t - \frac{1}{2} \sin(\omega_c - \omega_c) t
\]

\[
Q = \frac{1}{2} + \frac{1}{2} \cos 2\omega_c t - \frac{1}{2} \sin 2\omega_c t - \frac{1}{2} \sin 0
\]

\[
= \frac{1}{2} V(\text{logic 1})
\] (2.24)

The demodulated I and Q bits (0 and 1, respectively) correspond to the constellation diagram and truth table for the QPSK modulator shown in Figure 2-18.

**2-5-2-4 Offset QPSK.**

*Offset QPSK* (OQPSK) is a modified form of QPSK where the bit waveforms on the I and Q channels are offset or shifted in phase from each other by one-half of a bit time.
Because changes in the I channel occur at the midpoints of the Q channel bits and vice versa, there is never more than a single bit change in the dibit code and, therefore, there is never more than a 90° shift in the output phase. In conventional QPSK, a change in the input dibit from 00 to 11 or 01 to 10 causes a corresponding 180° shift in the output phase.

Therefore, an advantage of OQPSK is the limited phase shift that must be imparted during modulation.

A disadvantage of OQPSK is that changes in the output phase occur at twice the data rate in either the I or Q channel.

Consequently, with OQPSK the baud and minimum bandwidth are twice that of conventional QPSK for a given transmission bit rate. OQPSK is sometimes called OKQPSK (offset-keyed QPSK).

FIGURE 2-22 Offset keyed (OQPSK): (a) block diagram; (b) bit alignment; (c) constellation diagram
2-5-3 8-PSK

With 8-PSK, three bits are encoded, forming tribits and producing eight different output phases. To encode eight different phases, the incoming bits are encoded in groups of three, called tribits ($2^3 = 8$).

2-5-3-1 8-PSK transmitter.

A block diagram of an 8-PSK modulator is shown in Figure 2-23.

FIGURE 2.23 8-PSK modulator

FIGURE 2-24 I- and Q-channel 2-to-4-level converters: (a) I-channel truth table; (b) D-channel truth table; (c) PAM levels
The bit rate in each of the three channels is $f_b/3$.

The bits in the I and C channels enter the I channel 2-to-4-level converter and the bits in the Q and C channels enter the Q channel 2-to-4-level converter.

Essentially, the 2-to-4-level converters are parallel-input digital-to-analog converter (DACs). With two input bits, four output voltages are possible.

The I or Q bit determines the polarity of the output analog signal (logic 1 = +V and logic 0 = -V), whereas the C or C bit determines the magnitude (logic 1 = 1.307 V and logic 0 = 0.541 V).

Figure 2-24 shows the truth table and corresponding output conditions for the 2-to-4-level converters. Because the C and C bits can never be the same logic state, the outputs from the I and Q 2-to-4-level converters can never have the same magnitude, although they can have the same polarity. The output of a 2-to-4-level converter is an M-ary, pulse-amplitude-modulated (PAM) signal where $M = 4$.

**Example 2-7**

For a tribit input of $Q = 0$, $I = 0$, and $C = 0$ (000), determine the output phase for the S-PSK modulator shown in Figure 2-23.

Solution

The inputs to the I channel 2-to-4-level converter are $I = 0$ and $C = 0$. From Figure 2-24 the output is -0.541 V. The inputs to the Q channel 2-to-4-level converter are $Q = 0$ and $C = 1$.

Again from Figure 2-24, the output is 1.307 V.

Thus, the two inputs to the I channel product modulators are -0.541
and sin $\omega_c t$. The output is

$$I = (-0.541)(\sin \omega_c t) = -0.541 \sin \omega_c t$$

The two inputs to the Q channel product modulator are -1.307 V and $\cos \omega_c t$. The output is

$$Q = (-1.307)(\cos \omega_c t) = -1.307 \cos \omega_c t$$

The outputs of the I and Q channel product modulators are combined in the linear summer and produce a modulated output of

$$\text{summer output} = -0.541 \sin \omega_c t - 1.307 \cos \omega_c t$$

$$= 1.41 \sin(\omega_c t - 112.5^\circ)$$

For the remaining tribit codes (001, 010, 011, 100, 101, 110, and 111), the procedure is the same. The results are shown in Figure 2-25.
FIGURE 2-25 8-PSK modulator: (a) truth table; (b) phasor diagram; (c) constellation diagram.
From Figure 2-25, it can be seen that the angular separation between any two adjacent phasors is 45°, half what it is with QPSK.

Therefore, an 8-PSK signal can undergo almost a ± 22.5° phase shift during transmission and still retain its integrity. Also, each phasor is of equal magnitude; the tribit condition (actual information) is again contained only in the phase of the signal.

The PAM levels of 1.307 and 0.541 are relative values. Any levels may be used as long as their ratio is 0.541/1.307 and their arc tangent is equal to 22.5°. For example, if their values were doubled to 2.614 and 1.082, the resulting phase angles would not change, although the magnitude of the phasor would increase proportionally.

Figure 2-26 shows the output phase-versus-time relationship of an 8-PSK modulator.

![Figure 2-26 Output phase-versus-time relationship for an 8-PSK modulator](image)

**2-5-3-2 Bandwidth considerations of 8-PSK.**

With 8-PSK, because the data are divided into three channels, the bit rate in the I, Q, or C channel is equal to one-third of the binary input data rate ($f_b/3$).
\[ \theta = (X \sin \omega_d t)(\sin \omega_c t) \] (2.25)

where

\[
\omega_d t = 2\pi \frac{f_b}{6} t \quad \text{and} \quad \omega_c t = 2\pi f_c t
\]

modulating signal \hspace{1cm} \text{carrier}

And

\[ X = \pm 1.307 \text{ or } \pm 0.541 \]

Thus

\[
\theta = \left( X \sin 2\pi \frac{f_b}{6} t \right) (\sin 2\pi f_c t)
\]

\[ = \frac{X}{2} \cos 2\pi \left( f_c - \frac{f_b}{6} \right) t - \frac{X}{2} \cos 2\pi \left( f_c + \frac{f_b}{6} \right) t \]
Figure 2-27 shows that the highest fundamental frequency in the I, Q, or C channel is equal to one-sixth the bit rate of the binary input (one cycle in the I, Q, or C channel takes the same amount of time as six input bits).

With an 8-PSK modulator, there is one change in phase at the output for every three data input bits. Consequently, the baud for 8 PSK equals \( f_b / 3 \), the same as the minimum bandwidth. Again, the balanced modulators are product modulators; their outputs are the product of the carrier and the PAM signal.
Mathematically, the output of the balanced modulators is

The output frequency spectrum extends from \( f_c + \frac{f_b}{6} \) to \( f_c - \frac{f_b}{6} \), and the minimum bandwidth \( (f_N) \) is

\[
\left( f_c + \frac{f_b}{6} \right) - \left( f_c - \frac{f_b}{6} \right) = \frac{2f_b}{6} = \frac{f_b}{3}
\]

**Example 2-8**

For an 8-PSK modulator with an input data rate \( (f_b) \) equal to 10 Mbps and a carrier frequency of 70 MHz, determine the minimum double-sided Nyquist bandwidth \( (f_N) \) and the baud. Also, compare the results with those achieved with the BPSK and QPSK modulators in Examples 2-4 and 2-6. If the 8-PSK block diagram shown in Figure 2-23 as the modulator model.

**Solution**

The bit rate in the I, Q, and C channels is equal to one-third of the input bit rate, or 10 Mbps

\[ f_{bc} = f_{bQ} = f_{b1} = \frac{10 \text{ Mbps}}{3} = 3.33 \text{ Mbps} \]

Therefore, the fastest rate of change and highest fundamental frequency presented to either balanced modulator is

\[ f_a = \frac{f_{bc}}{2} = \frac{3.33 \text{ Mbps}}{2} = 1.667 \text{ Mbps} \]

The output wave from the balance modulators is \((\sin 2\pi f_a t)(\sin 2\pi f_c t)\)

\[
0.5 \cos 2\pi (f_c - f_a)t - 0.5 \cos 2\pi (f_c + f_a)t
\]

\[
0.5 \cos 2\pi [(70 - 1.667)\text{MHz}]t - 0.5 \cos 2\pi [(70
\]

50
The minimum Nyquist bandwidth is

\[ B = (71.667 - 68.333) \text{ MHz} = 3.333 \text{ MHz} \]

The minimum bandwidth for the 8-PSK can also be determined by simply substituting into Equation 2-10:

\[ B = \frac{10 \text{ Mbps}}{3} = 3.33 \text{ MHz} \]

Again, the baud equals the bandwidth; thus, baud = 3.333 megabaud

The output spectrum is as follows:

It can be seen that for the same input bit rate the minimum bandwidth required to pass the output of an 8-PSK modulator is equal to one-third that of the BPSK modulator in Example 2-4 and 50% less than that required for the QPSK modulator in Example 2-6. Also, in each case the baud has been reduced by the same proportions.
2-5-3-3 8-PSK receiver.

Figure 2-28 shows a block diagram of an 8-PSK receiver. The power splitter directs the input 8-PSK signal to the I and Q product detectors and the carrier recovery circuit.

The carrier recovery circuit reproduces the original reference oscillator signal. The incoming 8-PSK signal is mixed with the recovered carrier in the I product detector and with a quadrature carrier in the Q product detector.

The outputs of the product detectors are 4-level PAM signals that are fed to the 4-to-2-level analog-to-digital converters (ADCs). The outputs from the I channel 4-to-2-level converter are the I and C bits, whereas the outputs from the Q channel 4-to-2-level converter are the Q and C bits. The parallel-to-serial logic circuit converts the I/C and Q/C bit pairs to serial I, Q, and C output data streams.

FIGURE 2-28 8-PSK receiver.
16-PSK is an M-ary encoding technique where $M = 16$; there are 16 different output phases possible. With 16-PSK, four bits (called quadbits) are combined, producing 16 different output phases. With 16-PSK, $n = 4$ and $M = 16$; therefore, the minimum bandwidth and baud equal one-fourth the bit rate ($f_b/4$).

![Figure 2-29](image)

**FIGURE 2-29** 16-PSK: (a) truth table; (b) constellation diagram

Figure 2-29 shows the truth table and constellation diagram for 16-PSK, respectively. Comparing Figures 2-18, 2-25, and 2-29 shows that as the level of encoding increases (i.e., the values of $n$ and $M$ increase), more output phases are possible and the closer each point on the constellation diagram is to an adjacent point. With 16-PSK, the angular separation between adjacent output phases is only $22.5^\circ$ ($180^\circ / 8$). Therefore, 16-PSK can undergo only a $11.25^\circ$ ($180^\circ / 16$) phase shift during transmission and still retain its integrity.

For an M-ary PSK system with 64 output phases ($n = 6$), the angular separation between adjacent phases is only $5.6^\circ$ ($180^\circ / 32$). This is an obvious limitation in the level of encoding (and
bit rates) possible with PSK, as a point is eventually reached where receivers cannot discern the phase of the received signaling element. In addition, phase impairments inherent on communications lines have a tendency to shift the phase of the PSK signal, destroying its integrity and producing errors.

2.6 QUADRATURE – AMPLITUDE MODULATION

2-6-1 8-QAM

8-QAM is an M-ary encoding technique where M = 8. Unlike 8-PSK, the output signal from an 8-QAM modulator is not a constant-amplitude signal.

2-6-1-1 8-QAM transmitter.

Figure 2-30a shows the block diagram of an 8-QAM transmitter. As you can see, the only difference between the 8-QAM transmitter and the 8PSK transmitter shown in Figure 2-23 is the omission of the inverter between the C channel and the Q product modulator. As with 8-PSK, the incoming data are divided into groups of three bits (tribits): the I, Q, and C bit streams, each with a bit rate equal to one-third of the incoming data rate. Again, the I and Q bits determine the polarity of the PAM signal at the output of the 2-to-4-level converters, and the C channel determines the magnitude. Because the C bit is fed uninverted to both the I and the Q channel 2-to-4-level converters, the magnitudes of the I and Q PAM signals are always equal. Their polarities depend on the logic condition of the I and Q bits and, therefore, may be different. Figure 2-30b shows the truth table for the I and Q channel 2-to-4-level converters; they are identical.
Example 2-9

For a tribit input of Q = 0, I = 0, and C = 0 (000), determine the output amplitude and phase for the 8-QAM transmitter shown in Figure 2-30a.

Solution

The inputs to the I channel 2-to-4-level converter are I = 0 and C = 0. From Figure 2-30b, the output is -0.541 V. The inputs to the Q channel 2-to-4-level converter are Q = 0 and C = 0. Again from Figure 2-30b, the output is -0.541 V.

Thus, the two inputs to the I channel product modulator are -0.541 and \( \sin \omega_c t \).

The output is

\[ I = (-0.541)(\sin \omega_c t) = -0.541 \sin \omega_c t. \]

The two inputs to the Q channel product modulator are -0.541 and \( \cos \omega_c t \).

The output is

\[ Q = (-0.541)(\cos \omega_c t) = -0.541 \cos \omega_c t. \]
The outputs from the I and Q channel product modulators are combined in the linear summer and produce a modulated output of

\[
\text{summer output} = -0.541 \sin \omega_c t - 0.541 \cos \omega_c t.
\]

\[
= 0.765 \sin (\cos - 135^\circ)
\]

For the remaining tribit codes (001, 010, 011, 100, 101, 110, and 111), the procedure is the same. The results are shown in Figure 2-31.

Figure 2-32 shows the output phase-versus-time relationship for an 8-QAM modulator. Note that there are two output amplitudes, and only four phases are possible.

![Diagram](image)

**FIGURE 2-31** 8-QAM modulator: (a) truth table; (b) phasor diagram; (c) constellation diagram
2-6-1-2 Bandwidth considerations of 8-QAM.

The minimum bandwidth required for 8-QAM is $f_b/3$, the same as in 8-PSK.

2-6-1-3 8-QAM receiver.

An 8-QAM receiver is almost identical to the 8-PSK receiver shown in Figure 2-28.

2-6-2 16-QAM

As with the 16-PSK, 16-QAM is an M-ary system where $M = 16$. The input data are acted on in groups of four ($2^4 = 16$). As with 8-QAM, both the phase and the amplitude of the transmit carrier are varied.

2-6-2-1 QAM transmitter.

The block diagram for a 16-QAM transmitter is shown in Figure 2-33.
The input binary data are divided into four channels: I, I', Q, and Q'. The bit rate in each channel is equal to one-fourth of the input bit rate (f_b/4).

The I and Q bits determine the polarity at the output of the 2-to-4-level converters (a logic 1 = positive and a logic 0 = negative).

The I' and Q' bits determine the magnitude (a logic 1 = 0.821 V and a logic 0 = 0.22 V).

For the I product modulator they are +0.821 \sin \omega_c t, -0.821 \sin \omega_c t, +0.22 \sin \omega_c t, and -0.22 \sin \omega_c t.

For the Q product modulator, they are +0.821 \cos \omega_c t, +0.22 \cos \omega_c t, -0.821 \cos \omega_c t, and -0.22 \cos \omega_c t.

The linear summer combines the outputs from the I and Q channel product modulators and produces the 16 output conditions necessary for 16-QAM. Figure 2-34 shows the truth table for the I and Q channel 2-to-4-level converters.
Example 2-10

For a quadbit input of I = 0, I' = 0, Q = 0, and Q' = 0 (0000), determine the output amplitude and phase for the 16-QAM modulator shown in Figure 2-33.

Solution

The inputs to the I channel 2-to-4-level converter are I = 0 and I' = 0. From Figure 2-34, the output is -0.22 V. The inputs to the Q channel 2-to-4-level converter are Q = 0 and Q' = 0. Again from Figure 2-34, the output is -0.22 V.

Thus, the two inputs to the I channel product modulator are -0.22 V and sin ω₀t. The output is

\[ I = (-0.22) \sin \omega_0 t = -0.22 \sin \omega_0 t \]

The two inputs to the Q channel product modulator are -0.22 V and cos ω₀t. The output is

\[ Q = (-0.22) \cos \omega_0 t = -0.22 \cos \omega_0 t \]

The outputs from the I and Q channel product modulators are combined in the linear summer and produce a modulated output of

\[ \text{summer output} = -0.22 \sin \omega_0 t - 0.22 \cos \omega_0 t = 0.311 \sin(\omega_0 t - 135^\circ) \]

For the remaining quadbit codes, the procedure is the same. The
results are shown in Figure 2-35.

FIGURE 2-35 16-QAM modulator: (a) truth table; (b) phasor diagram; (c) constellation diagram.
FIGURE 2-36 Bandwidth considerations of a 16-QAM modulator

2-6-2-2 Bandwidth considerations of 16-QAM.

With a 16-QAM, the bit rate in the I, I', Q, or Q' channel is equal to one-fourth of the binary input data rate (f_b/4).

Figure 2-36 shows the bit timing relationship between the binary input data; the I, I', Q, and Q' channel data; and the I PAM signal. It can be seen that the highest fundamental frequency in the I, I', Q, or Q' channel is equal to one-eighth of the bit rate of the binary input data (one cycle in the I, I', Q, or
Q' channel takes the same amount of time as eight input bits). Also, the highest fundamental frequency of either PAM signal is equal to one-eighth of the binary input bit rate. With a 16-QAM modulator, there is one change in the output signal (either its phase, amplitude, or both) for every four input data bits. Consequently, the baud equals \( f_b / 4 \), the same as the minimum bandwidth.

Again, the balanced modulators are product modulators and their outputs can be represented mathematically as

\[
\text{output} = (X \sin \omega_a t)(\sin \omega_c t) \tag{2.26}
\]

where

\[
\omega_a t = 2\pi \frac{f_b}{8} t \quad \text{and} \quad \omega_c t = 2\pi f_b t
\]

and

\[
X = \pm 0.22 \text{ or } \pm 0.821
\]

Thus,

\[
\text{output} = \left( X \sin 2\pi \frac{f_b}{8} t \right) \left( \sin 2\pi f_c t \right) = \frac{X}{2} \cos 2\pi \left( f_c - \frac{f_b}{8} \right) t = \frac{X}{2} \cos 2\pi \left( f_c + \frac{f_b}{8} \right) t
\]

The output frequency spectrum extends from \( f_c + f_b / 8 \) and \( f_c - f_b / 8 \) the minimum bandwidth (\( f_b \)) is

\[
\left( f_c + \frac{f_b}{8} \right) - \left( f_c - \frac{f_b}{8} \right) = \frac{2f_b}{8} = \frac{f_b}{4}
\]
Example 2 -11

For a 16-QAM modulator with an input data rate \(f_b\) equal to 10 Mbps and a carrier frequency of 70 MHz, determine the minimum double-sided Nyquist frequency \(f_{N}\) and the baud. Also, compare the results with those achieved with the BPSK, QPSK, and 8-PSK modulators in Examples 2-4, 2-6, and 2-8. Use the 16-QAM block diagram shown in Figure 2-33 as the modulator model.

Solution

The bit rate in the I, I’, Q, and Q’ channels is equal to one-fourth of the input bit rate,

\[f_{bI} = f_{bI'} = f_{bQ} = f_{bQ'} = f_b / 4 = 10 \text{ Mbps} / 4 = 2.5 \text{ Mbps}\]

Therefore, the fastest rate of change and highest fundamental frequency presented to either balanced modulator is

\[f_a = f_{bI} / 2 = 2.5 \text{ Mbps} / 2 = 1.25 \text{ MHz}\]

The output wave from the balanced modulator is

\[(\sin 2\pi f_{a} t)(\sin 2\pi f_{c} t)\]

\[0.5 \cos 2\pi (f_{c} - f_{a}) t - 0.5 \cos 2\pi (f_{c} + f_{a}) t\]

\[0.5 \cos 2\pi [(70 - 1.25) \text{ MHz}] t - 0.5 \cos 2\pi [(70 + 1.25) \text{ MHz}] t\]

\[0.5 \cos 2\pi (68.75 \text{ MHz}) t - 0.5 \cos 2\pi (71.25 \text{ MHz}) t\]

The minimum Nyquist bandwidth is
B = (71.25 - 68.75) MHz = 2.5 MHz

The minimum bandwidth for the 16-QAM can also be determined by simply substituting into Equation 2-10:

\[ B = \frac{10 \text{ Mbps}}{4} = 2.5 \text{ MHz}. \]

The symbol rate equals the bandwidth; thus,

symbol rate = 2.5 megabaud

The output spectrum is as follows:

For the same input bit rate, the minimum bandwidth required to pass the output of a 16-QAM modulator is equal to one-fourth that of the BPSK modulator, one-half that of QPSK, and 25% less than with 8-PSK. For each modulation technique, the baud is also reduced by the same proportions.

**Example 2-12**

For the following modulation schemes, construct a table showing the number of bits encoded, number of output conditions, minimum bandwidth, and baud for an information data rate of 12 kbps: QPSK, 8-PSK, 8-QAM, 16-PSK, and 16-QAM.
From Example 2-12, it can be seen that a 12-kbps data stream can be propagated through a narrower bandwidth using either 16-PSK or 16-QAM than with the lower levels of encoding.

Table 2-1 summarizes the relationship between the number of bits encoded, the number of output conditions possible, the minimum bandwidth, and the baud for ASK, FSK, PSK, and QAM.

When data compression is performed, higher data transmission rates are possible for a given bandwidth.

Table 2-1 ASK, FSK, PSK AND QAM summary.

<table>
<thead>
<tr>
<th>Modulation</th>
<th>Encoding Scheme</th>
<th>Outputs Possible</th>
<th>Minimum Bandwidth</th>
<th>Baud</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASK</td>
<td>Single bit</td>
<td>2</td>
<td>( f_b )</td>
<td>( f_b )</td>
</tr>
<tr>
<td>FSK</td>
<td>Single bit</td>
<td>2</td>
<td>( f_b )</td>
<td>( f_b )</td>
</tr>
<tr>
<td>BPSK</td>
<td>Single bit</td>
<td>2</td>
<td>( f_b )</td>
<td>( f_b )</td>
</tr>
<tr>
<td>QPSK</td>
<td>Dibits</td>
<td>4</td>
<td>( f_b \times 12 )</td>
<td>( f_b \times 12 )</td>
</tr>
<tr>
<td>8-PSK</td>
<td>Tribits</td>
<td>8</td>
<td>( f_b \times 13 )</td>
<td>( f_b \times 13 )</td>
</tr>
<tr>
<td>8-QAM</td>
<td>Tribits</td>
<td>8</td>
<td>( f_b \times 13 )</td>
<td>( f_b \times 13 )</td>
</tr>
<tr>
<td>16-QAM</td>
<td>Quadbits</td>
<td>16</td>
<td>( f_b \times 14 )</td>
<td>( f_b \times 14 )</td>
</tr>
<tr>
<td>16-PSK</td>
<td>Quadbits</td>
<td>16</td>
<td>( f_b \times 14 )</td>
<td>( f_b \times 14 )</td>
</tr>
<tr>
<td>32-PSK</td>
<td>Five bits</td>
<td>32</td>
<td>( f_b \times 15 )</td>
<td>( f_b \times 15 )</td>
</tr>
<tr>
<td>32-QAM</td>
<td>Five bits</td>
<td>32</td>
<td>( f_b \times 15 )</td>
<td>( f_b \times 15 )</td>
</tr>
<tr>
<td>64-PSK</td>
<td>Six bits</td>
<td>64</td>
<td>( f_b \times 16 )</td>
<td>( f_b \times 16 )</td>
</tr>
<tr>
<td>64-QAM</td>
<td>Six bits</td>
<td>64</td>
<td>( f_b \times 16 )</td>
<td>( f_b \times 16 )</td>
</tr>
<tr>
<td>128-PSK</td>
<td>Seven bits</td>
<td>128</td>
<td>( f_b \times 17 )</td>
<td>( f_b \times 17 )</td>
</tr>
<tr>
<td>128-QAM</td>
<td>Seven bits</td>
<td>128</td>
<td>( f_b \times 17 )</td>
<td>( f_b \times 17 )</td>
</tr>
</tbody>
</table>

Note: \( f_b \) indicates a magnitude equal to the input bit rate.
2-7 BANDWIDTH EFFICIENCY

Bandwidth efficiency (sometimes called information density or spectral efficiency), often used to compare the performance of one digital modulation technique to another.

Mathematical bandwidth efficiency is

\[
\eta = \frac{\text{transmission bit rate (bps)}}{\text{bits/s}} \times \frac{\text{bits/s}}{\text{minimum bandwidth (Hz)}} \times \frac{\text{Hz}}{\text{Hertz}}
\]

Where \( \eta \) = bandwidth efficiency

Example 2-13

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

Solution

a. Baud is determined by substituting into Equation 2-10,

\[
\text{baud} = \frac{24 \text{ kbps}}{3} = 8000
\]

b. Bandwidth is determined by substituting into Equation 2-11:

\[
B = \frac{24 \text{ kbps}}{3} = 8000
\]

c. Bandwidth efficiency is calculated from Equation 2-27:

\[
\eta = \frac{24,000}{8000} = 3 \text{ bits per second per cycle of bandwidth}
\]

Example 2-14

For 16-PSK and a transmission system with a 10 kHz bandwidth, determine the maximum bit rate.
Solution

The bandwidth efficiency for 16-PSK is 4, which means that four bits can be propagated through the system for each hertz of bandwidth. Therefore, the maximum bit rate is simply the product of the bandwidth and the bandwidth efficiency, or

\[
\text{bit rate} = 4 \times 10,000 = 40,000 \text{ bps}
\]

Table 2-2 ASK, FSK, PSK and QAM summary

<table>
<thead>
<tr>
<th>Modulation</th>
<th>Encoding Scheme</th>
<th>Outputs Possible</th>
<th>Minimum Bandwidth</th>
<th>Baud</th>
<th>$\beta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASK</td>
<td>Single bit</td>
<td>2</td>
<td>$f_b$</td>
<td>$f_b$</td>
<td>1</td>
</tr>
<tr>
<td>FSK</td>
<td>Single bit</td>
<td>2</td>
<td>$f_b$</td>
<td>$f_b$</td>
<td>1</td>
</tr>
<tr>
<td>BPSK</td>
<td>Single bit</td>
<td>2</td>
<td>$f_b$</td>
<td>$f_b$</td>
<td>1</td>
</tr>
<tr>
<td>QPSK</td>
<td>Dibits</td>
<td>4</td>
<td>$f_b/2$</td>
<td>$f_b/2$</td>
<td>2</td>
</tr>
<tr>
<td>8-PSK</td>
<td>Tribits</td>
<td>8</td>
<td>$f_b/3$</td>
<td>$f_b/3$</td>
<td>3</td>
</tr>
<tr>
<td>8-QAM</td>
<td>Tribits</td>
<td>8</td>
<td>$f_b/3$</td>
<td>$f_b/3$</td>
<td>3</td>
</tr>
<tr>
<td>16-PSK</td>
<td>Quadbits</td>
<td>16</td>
<td>$f_b/4$</td>
<td>$f_b/4$</td>
<td>4</td>
</tr>
<tr>
<td>16-QAM</td>
<td>Quadbits</td>
<td>16</td>
<td>$f_b/4$</td>
<td>$f_b/4$</td>
<td>4</td>
</tr>
<tr>
<td>32-PSK</td>
<td>Five bits</td>
<td>32</td>
<td>$f_b/5$</td>
<td>$f_b/5$</td>
<td>5</td>
</tr>
<tr>
<td>64-QAM</td>
<td>Six bits</td>
<td>64</td>
<td>$f_b/6$</td>
<td>$f_b/6$</td>
<td>6</td>
</tr>
</tbody>
</table>

Note: $f_b$ indicates a magnitude equal to the input bit rate.

2-8 DIFFERENTIAL PHASE-SHIFT KEYING

*Differential phase-shift keying* (DPSK) is an alternative form of digital modulation where the binary input information is contained in the difference between two successive signaling elements rather than the absolute phase.

2-8-1 Differential BPSK

2-8-1-I DBPSK transmitter.

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

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

In Figure 2-37b, the first data bit is XNORed with the reference bit. If they are the same, the XNOR output is a logic 1; if they are different, the XNOR output is a logic 0. The balanced modulator operates the same as a conventional BPSK modulator; a logic 1 produces +sin ωc t at the output, and a logic 0 produces -sin ωc t at the output.
2-8-1-2 DBPSK receiver.

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

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

FIGURE 2-38 DBPSK demodulator: (a) block diagram; (b) timing sequence

2-9 PROBABILITY OF ERROR AND BIT ERROR RATE

Probability of error \( P(e) \) and bit error rate (BER) are often used interchangeably

BER is an empirical (historical) record of a system's actual bit error performance.

For example, if a system has a BER of \( 10^{-5} \), this means that in past performance there was one bit error for every
100,000 bits transmitted.

Probability of error is a function of the *carrier-to-noise power ratio* (or, more specifically, the average *energy per bit-to-noise power density ratio*) and the number of possible encoding conditions used (M-ary).

Carrier-to-noise power ratio is the ratio of the average carrier power (the combined power of the carrier and its associated sidebands) to the *thermal noise power* Carrier power can be stated in watts or dBm. where

\[ C_{(\text{dBm})} = 10 \log \left[ C_{(\text{watts})} / 0.001 \right] \]  

(2.28)

Thermal noise power is expressed mathematically as

\[ N = KT B \text{ (watts)} \]  

(2.29)

where

- \( N \) = thermal noise power (watts)
- \( K \) = Boltzmann's proportionality constant (1.38 \( \times \) 10\(^{-23} \) joules per kelvin)
- \( T \) = temperature (kelvin: 0 K=-273° C, room temperature = 290 K)
- \( B \) = bandwidth (hertz)

Stated in dBm,

\[ N_{(\text{dBm})} = 10 \log \left[ KT B / 0.001 \right] \]  

(2.30)

Mathematically, the carrier-to-noise power ratio is

\[ C / N = C / KT B \text{ (unitless ratio)} \]  

(2.31)

where

- \( C \) = carrier power (watts)
- \( N \) = noise power (watts)
Stated in dB, \[ C / N \text{ (dB)} = 10 \log \left( \frac{C}{N} \right) \]

\[ = \frac{C}{N} \text{ (dBm)} \] (2.32)

Energy per bit is simply the energy of a single bit of information. Mathematically, energy per bit is

\[ E_b = C T_b \text{ (J/bit)} \] (2.33)

where

- \( E_b \) = energy of a single bit (joules per bit)
- \( T_b \) = time of a single bit (seconds)
- \( C \) = carrier power (watts)

Stated in dBJ, \[ E_b\text{(dBJ)} = 10 \log E_b \] (2.34)

and because \( T_b = 1/f_b \), where \( f_b \) is the bit rate in bits per second, \( E_b \) can be rewritten as

\[ E_b = \frac{C}{f_b} \text{ (J/bit)} \] (2.35)

Stated in dBJ, \[ E_b\text{(dBJ)} = 10 \log \frac{C}{f_b} \] (2.36)

\[ = 10 \log C - 10 \log f_b \] (2.37)

Noise power density is the thermal noise power normalized to a 1-Hz bandwidth (i.e., the noise power present in a 1-Hz bandwidth). Mathematically, noise power density is

\[ N_o = \frac{N}{B} \text{ (W/Hz)} \] (2.38)

where
\( N_o \) = noise power density (watts per hertz)
\( N \) = thermal noise power (watts)
\( B \) = bandwidth (hertz)

Stated in dBm,
\[
N_o(dBm) = 10 \log \left( \frac{N}{0.001} \right) - 10 \log B
\]  
(2.39)

Combining Equations 2.29 and 2.38 yields
\[
N_o = \frac{KTB}{B} = KT \left( \frac{W}{Hz} \right)
\]  
(2.41)

Stated in dBm,
\[
N_o(dBm) = 10 \log \left( \frac{K}{0.001} \right) + 10 \log T
\]  
(2.42)

Energy per bit-to-noise power density ratio is used to compare two or more digital modulation systems that use different transmission rates (bit rates), modulation schemes (FSK, PSK, QAM), or encoding techniques (M-ary).

Mathematically, \( E_b/N_o \) is
\[
E_b/N_o = \frac{(C/f_b)}{(N/B)}
\]  
(2.43)

where \( E_b/N_o \) is the energy per bit-to-noise power density ratio. Rearranging Equation 2.43 yields the following expression:
\[
E_b/N_o = (C/N) x (B/f_b)
\]  
(2.44)

where
\( E_b/N_o \) = energy per bit-to-noise power density ratio
\( C/N \) = carrier-to-noise power ratio
\( B/f_b \) = noise bandwidth-to-bit rate ratio

Stated in dB,
\[
E_b/N_o (dB) = 10 \log \left( \frac{C}{N} \right) + 10 \log \left( \frac{B}{f_b} \right)
\]  
(2.45)
\[ = 10 \log E_b - 10 \log N_o \quad (2.46) \]

**Example 2-15**

For a QPSK system and the given parameters, determine
a. Carrier power in dBm.
b. Noise power in dBm.
c. Noise power density in dBm.
d. Energy per bit in dB.
e. Carrier-to-noise power ratio in dB.
f. \( E_b/N_0 \) ratio.

\[ C = 10^{-12} \text{ W} \]
\[ F_b = 60 \text{ kbps} \]
\[ N = 1.2 \times 10^{-14} \text{ W} \]
\[ B = 120 \text{ kHz} \]

**Solution**

a. The carrier power in dBm is determined by substituting into Equation 2.28:
\[ C = 10 \log (10^{-12} / 0.001) = -90 \text{ dBm} \]

b. The noise power in dBm is determined by substituting into Equation 2-30:
\[ N = 10 \log [(1.2 \times 10^{-14}) / 0.001] = -109.2 \text{ dBm} \]

c. The noise power density is determined by substituting into Equation 2-40:
\[ N_0 = -109.2 \text{ dBm} - 10 \log 120 \text{ kHz} = -160 \text{ dBm} \]

d. The energy per bit is determined by substituting into equation
2.36:

\[ E_b = 10 \log \left( \frac{10^{-12}}{60 \text{ kbps}} \right) = -167.8 \text{ dBJ} \]

\( e. \) The carrier-to-noise power ratio is determined by substituting into Equation 2.34:

\[ C / N = 10 \log \left( \frac{10^{-12}}{1.2 \times 10^{-14}} \right) = 19.2 \text{ dB} \]

\( f. \) The energy per bit-to-noise density ratio is determined by substituting into Equation 2.45:

\[ E_b / N_0 = 19.2 + 10 \log 120 \text{ kHz} / 60 \text{ kbps} = 22.2 \text{ dB} \]

2-10 ERROR PERFORMANCE

2-10-1 PSK Error Performance

The bit error performance is related to the distance between points on a signal state-space diagram.

For example, on the signal state-space diagram for BPSK shown in Figure 2.39a, it can be seen that the two signal points (logic 1 and logic 0) have maximum separation \( (d) \) for a given power level \( (D) \).

The figure shows a noise vector \( (V_N) \), when combined with the signal vector \( (V_s) \), effectively shifts the phase of the signaling element \( (V_{SE}) \) \( \alpha \) degrees.

If the phase shift exceeds \( +90^\circ \), the signal element is shifted beyond the threshold points into the error region.

For BPSK, it would require a noise vector of sufficient amplitude and phase to produce more than a \( \pm 90^\circ \) phase shift in the signaling element to produce an error.
For PSK systems, the general formula for the threshold points is

\[ TP = \pm \frac{\pi}{M} \]

where \( M \) is the number of signal states.

**FIGURE 2-39 PSK error region: (a) BPSK; (b) QPSK**

The phase relationship between signaling elements for BPSK (i.e., 180° out of phase) is the optimum signaling format, referred to as *antipodal signaling*, and occurs only when two binary signal levels are allowed and when one signal is the exact negative of the other. Because no other bit-by-bit signaling scheme is any better, antipodal performance is often used as a reference for comparison.
The error performance of the other multiphase PSK systems can be compared with that of BPSK simply by determining the relative decrease in error distance between points on a signal state-space diagram.

For PSK, the general formula for the maximum distance between signaling points is given by

\[
\sin \theta = \sin \frac{360^0}{2M} = \frac{(d/2)}{D} 
\]  

(2.48)

d = error distance  
M = number of phases  
D = peak signal amplitude

Rearranging equation 2.48 and solving for \(d\) yields

\[
d = \sin \left(\frac{180}{M} \right) \cdot xD 
\]  

(2.49)

Figure 2-39b shows the signal state-space diagram for QPSK. From Figure 2-39 and Equation 2.48, it can be seen that QPSK can tolerate only a ±45° phase shift.

From Equation 2.47 the maximum phase shift for 8-PSK and 16-PSK is ±22.5° and ±11.25°, respectively.

The higher the level of modulation, the smaller the angular separation between signal points and the smaller the error distance.

The general expression for the bit error probability of an M-phase PSK system is

\[
P(e) = \left(\frac{1}{\log_2M}\right) \text{erf} (z) 
\]  

(2.50)

where \(\text{erf} = \text{error function}\)
By substituting into Equation 2.50, it can be shown that QPSK provides the same error performance as BPSK. This is because the 3-dB reduction in error distance for QPSK is offset by the 3-dB decrease in its bandwidth (in addition to the error distance, the relative widths of the noise bandwidths must also be considered).

Thus, both systems provide optimum performance. Figure 2-40 shows the error performance for 2-, 4-, 8-, 16-, and 32-PSK systems as a function of $E_b / N_0$. 

FIGURE 2-40 Error rates of PSK modulation systems
Example 2-16

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

Solution

From Figure 2-40, the minimum $E_b / N_o$ ratio to achieve a $P(e)$ of $10^{-7}$ for an 8-PSK system is 14.7 dB. The minimum bandwidth is found by rearranging Equation 2.44:

$$B / f_b = E_b / N_o = C / N$$

$$= 14.7 \text{ dB} - 11.7 \text{ dB} = 3 \text{ dB}$$

$$B / f_b = \text{antilog} 3 = 2 ; B = 2 \times 10 \text{ Mbps} = 20 \text{ MHz}$$

2-10-2 QAM Error Performance

For a large number of signal points (i.e., M-ary systems greater than 4), QAM outperforms PSK. This is because the distance between signaling points in a PSK system is smaller than the distance between points in a comparable QAM system. The general expression for the distance between adjacent signaling points for a QAM system with L levels on each axis is

$$d = \frac{\sqrt{2}}{L - 1}$$

(2.51)

where $d =$ error distance

$L =$ number of levels on each axis

$D =$ peak signal amplitude
In comparing Equation 2-49 to Equation 2-51, it can be seen that QAM systems have an advantage over PSK systems with the same peak signal power level. The general expression for the bit error probability of an L-level QAM system is

\[
P(e) = \frac{1}{\log_2 L} \frac{L - 1}{L} \text{erfc}(z)
\]

(2.52)

Where \( \text{erfc}(z) \) is the complementary error function.

Figure 2-41 shows the error performance for 4-, 16-, 32-, and 64-QAM systems as a function of \( E_b/N_0 \).

Table 2-4 lists the minimum carrier-to-noise power ratios and energy per bit-to-noise power density ratios required for a probability of error \( 10^{-6} \) for several PSK and QAM modulation schemes.

**Example 2-17**

Which system requires the highest \( E_b/N_0 \) ratio for a probability of error of \( 10^{-6} \), a four-level QAM system or an 8-PSK system?

**Solution**

From Figure 2-41, the minimum \( E_b/N_0 \) ratio required for a four-level QAM system is 10.6 dB. From Figure 2-40, the minimum \( E_b/N_0 \) ratio required for an 8-PSK system is 14 dB. Therefore, to achieve a \( P(e) \) of \( 10^{-6} \), a four-level QAM system would require 3.4 dB less \( E_b/N_0 \) ratio.
FIGURE 2-41 Error rates of QAM modulation systems.

Table 2-4 Performance Comparison of various digital modulation schemes (BER = 10^6)

<table>
<thead>
<tr>
<th>Modulation Technique</th>
<th>C/N Ratio (dB)</th>
<th>$E_p/N_0$ Ratio (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BPSK</td>
<td>10.6</td>
<td>10.6</td>
</tr>
<tr>
<td>QPSK</td>
<td>13.6</td>
<td>10.6</td>
</tr>
<tr>
<td>4-QAM</td>
<td>13.6</td>
<td>10.6</td>
</tr>
<tr>
<td>8-QAM</td>
<td>17.6</td>
<td>10.6</td>
</tr>
<tr>
<td>8-PSK</td>
<td>18.5</td>
<td>14.0</td>
</tr>
<tr>
<td>16-PSK</td>
<td>24.3</td>
<td>18.3</td>
</tr>
<tr>
<td>16-QAM</td>
<td>20.5</td>
<td>14.5</td>
</tr>
<tr>
<td>32-QAM</td>
<td>24.4</td>
<td>17.4</td>
</tr>
<tr>
<td>64-QAM</td>
<td>26.6</td>
<td>18.8</td>
</tr>
</tbody>
</table>

2-10-3 FSK Error Performance

With noncoherent FSK, the transmitter and receiver are not frequency or phase synchronized. With coherent FSK, local
receiver reference signals are in frequency and phase lock with the transmitted signals. The probability of error for noncoherent FSK is

\[ P(e) = \frac{1}{\sqrt{2\pi N_0}} e^{-\frac{E_b}{2N_0}} \]  

(2.53)

The probability of error for coherent FSK is

\[ P(e) = \text{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right) \]  

(2.54)

Figure 2-42 shows probability of error curves for both coherent and noncoherent FSK for several values of \( E_b/N_0 \). From Equations 2-53 and 2-54, it can be determined that the probability of error for noncoherent FSK is greater than that of coherent FSK for equal energy per bit-to-noise power density ratios.
CHAPTER-V

Error Detection and Correction
Special Instructional Objectives:

On completion of this lesson, the student will be able to:

- Explain the need for error detection and correction
- State how simple parity check can be used to detect error
- Explain how two-dimensional parity check extends error detection capability
- State how checksum is used to detect error
- Explain how cyclic redundancy check works
- Explain how Hamming code is used to correct error

3.2.1 Introduction

Environmental interference and physical defects in the communication medium can cause random bit errors during data transmission. Error coding is a method of detecting and correcting these errors to ensure information is transferred intact from its source to its destination. Error coding is used for fault tolerant computing in computer memory, magnetic and optical data storage media, satellite and deep space communications, network communications, cellular telephone networks, and almost any other form of digital data communication. Error coding uses mathematical formulas to encode data bits at the source into longer bit words for transmission. The "code word" can then be decoded at the destination to retrieve the information. The extra bits in the code word provide redundancy that, according to the coding scheme used, will allow the destination to use the decoding process to determine if the communication medium introduced errors and in some cases correct them so that the data need not be retransmitted. Different error coding schemes are chosen depending on the types of errors expected, the communication medium's expected error rate, and whether or not data retransmission is possible. Faster processors and better communications technology make more complex coding schemes, with better error detecting and correcting capabilities, possible for smaller embedded systems, allowing for more robust communications. However, tradeoffs between bandwidth and coding overhead, coding complexity and allowable coding delay between transmissions, must be considered for each application.

Even if we know what type of errors can occur, we can’t simple recognize them. We can do this simply by comparing this copy received with another copy of intended transmission. In this mechanism the source data block is sent twice. The receiver compares them with the help of a comparator and if those two blocks differ, a request for re-transmission is made. To achieve forward error correction, three sets of the same data block are sent and majority decision selects the correct block. These methods are very inefficient and increase the traffic two or three times. Fortunately there are more efficient error detection and correction codes. There are two basic strategies for dealing with errors. One way is to include enough redundant information (extra bits are introduced into the data stream at the transmitter on a regular and logical basis) along with each block of data sent to enable the receiver to deduce what the transmitted character must have been. The other way is to include only enough redundancy to allow the receiver to deduce that error has occurred, but not which error has occurred and the receiver asks for
a retransmission. The former strategy uses Error-Correcting Codes and latter uses Error-detecting Codes.

To understand how errors can be handled, it is necessary to look closely at what error really is. Normally, a frame consists of \( m \) data bits (i.e., message bits) and \( r \) redundant bits (or check bits). Let the total number of bits be \( n \) (\( m + r \)). An \( n \)-bit unit containing data and check-bits is often referred to as an n-bit codeword.

Given any two code-words, say 10010101 and 11010100, it is possible to determine how many corresponding bits differ, just EXCLUSIVE OR the two code-words, and count the number of 1’s in the result. The number of bits position in which code words differ is called the Hamming distance. If two code words are a Hamming distance \( d \)-apart, it will require \( d \) single-bit errors to convert one code word to other. The error detecting and correcting properties depends on its Hamming distance.

- To detect \( d \) errors, you need a distance \((d+1)\) code because with such a code there is no way that \( d \)-single bit errors can change a valid code word into another valid code word. Whenever receiver sees an invalid code word, it can tell that a transmission error has occurred.
- Similarly, to correct \( d \) errors, you need a distance \((2d+1)\) code because that way the legal code words are so far apart that even with \( d \) changes, the original codeword is still closer than any other code-word, so it can be uniquely determined.

First, various types of errors have been introduced in Sec. 3.2.2 followed by different error detecting codes in Sec. 3.2.3. Finally, error correcting codes have been introduced in Sec. 3.2.4.

### 3.2.2 Types of errors

These interferences can change the timing and shape of the signal. If the signal is carrying binary encoded data, such changes can alter the meaning of the data. These errors can be divided into two types: Single-bit error and Burst error.

**Single-bit Error**

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

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
</table>

*Sent*

Single bit change (1 is changed to 0)

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
</table>

*Received*

**Figure 3.2.1** Single bit error
Single bit errors are least likely type of errors in serial data transmission. To see why, imagine a sender sends data at 10 Mbps. This means that each bit lasts only for 0.1 μs (micro-second). For a single bit error to occur noise must have duration of only 0.1 μs (micro-second), which is very rare. However, a single-bit error can happen if we are having a parallel data transmission. For example, if 16 wires are used to send all 16 bits of a word at the same time and one of the wires is noisy, one bit is corrupted in each word.

**Burst Error**

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

![Sent and Received Bits](image)

**Length of burst (6 bits)**

**Figure 3.2.2 Burst Error**

Burst errors are mostly likely to happen in serial transmission. The duration of the noise is normally longer than the duration of a single bit, which means that the noise affects data; it affects a set of bits as shown in Fig. 3.2.2. The number of bits affected depends on the data rate and duration of noise.

**3.2.3 Error Detecting Codes**

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

- Simple Parity check
- Two-dimensional Parity check
- Checksum
- Cyclic redundancy check
3.2.3.1 Simple Parity Checking or One-dimension Parity Check

The most common and least expensive mechanism for error-detection is the simple parity check. In this technique, a redundant bit called **parity bit**, is appended to every data unit so that the number of 1s in the unit (including the parity) becomes even.

Blocks of data from the source are subjected to a check bit or *Parity bit generator* form, where a parity of 1 is added to the block if it contains an odd number of 1’s (ON bits) and 0 is added if it contains an even number of 1’s. At the receiving end the parity bit is computed from the received data bits and compared with the received parity bit, as shown in Fig. 3.2.3. This scheme makes the total number of 1’s even, that is why it is called *even parity checking*. Considering a 4-bit word, different combinations of the data words and the corresponding code words are given in Table 3.2.1.

![Even-parity checking scheme](image-url)

**Figure 3.2.3** Even-parity checking scheme
Table 3.2.1 Possible 4-bit data words and corresponding code words

<table>
<thead>
<tr>
<th>Decimal value</th>
<th>Data Block</th>
<th>Parity bit</th>
<th>Code word</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0000</td>
<td>0</td>
<td>00000</td>
</tr>
<tr>
<td>1</td>
<td>0001</td>
<td>1</td>
<td>00111</td>
</tr>
<tr>
<td>2</td>
<td>0010</td>
<td>1</td>
<td>00101</td>
</tr>
<tr>
<td>3</td>
<td>0011</td>
<td>0</td>
<td>00110</td>
</tr>
<tr>
<td>4</td>
<td>0100</td>
<td>1</td>
<td>01001</td>
</tr>
<tr>
<td>5</td>
<td>0101</td>
<td>0</td>
<td>01010</td>
</tr>
<tr>
<td>6</td>
<td>0110</td>
<td>0</td>
<td>01100</td>
</tr>
<tr>
<td>7</td>
<td>0111</td>
<td>1</td>
<td>01111</td>
</tr>
<tr>
<td>8</td>
<td>1000</td>
<td>1</td>
<td>10001</td>
</tr>
<tr>
<td>9</td>
<td>1001</td>
<td>0</td>
<td>10010</td>
</tr>
<tr>
<td>10</td>
<td>1010</td>
<td>0</td>
<td>10100</td>
</tr>
<tr>
<td>11</td>
<td>1011</td>
<td>1</td>
<td>10111</td>
</tr>
<tr>
<td>12</td>
<td>1100</td>
<td>0</td>
<td>11000</td>
</tr>
<tr>
<td>13</td>
<td>1101</td>
<td>1</td>
<td>11011</td>
</tr>
<tr>
<td>14</td>
<td>1110</td>
<td>1</td>
<td>11101</td>
</tr>
<tr>
<td>15</td>
<td>1111</td>
<td>0</td>
<td>11110</td>
</tr>
</tbody>
</table>

Note that for the sake of simplicity, we are discussing here the even-parity checking, where the number of 1’s should be an even number. It is also possible to use odd-parity checking, where the number of 1’s should be odd.

**Performance**

An observation of the table reveals that to move from one code word to another, at least two data bits should be changed. Hence these set of code words are said to have a minimum distance (hamming distance) of 2, which means that a receiver that has knowledge of the code word set can detect all single bit errors in each code word. However, if two errors occur in the code word, it becomes another valid member of the set and the decoder will see only another valid code word and know nothing of the error. Thus errors in more than one bit cannot be detected. In fact it can be shown that a single parity check code can detect only odd number of errors in a code word.

3.2.3.2 Two-dimension Parity Check

Performance can be improved by using two-dimensional parity check, which organizes the block of bits in the form of a table. Parity check bits are calculated for each row, which is equivalent to a simple parity check bit. Parity check bits are also calculated for all columns then both are sent along with the data. At the receiving end these are compared with the parity bits calculated on the received data. This is illustrated in Fig. 3.2.4.
Two-Dimension Parity Checking increases the likelihood of detecting burst errors. As we have shown in Fig. 3.2.4 that a 2-D Parity check of $n$ bits can detect a burst error of $n$ bits. A burst error of more than $n$ bits is also detected by 2-D Parity check with a high-probability. There is, however, one pattern of error that remains elusive. If two bits in one data unit are damaged and two bits in exactly same position in another data unit are also damaged, the 2-D Parity check checker will not detect an error. For example, if two data units: 11001100 and 10101100. If first and second from last bits in each of them is changed, making the data units as 01001110 and 00101110, the error cannot be detected by 2-D Parity check.

### 3.2.3.3 Checksum

In checksum error detection scheme, the data is divided into $k$ segments each of $m$ bits. In the sender’s end the segments are added using 1’s complement arithmetic to get the sum. The sum is complemented to get the checksum. The checksum segment is sent along with the data segments as shown in Fig. 3.2.5 (a). At the receiver’s end, all received segments are added using 1’s complement arithmetic to get the sum. The sum is complemented. If the result is zero, the received data is accepted; otherwise discarded, as shown in Fig. 3.2.5 (b).

**Performance**

The checksum detects all errors involving an odd number of bits. It also detects most errors involving even number of bits.
3.2.3.4 Cyclic Redundancy Checks (CRC)

This Cyclic Redundancy Check is the most powerful and easy to implement technique. Unlike checksum scheme, which is based on addition, CRC is based on binary division. In CRC, a sequence of redundant bits, called **cyclic redundancy check bits**, are appended to the end of data unit so that the resulting data unit becomes exactly divisible by a second, predetermined binary number. At the destination, the incoming data unit is divided by the same number. If at this step there is no remainder, the data unit is assumed to be correct and is therefore accepted. A remainder indicates that the data unit has been damaged in transit and therefore must be rejected. The generalized technique can be explained as follows.

If a $k$ bit message is to be transmitted, the transmitter generates an $r$-bit sequence, known as **Frame Check Sequence** (FCS) so that the $(k+r)$ bits are actually being transmitted. Now this $r$-bit FCS is generated by dividing the original number, appended by $r$ zeros, by a predetermined number. This number, which is $(r+1)$ bit in length, can also be considered as the coefficients of a polynomial, called **Generator Polynomial**. The remainder of this division process generates the $r$-bit FCS. On receiving the packet, the receiver divides the $(k+r)$ bit frame by the same predetermined number and if it produces no remainder, it can be assumed that no error has occurred during the transmission. Operations at both the sender and receiver end are shown in Fig. 3.2.6.

![Figure 3.2.5](image)

(a) Sender’s end for the calculation of the checksum, (b) Receiving end for checking the checksum
This mathematical operation performed is illustrated in Fig. 3.2.7 by dividing a sample 4-bit number by the coefficient of the generator polynomial $x^3+x+1$, which is 1011, using the modulo -2 arithmetic. Modulo-2 arithmetic is a binary addition process without any carry over, which is just the Exclusive-OR operation. Consider the case where $k=1101$. Hence we have to divide 1101000 (i.e. $k$ appended by 3 zeros) by 1011, which produces the remainder $r=001$, so that the bit frame $(k+r) =1101001$ is actually being transmitted through the communication channel. At the receiving end, if the received number, i.e., 1101001 is divided by the same generator polynomial 1011 to get the remainder as 000, it can be assumed that the data is free of errors.

```
1 1 1 1
1 0 1 1
1 1 0 1 0 0 0
  1 0 1 1
  1 1 0 0
  1 0 1 1
  1 1 1 0
  1 0 1 1
  1 0 1 0
  1 0 1 1
  0 0 1
```

**Figure 3.2.7** Cyclic Redundancy Checks (CRC)
The transmitter can generate the CRC by using a feedback shift register circuit. The same circuit can also be used at the receiving end to check whether any error has occurred. All the values can be expressed as polynomials of a dummy variable X. For example, for $P = 11001$ the corresponding polynomial is $X^4 + X^3 + 1$. A polynomial is selected to have at least the following properties:

1. It should not be divisible by $X$.
2. It should not be divisible by $(X+1)$.

The first condition guarantees that all burst errors of a length equal to the degree of polynomial are detected. The second condition guarantees that all burst errors affecting an odd number of bits are detected.

CRC process can be expressed as $X^n M(X)/P(X) = Q(X) + R(X)/P(X)$

Commonly used divisor polynomials are:
- CRC-16 = $X^{16} + X^{15} + X^2 + 1$
- CRC-CCITT = $X^{16} + X^{12} + X^5 + 1$
- CRC-32 = $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + 1$

**Performance**

CRC is a very effective error detection technique. If the divisor is chosen according to the previously mentioned rules, its performance can be summarized as follows:

- CRC can detect all single-bit errors
- CRC can detect all double-bit errors (three 1’s)
- CRC can detect any odd number of errors (X+1)
- CRC can detect all burst errors of less than the degree of the polynomial.
- CRC detects most of the larger burst errors with a high probability.
- For example CRC-12 detects 99.97% of errors with a length 12 or more.

### 3.2.4 Error Correcting Codes

The techniques that we have discussed so far can detect errors, but do not correct them.

**Error Correction** can be handled in two ways.

1. One is when an error is discovered; the receiver can have the sender retransmit the entire data unit. This is known as **backward error correction**.
2. In the other, receiver can use an error-correcting code, which automatically corrects certain errors. This is known as **forward error correction**.

In theory it is possible to correct any number of errors atomically. Error-correcting codes are more sophisticated than error detecting codes and require more redundant bits. The number of bits required to correct multiple-bit or burst error is so high that in most of the...
cases it is inefficient to do so. For this reason, most error correction is limited to one, two or at the most three-bit errors.

3.2.4.1 Single-bit error correction

Concept of error-correction can be easily understood by examining the simplest case of single-bit errors. As we have already seen that a single-bit error can be detected by addition of a parity bit (VRC) with the data, which needed to be send. A single additional bit can detect error, but it’s not sufficient enough to correct that error too. For correcting an error one has to know the exact position of error, i.e. exactly which bit is in error (to locate the invalid bits). For example, to correct a single-bit error in an ASCII character, the error correction must determine which one of the seven bits is in error. To this, we have to add some additional redundant bits.

To calculate the numbers of redundant bits (r) required to correct d data bits, let us find out the relationship between the two. So we have (d+r) as the total number of bits, which are to be transmitted; then r must be able to indicate at least d+r+1 different values. Of these, one value means no error, and remaining d+r values indicate error location of error in each of d+r locations. So, d+r+1 states must be distinguishable by r bits, and r bits can indicates $2^r$ states. Hence, $2^r$ must be greater than d+r+1.

$$2^r \geq d+r+1$$

The value of r must be determined by putting in the value of d in the relation. For example, if d is 7, then the smallest value of r that satisfies the above relation is 4. So the total bits, which are to be transmitted is 11 bits (d+r = 7+4 =11).

Now let us examine how we can manipulate these bits to discover which bit is in error. A technique developed by R.W.Hamming provides a practical solution. The solution or coding scheme he developed is commonly known as Hamming Code. Hamming code can be applied to data units of any length and uses the relationship between the data bits and redundant bits as discussed.

![Positions of redundancy bits in hamming code](image)

**Figure 3.2.8** Positions of redundancy bits in hamming code
Basic approach for error detection by using Hamming code is as follows:

- To each group of m information bits k parity bits are added to form (m+k) bit code as shown in Fig. 3.2.8.
- Location of each of the (m+k) digits is assigned a decimal value.
- The k parity bits are placed in positions 1, 2, …, 2^{k-1} positions.
- K parity checks are performed on selected digits of each codeword.
- At the receiving end the parity bits are recalculated. The decimal value of the k parity bits provides the bit-position in error, if any.

![Hamming Code Diagram](image)

**Figure 3.2.9** Use of Hamming code for error correction for a 4-bit data

Figure 3.2.9 shows how hamming code is used for correction for 4-bit numbers (d4d3d2d1) with the help of three redundant bits (r3r2r1). For the example data 1010, first r1 (0) is calculated considering the parity of the bit positions, 1, 3, 5 and 7. Then the parity bits r2 is calculated considering bit positions 2, 3, 6 and 7. Finally, the parity bits r4 is calculated considering bit positions 4, 5, 6 and 7 as shown. If any corruption occurs in any of the transmitted code 1010010, the bit position in error can be found out by calculating r3r2r1 at the receiving end. For example, if the received code word is 1110010, the recalculated value of r3r2r1 is 110, which indicates that bit position in error is 6, the decimal value of 110.
Example:

Let us consider an example for 5-bit data. Here 4 parity bits are required. Assume that during transmission bit 5 has been changed from 1 to 0 as shown in Fig. 3.2.11. The receiver receives the code word and recalculates the four new parity bits using the same set of bits used by the sender plus the relevant parity (r) bit for each set (as shown in Fig. 3.2.11). Then it assembles the new parity values into a binary number in order of r positions (r8, r4, r2, r1).

Calculations:

Parity recalculated (r8, r4, r2, r1) = 0101₂ = 5₁₀.

Hence, bit 5ᵗʰ is in error i.e. d₅ is in error.
So, correct code-word which was transmitted is:

Figure 3.2.11 Use of Hamming code for error correction for a 5-bit data
LINEAR BLOCK CODES:

We assume that the output of an information source is a sequence of binary digits "0" or "1." In block coding, this binary information sequence is segmented into message blocks of fixed length; each message block, denoted by \( u \), consists of \( k \) information digits. There are a total of \( 2^k \) distinct messages. The encoder, according to certain rules, transforms each input message \( u \) into a binary \( n \)-tuple \( v \) with \( n > k \). This binary \( n \)-tuple \( v \) is referred to as the code word (or code vector) of the message \( u \), as shown in Figure 1.

![Figure 1](image)

Therefore, corresponding to the \( 2^k \) possible messages, there are \( 2^k \) code words. This set of \( 2^k \) code words is called a block code. For a block code to be useful, the \( 2^k \) code words must be distinct. Therefore, there should be a one-to-one correspondence between a message \( u \) and its code word \( v \).
Definition: A block code of length $n$ and $2^k$ code words is called a *linear* $(n, k)$ code if and only if its $2^k$ code words form a $k$-dimensional subspace of the vector space of all the $n$-tuples over the field $GF(2)$.

In fact, a binary block code is linear if and only if the modulo-2 sum of two code words is also a code word. The block code given in Table 1 is a $(7, 4)$ linear code. One can easily check that the sum of any two code words in this code is also a code word.

**TABLE 1**  LINEAR BLOCK CODE WITH $k = 4$ AND $n = 7$

<table>
<thead>
<tr>
<th>Messages</th>
<th>Code words</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(0 0 0 0)$</td>
<td>$(0 0 0 0 0 0 0)$</td>
</tr>
<tr>
<td>$(1 0 0 0)$</td>
<td>$(1 1 0 1 0 0 0)$</td>
</tr>
<tr>
<td>$(0 1 0 0)$</td>
<td>$(0 1 1 0 1 0 0)$</td>
</tr>
<tr>
<td>$(1 1 0 0)$</td>
<td>$(1 0 1 1 1 0 0)$</td>
</tr>
<tr>
<td>$(0 0 1 0)$</td>
<td>$(1 1 1 0 0 1 0)$</td>
</tr>
<tr>
<td>$(1 0 1 0)$</td>
<td>$(0 0 1 1 0 1 0)$</td>
</tr>
<tr>
<td>$(0 1 1 0)$</td>
<td>$(1 0 0 0 1 1 0)$</td>
</tr>
<tr>
<td>$(1 1 1 0)$</td>
<td>$(0 1 0 1 1 1 0)$</td>
</tr>
<tr>
<td>$(0 0 0 1)$</td>
<td>$(1 0 1 0 0 0 1)$</td>
</tr>
<tr>
<td>$(1 0 0 1)$</td>
<td>$(0 1 1 1 0 0 1)$</td>
</tr>
<tr>
<td>$(0 1 0 1)$</td>
<td>$(1 1 0 0 1 0 1)$</td>
</tr>
<tr>
<td>$(1 1 0 1)$</td>
<td>$(0 0 0 1 1 0 1)$</td>
</tr>
<tr>
<td>$(0 0 1 1)$</td>
<td>$(0 1 0 0 0 1 1)$</td>
</tr>
<tr>
<td>$(1 0 1 1)$</td>
<td>$(1 0 0 1 0 1 1)$</td>
</tr>
<tr>
<td>$(0 1 1 1)$</td>
<td>$(0 0 1 0 1 1 1)$</td>
</tr>
<tr>
<td>$(1 1 1 1)$</td>
<td>$(1 1 1 1 1 1 1)$</td>
</tr>
</tbody>
</table>

Since an $(n, k)$ linear code $C$ is a $k$-dimensional subspace of the vector space $V_n$ of all the binary $n$-tuples, it is possible to find $k$ linearly independent code words, go,
g_1, \ldots, g_{k-1} in C such that every code word v in C is a linear combination of these k code words, that is,

\[ v = u_0 g_0 + u_1 g_1 + \cdots + u_{k-1} g_{k-1}, \]

Where \( u_i = 0 \) or \( 1 \) for \( 0 \leq i \leq k \). Let us arrange these \( k \) linearly independent code words as the rows of a \( k \times n \) matrix as follows:

\[
G = \begin{bmatrix}
    g_0 \\
g_1 \\
    \vdots \\
g_{k-1}
\end{bmatrix} = \begin{bmatrix}
g_{00} & g_{01} & g_{02} & \cdots & g_{0,n-1} \\
g_{10} & g_{11} & g_{12} & \cdots & g_{1,n-1} \\
    \vdots & \vdots & \vdots & \ddots & \vdots \\
g_{k-1,0} & g_{k-1,1} & g_{k-1,2} & \cdots & g_{k-1,n-1}
\end{bmatrix},
\]

\[..........(2)\]

where \( g_i = (g_{i0}, g_{i1}, \ldots, g_{i,n-1}) \) for \( 0 \leq i < k \). If \( u = (u_0, u_1, \ldots, u_{k-1}) \) is the message to be encoded, the corresponding code word can be given as follows:

\[ v = u \cdot G \]
Clearly, the rows of $G$ generate (or span) the $(n, k)$ linear code $C$. For this reason, the matrix $G$ is called a generator matrix for $C$. It follows from (3) that an $(n, k)$ linear code is completely specified by the $k$ rows of a generator matrix $G$. Therefore, the encoder has only to store the $k$ rows of $G$ and to form a linear combination of these $k$ rows based on the input message.

**Example 1** The $(7, 4)$ linear code given in Table 1 has the following matrix as a generator matrix.

$$G = \begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ g_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix},$$

If $u = (1 \ 1 \ 0 \ 1)$ is the message to be encoded, its corresponding code word, according to (3), would be
A desirable property for a linear block code to possess is the **systematic structure** of the code words as shown in Figure 2, where a code word is divided into two parts, the message part and the redundant checking part. The message part consists of \( k \) unaltered information (or message) digits and the redundant checking part consists of \( n - k \) parity-check digits, which are linear sums of the information digits. A linear block code with this structure is referred to as a **linear systematic block code**. The \((7, 4)\) code given in Table 1 is a linear systematic block code; the rightmost four digits of each code word are identical to the corresponding information digits.

\[
v = 1 \cdot g_0 + 1 \cdot g_1 + 0 \cdot g_2 + 1 \cdot g_3 \\
= (1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0) + (0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 0) + (1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1) \\
- (0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 1).
\]

![Figure 2](image)

**Figure 2** Systematic format of a code word.

A linear systematic \((n, k)\) code is completely specified by a \(k \times n\) matrix \(G\) of the following form:
EXAMPLE 2: The matrix G given in Example 1 is in systematic form. Let $u = (u_0, u_1, \ldots, u_{k-1})$ be the message to be encoded and let $v = (v_0, v_1, v_2, \ldots, v_{n-1})$ be the corresponding code word. Then 

$$v = (v_0, v_1, v_2, \ldots, v_{n-1}) = (u_0, u_1, \ldots, u_{k-1}) \cdot G.$$
There is another useful matrix associated with every linear block code. For any $k \times n$ matrix $G$ with $k$ linearly independent rows, there exists an $(n-k) \times n$ matrix $H$ with $n-k$ linearly independent rows such that any vector in the row space of $G$ is orthogonal to the rows of $H$ and any vector that is orthogonal to the rows of $H$ is in the row space of $G$. Hence, we can describe the $(n,k)$ linear code generated by $G$ in an alternate way as follows: An $n$-tuple $v$ is a code word in the code generated by $G$ if and only if $v \cdot H^T = 0$. This matrix $H$ is called a parity-check matrix of the code.

\[
v = (u_0, u_1, u_2, u_3) \cdot \begin{bmatrix}
1 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 1 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 1
\end{bmatrix}.
\]

By matrix multiplication, we obtain the following digits of the code word $v$:

\[
\begin{align*}
v_6 &= u_3 \\
v_5 &= u_2 \\
v_4 &= u_1 \\
v_3 &= u_0 \\
v_2 &= u_1 + u_2 + u_3 \\
v_1 &= u_0 + u_1 + u_2 \\
v_0 &= u_0 + u_2 + u_3.
\end{align*}
\]

The code word corresponding to the message $(1 \ 0 \ 1 \ 1)$ is $(1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1)$.
If the generator matrix of an \((n, k)\) linear code is in the systematic form of (4), the parity-check matrix may take the following form:

\[
H = [I_{n-k} \ P^T]
\]

\[
\begin{bmatrix}
1 & 0 & 0 & \cdots & 0 & p_{00} & p_{10} & \cdots & p_{k-1, 0} \\
0 & 1 & 0 & \cdots & 0 & p_{01} & p_{11} & \cdots & p_{k-1, 1} \\
0 & 0 & 1 & \cdots & 0 & p_{02} & p_{12} & \cdots & p_{k-1, 2} \\
& & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
0 & 0 & 0 & \cdots & 1 & p_{0, n-k-1} & p_{1, n-k-1} & \cdots & p_{k-1, n-k-1} \\
\end{bmatrix}
\]

\[\text{...(6)}\]

**Example 2:** Consider the generator matrix of a \((7, 4)\) linear code given in Example 1. The corresponding parity-check matrix is

\[
H = \begin{bmatrix}
1 & 0 & 0 & 1 & 0 & 1 & 1 \\
0 & 1 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 1 & 1 \\
\end{bmatrix}
\]
SYNDROME AND ERROR DETECTION:

Consider an \((n, k)\) linear code with generator matrix \(G\) and parity-check matrix \(H\). Let \(v = (v_0, v_1, \ldots, v_{n-1})\) be a code word that was transmitted over a noisy channel. Let \(r = (r_0, r_1, \ldots, r_{n-1})\) be the received vector at the output of the channel. Because of the channel noise, \(r\) may be different from \(v\). The vector sum

\[
e = r + v
\]

\[
= (e_0, e_1, \ldots, e_{n-1}) \quad \text{...}(7)
\]

Is an n-tuple where \(e_i = 1\) for \(r_i \neq v_i\) and \(e_i = 0\) for \(r_i = v_i\). This n-tuple is called the error vector (or error pattern). The 1's in \(e\) are the transmission errors caused by the channel noise. It follows from (7) that the received vector \(r\) is the vector sum of the transmitted code word and the error vector, that is,

\[
r = v + e.
\]

\[\text{.................}(8)\]

Of course, the receiver does not know either \(v\) or \(e\). Upon receiving \(r\), the decoder must first determine whether \(r\) contains transmission errors. If the presence of errors is detected, the decoder will either take actions to locate the errors and correct them (FEC) or request for a retransmission of (ARQ).

When \(r\) is received, the decoder computes the following \((n-k)\)-tuple:
\[ s = r \cdot H^T \]

\[ = (s_0, s_1, \ldots, s_{n-k-1}). \] ....(9)

Which is called the syndrome of \( r \). Then \( s = 0 \) if and only if \( r \) is a code word, and \( s \neq 0 \) if and only if \( r \) is not a code word. Therefore, when \( s \neq 0 \), we know that \( r \) is not a code word and the presence of errors has been detected.

When \( s = 0 \), \( r \) is a code word and the receiver accepts \( r \) as the transmitted code word. It is possible that the errors in certain error vectors are not detectable (i.e., \( r \) contains errors but \( s = r \cdot H^T = 0 \)). This happens when the error pattern \( e \) is identical to a nonzero code word. In this event, \( r \) is the sum of two code words which is a code word, and consequently \( r \cdot H^T = 0 \).

Error patterns of this kind are called undetectable error patterns. Since there are \( 2^k - 1 \) nonzero code words, there are \( 2^k - 1 \) undetectable error patterns. When an undetectable error pattern occurs, the decoder makes a decoding error.

Based on (6) and (9), the syndrome digits are as follows:
Example 4: Consider the (7, 4) linear code whose parity-check matrix is given in Example 3. Let 
\[ r = (r_0, r_1, r_2, r_3, r_4, r_5, r_6) \] 
be the received vector. Then the syndrome is given by

\[
\begin{align*}
    s_0 &= r_0 + r_{n-k}p_{00} + r_{n-k+1}p_{10} + \cdots + r_{n-1}p_{k-1,0} \\
    s_1 &= r_1 + r_{n-k}p_{01} + r_{n-k+1}p_{11} + \cdots + r_{n-1}p_{k-1,1} \\
    &\vdots \\
    s_{n-k-1} &= r_{n-k-1} + r_{n-k}p_{0,n-k-1} + r_{n-k+1}p_{1,n-k-1} + \cdots + r_{n-1}p_{k-1,n-k-1}.
\end{align*}
\]

\[ \text{.....(10)} \]

\[ s = (s_0, s_1, s_2) \] 

\[
\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1 \\
\end{bmatrix}
\begin{bmatrix}
r_0 \\
r_1 \\
r_2 \\
\end{bmatrix}
\] 

\[ = (r_0, r_1, r_2, r_3, r_4, r_5, r_6)
\begin{bmatrix}
1 & 1 & 0 \\
0 & 1 & 1 \\
1 & 1 & 1 \\
1 & 0 & 1 \\
\end{bmatrix}
\]
The syndrome digits are

\[ s_0 = r_0 + r_3 + r_5 + r_6 \]
\[ s_1 = r_1 + r_3 + r_4 + r_5 \]
\[ s_2 = r_2 + r_4 + r_5 + r_6. \]

The syndrome \( s \) computed from the received vector \( r \) actually depends only on the error pattern \( e \), and not on the transmitted code word \( v \). Since \( r \) is the vector sum of \( v \) and \( e \), it follows from (9) that

\[ s = r \cdot H^T = (v + e)H^T = v \cdot H^T + e \cdot H^T. \]

However, \( v \cdot H^T = 0 \). Consequently, we obtain the following relation between the syndrome and the error pattern:

\[ s = e \cdot H^T. \]

\[ \ldots (11) \]

If the parity-check matrix \( H \) is expressed in the systematic form as given by (6), multiplying out \( e \cdot H^T \) yields the following linear relationship between the syndrome digits and the error digits:
EXAMPLE 5: Again, we consider the (7, 4) code whose parity-check matrix is given in Example 3. Let \( v = (1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1) \) be the transmitted code word and \( r = (1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1) \) be the received vector. Upon receiving \( r \), the receiver computes the syndrome:

\[
\mathbf{s} = \mathbf{r} \cdot \mathbf{H}^T = (1 \ 1 \ 1).
\]

Next, the receiver attempts to determine the true error vector \( \mathbf{e} = (e_0, e_1, e_2, e_3, e_4, e_5, e_6) \), which yields the syndrome above. It follows from (11) or (12) that the error digits are related to the syndrome digits by the following linear equations:

\[
\begin{align*}
1 &= e_0 + e_3 + e_5 + e_6 \\
1 &= e_1 + e_3 + e_4 + e_5 \\
1 &= e_2 + e_4 + e_5 + e_6.
\end{align*}
\]
There are \(2^4 = 16\) error patterns that satisfy the equations above. They are

\[
\begin{align*}
& (0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0), \quad (1 \ 0 \ 1 \ 0 \ 0 \ 1 \ 1), \\
& (1 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0), \quad (0 \ 1 \ 1 \ 1 \ 0 \ 1 \ 1), \\
& (0 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0), \quad (1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 1), \\
& (1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0), \quad (0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1), \\
& (1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0), \quad (0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1), \\
& (0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0), \quad (1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1), \\
& (1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0), \quad (0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1), \\
& (0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0), \quad (1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0).
\end{align*}
\]

The error vector \(e = (0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0)\) has the smallest number of nonzero components. If the channel is a BSC, \(e = (0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0)\) is the most probable error vector that satisfies the equations above. Taking \(e = (0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0)\) as the true error vector, the receiver decodes the received vector \(r = (1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1)\) into the following code word:

\[
v^* = r + e = (1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1) + (0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0) = (1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1).
\]

We see that \(v^*\) is the actual transmitted code word. Hence, the receiver has made a correct decoding. Later we show that the \((7, 4)\) linear code considered in this example is capable of correcting
any single error over a span of seven digits; that is, if a code word is transmitted and if only one digit is changed by the channel noise, the receiver will be able to determine the true error vector and to perform a correct decoding.

**THE MINIMUM DISTANCE OF A BLOCK CODE:**

In this section an important parameter of a block code called the *minimum distance* is introduced. This parameter determines the random-error-detecting and random-error-correcting capabilities of a code. Let \( \mathbf{v} = (v_1, v_2, \ldots, v_{n-1}) \) be a binary \( n \)-tuple. The *Hamming weight* (or simply weight) of \( \mathbf{v} \), denoted by \( w(\mathbf{v}) \), is defined as the number of nonzero components of \( \mathbf{v} \).

For example, the Hamming weight of \( \mathbf{v} = (1 0 0 1 0 1 1) \) is 4. Let \( \mathbf{v} \) and \( \mathbf{w} \) be two \( n \)-tuples. The *Hamming distance* (or simply distance) between \( \mathbf{v} \) and \( \mathbf{w} \), denoted \( d(\mathbf{v}, \mathbf{w}) \), is defined as the number of places where they differ.

For example, the Hamming distance between \( \mathbf{v} = (1 0 0 1 0 1 1) \) and \( \mathbf{w} = (0 1 0 0 1 1 1) \) is 3; they differ in the zeroth, first, and third places.

It follows from the definition of Hamming distance and the definition of modulo-2 addition that the Hamming distance between two \( n \)-tuples, \( \mathbf{v} \) and \( \mathbf{w} \), is equal to the Hamming weight of the sum of \( \mathbf{v} \) and \( \mathbf{w} \), that is,

\[
d(\mathbf{v}, \mathbf{w}) = w(\mathbf{v} + \mathbf{w}).
\]

For example, the Hamming distance between \( \mathbf{v} = (1 0 0 1 0 1 1) \) and \( \mathbf{w} = (1 1 1 0 0 1 0) \) is 4 and the weight of \( \mathbf{v} + \mathbf{w} = (0 1 1 1 0 0 1) \) is also 4.
Given a block code $C$, one can compute the Hamming distance between any two distinct code words. The *minimum distance* of $C$, denoted $d_{\text{min}}$, is defined as

$$d_{\text{min}} = \min \{ d(v, w) : v, w \in C, v \neq w \}.$$ 

....(14) If $C$ is a linear block code, the sum of two vectors is also a code vector. It follows from (13) that the Hamming distance between two code vectors in $C$ is equal to the Hamming weight of a third code vector in $C$. Then it follows from (14) that

$$d_{\text{min}} = \min \{ w(v + w) : v, w \in C, v \neq w \}$$

$$= \min \{ w(x) : x \in C, x \neq 0 \}$$

$$\triangleq w_{\text{min}}.$$ 

.....(15)

The parameter $w_{\text{min}} \{w(x) : x \in C, x \neq 0\}$ is called the *minimum weight* of the linear code $C$.

Summarizing the result above, we have the following:

“The minimum distance of a linear block code is equal to the minimum weight of its nonzero code words”.

Therefore, for a linear block code, to determine the minimum distance of the code is equivalent to determining its minimum weight. The $(7, 4)$ code given in Table 1 has minimum weight 3; thus, its minimum distance is 3.
Convolutional Coding

This lecture introduces a powerful and widely used class of codes, called convolutional codes, which are used in a variety of systems including today’s popular wireless standards (such as 802.11) and in satellite communications. Convolutional codes are beautiful because they are intuitive, one can understand them in many different ways, and there is a way to decode them so as to recover the mathematically most likely message from among the set of all possible transmitted messages. This lecture discusses the encoding; the next one discusses how to decode convolutional codes efficiently.

5.1 Overview

Convolutional codes are a bit like the block codes discussed in the previous lecture in that they involve the transmission of parity bits that are computed from message bits. Unlike block codes in systematic form, however, the sender does not send the message bits followed by (or interspersed with) the parity bits; in a convolutional code, the sender sends only the parity bits.

The encoder uses a sliding window to calculate $r > 1$ parity bits by combining various subsets of bits in the window. The combining is a simple addition in $\mathbb{F}_2$, as in the previous lectures (i.e., modulo 2 addition, or equivalently, an exclusive-or operation). Unlike a block code, the windows overlap and slide by 1, as shown in Figure 8-1. The size of the window, in bits, is called the code’s constraint length. The longer the constraint length, the larger the number of parity bits that are influenced by any given message bit. Because the parity bits are the only bits sent over the channel, a larger constraint length generally implies a greater resilience to bit errors. The trade-off, though, is that it will take considerably longer to decode codes of long constraint length, so one can’t increase the constraint length arbitrarily and expect fast decoding.

If a convolutional code that produces $r$ parity bits per window and slides the window forward by one bit at a time, its rate (when calculated over long messages) is $1/r$. The greater the value of $r$, the higher the resilience of bit errors, but the trade-off is that a proportionally higher amount of communication bandwidth is devoted to coding overhead. In practice, we would like to pick $r$ and the constraint length to be as small as possible
while providing a low enough resulting probability of a bit error.

In 6.02, we will use $K$ (upper case) to refer to the constraint length, a somewhat unfortunate choice because we have used $k$ (lower case) in previous lectures to refer to the number of message bits that get encoded to produce coded bits. Although “L” might be a better way to refer to the constraint length, we’ll use $K$ because many papers and documents in the field use $K$ (in fact, most use $k$ in lower case, which is especially confusing). Because we will rarely refer to a “block” of size $k$ while talking about convolutional codes, we hope that this notation won’t cause confusion.

Armed with this notation, we can describe the encoding process succinctly. The encoder looks at $K$ bits at a time and produces $r$ parity bits according to carefully chosen functions that operate over various subsets of the $K$ bits. One example is shown in Figure 8-1, which shows a scheme with $K = 3$ and $r = 2$ (the rate of this code, $1/r = 1/2$). The encoder spits out $r$ bits, which are sent sequentially, slides the window by 1 to the right, and then repeats the process. That’s essentially it.

At the transmitter, the only remaining details that we have to worry about now are:

- What are good parity functions and how can we represent them conveniently?
- How can we implement the encoder efficiently?

The rest of this lecture will discuss these issues, and also explain why these codes are called “convolutional”.

Parity Equations

The example in Figure 8-1 shows one example of a set of parity equations, which govern the way in which parity bits are produced from the sequence of message bits, $X$. In this example, the equations are as follows (all additions are in $F_2$):

$$p_0[n] = x[n] + x[n-1] + x[n-2]$$
$$p_1[n] = x[n] + x[n-1]$$

(8.1)
By convention, we will assume that each message has $K - I$ "0" bits padded in front, so that the initial conditions work out properly.

An example of parity equations for a rate 1/3 code is

$$p_0[n] = x[n] + x[n-1] + x[n-2]$$
$$p_1[n] = x[n] + x[n-1]$$
$$p_2[n] = x[n] + x[n-2]$$

(8.2)

In general, one can view each parity equation as being produced by composing the message bits, $X$, and a generator polynomial, $g$. In the first example above, the generator polynomial coefficients are $(1, 1, 1)$ and $(1, 1, 0)$, while in the second, they are $(1, 1, 1)$, $(1, 1, 0)$, and $(1, 0, 1)$.

We denote by $g_i$ the $K$-element generator polynomial for parity bit $p_i$. We can then write $p_i$ as follows:

$$p_i[n] = (\sum_{j=0}^{k-1} g_i[j]x[n-j]) \mod 2.$$  

(8.3)

The form of the above equation is a convolution of $g$ and $x$—hence the term “convolutional code”. The number of generator polynomials is equal to the number of generated parity bits, $r$, in each sliding window.

8.2.1 An Example

Let’s consider the two generator polynomials of Equations 8.1 (Figure 8-1). Here, the generator polynomials are

$$g_0 = 1, 1, 1$$
$$g_1 = 1, 1, 0$$

(8.4)

If the message sequence, $X = [1, 0, 1, 1, \ldots]$ (as usual, $x[n] = 0 \forall n < 0$), then the parity bits from Equations 8.1 work out to be

$$p_0[0] = (1 + 0 + 0) = 1$$
$$p_0[0] = (1 + 0) = 1$$
$$p_0[1] = (0 + 1 + 0) = 1$$
$$p_1[1] = (0 + 1) = 1$$
$$p_0[2] = (1 + 0 + 1) = 0$$
$$p_1[2] = (1 + 0) = 1$$
$$p_0[3] = (1 + 1 + 0) = 0$$
\[ p_1[3] = (1 + 1) = 0. \quad \text{(8.5)} \]

Therefore, the parity bits sent over the channel are \([1, 1, 1, 1, 0, 0, 0, 0, \ldots]\).

There are several generator polynomials, but understanding how to construct good ones is outside the scope of 6.02. Some examples (found by J. Busgang) are shown in Table 8-1.

<table>
<thead>
<tr>
<th>Constraint length</th>
<th>( G_1 )</th>
<th>( G_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>110</td>
<td>111</td>
</tr>
<tr>
<td>4</td>
<td>1101</td>
<td>1110</td>
</tr>
<tr>
<td>5</td>
<td>11010</td>
<td>11101</td>
</tr>
<tr>
<td>6</td>
<td>110101</td>
<td>111011</td>
</tr>
<tr>
<td>7</td>
<td>110101</td>
<td>110101</td>
</tr>
<tr>
<td>8</td>
<td>110111</td>
<td>1110011</td>
</tr>
<tr>
<td>9</td>
<td>110111</td>
<td>111001101</td>
</tr>
<tr>
<td>10</td>
<td>110111001</td>
<td>1110011001</td>
</tr>
</tbody>
</table>

Table: Examples of generator polynomials for rate 1/2 convolutional codes with different constraint lengths.

**Figure 8-2: Block diagram view of convolutional coding with shift registers.**

8.3 Two Views of the Convolutional Encoder

We now describe two views of the convolutional encoder, which we will find useful in better understanding convolutional codes and in implementing the encoding and decoding procedures. The first view is in terms of a **block diagram**, where one can construct the mechanism using shift registers that are connected together. The second is in terms of a **state machine**, which corresponds to a view of the encoder as a set of states with well-defined transitions between them. The state machine view will turn out to be extremely useful in figuring out how to decode a set of parity bits to reconstruct the original message bits.
Block Diagram View

Figure 8-2 shows the same encoder as Figure 8-1 and Equations (8.1) in the form of a block diagram. The $x[n-i]$ values (here there are two) are referred to as the state of the encoder. The way to think of this block diagram is as a “black box” that takes message bits in and spits out parity bits.

Input message bits, $x[n]$, arrive on the wire from the left. The box calculates the parity bits using the incoming bits and the state of the encoder (the $k-1$ previous bits; 2 in this example). After the $r$ parity bits are produced, the state of the encoder shifts by 1, with $x[n]$ taking the place of $x[n-1]$, $x[n-1]$ taking the place of $x[n-2]$, and so on, with $x[n-K+1]$ being discarded. This block diagram is directly amenable to a hardware implementation using shift registers.

State Machine View

Another useful view of convolutional codes is as a state machine, which is shown in Figure 8-3 for the same example that we have used throughout this lecture (Figure 8-1).

The state machine for a convolutional code is identical for all codes with a given constraint length, $K$, and the number of states is always $2^{K-1}$. Only the $p_i$ labels change depending on the number of generator polynomials and the values of their coefficients. Each
state is labeled with $x[n-1]x[n-2]\ldots x[n-K+1]$. Each arc is labeled with $x[n]/p_0p_1p_2\ldots p_{K-1}$.

In this example, if the message is 101100, the transmitted bits are 11 11 01 00 01 10.

This state machine view is an elegant way to explain what the transmitter does, and also what the receiver ought to do to decode the message, as we now explain. The transmitter begins in the initial state (labeled “STARTING STATE” in Figure 8-3) and processes the message one bit at a time. For each message bit, it makes the state transition from the current state to the new one depending on the value of the input bit, and sends the parity bits that are on the corresponding arc.

The receiver, of course, does not have direct knowledge of the transmitter’s state transitions. It only sees the received sequence of parity bits, with possible corruptions. Its task is to determine the best possible sequence of transmitter states that could have produced the parity bit sequence. This task is called decoding, which we will introduce next, and then study in more detail in the next lecture.

### 8.4 The Decoding Problem

As mentioned above, the receiver should determine the “best possible” sequence of transmitter states. There are many ways of defining “best”, but one that is especially appealing is the most likely sequence of states (i.e., message bits) that must have been traversed (sent) by the transmitter. A decoder that is able to infer the most likely sequence is also called a maximum likelihood decoder.

Consider the binary symmetric channel, where bits are received erroneously with probability $p < 1/2$. What should a maximum likelihood decoder do when it receives r? We show now that if it decodes r as c, the nearest valid codeword with smallest Hamming distance from r, then the decoding is a maximum likelihood one.

A maximum likelihood decoder maximizes the quantity $P(r|c)$; i.e., it finds c so that the probability that r was received given that c was sent is maximized. Consider any codeword $c'$. If r and $c'$ differ in d bits (i.e., their Hamming distance is d), then $P(r|c) = p^d(1-p)^{N-d}$, where N is the length of the received word (and also the length of each valid codeword). It’s more convenient to take the logarithm of this conditional probability, also termed the log-likelihood:

$$
\log P(r|c') = d \log p + (N - d) \log(1-p) = d \log \frac{p}{1-p} + N \log(1-p). \quad (8.6)
$$

If $p < 1/2$, which is the practical realm of operation, then $\frac{p}{1-p} < 1$ and the log term is negative (otherwise, it’s non-negative). As a result, minimizing the log likelihood boils down to minimizing d, because the second term on the RHS of Eq. (8.6) is a constant.

A simple numerical example may be useful. Suppose that bit errors are independent and identically distributed with a BER of 0.001, and that the receiver digitizes a sequence of analog samples into the bits 1101001. Is the sender more likely to have sent 1100111 or 1100001? The first has a Hamming distance of 3, and the probability of receiving that sequence is $(0.999)^4(0.001)^3 = 9.9 \times 10^{-8}$. The second choice has a Hamming distance of 1 and a probability of $(0.999)^6(0.001)^1 = 9.9 \times 10^{-1}$, which is six orders of magnitude higher and is overwhelmingly more likely.

Thus, the most likely sequence of parity bits that was transmitted must be the one with the smallest Hamming distance from the sequence of parity bits received. Given a choice
of possible transmitted messages, the decoder should pick the one with the smallest such Hamming distance.

Determining the nearest valid codeword to a received word is easier said than done for convolutional codes. For example, see Figure 8-4, which shows a convolutional code with \( k = 3 \) and rate 1/2. If the receiver gets 111011000110, then some errors have occurred, because no valid transmitted sequence matches the received one. The last column in the example shows \( d \), the Hamming distance to all the possible transmitted sequences, with the smallest one circled. To determine the most-likely 4-bit message that led to the parity sequence received, the receiver could look for the message whose transmitted parity bits have smallest Hamming distance from the received bits. (If there are ties for the smallest, we can break them arbitrarily, because all these possibilities have the same resulting post-

The base of the logarithm doesn’t matter to us at this stage, but traditionally the log likelihood is defined as the natural logarithm (base \( e \)).

<table>
<thead>
<tr>
<th>( \text{Msg} )</th>
<th>( \text{Xmit}^* )</th>
<th>( \text{Rcvd} )</th>
<th>( d )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>000000000000</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>0001</td>
<td>000000111110</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>0010</td>
<td>000011111000</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>0011</td>
<td>000011010111</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>0100</td>
<td>001111100000</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>0101</td>
<td>001111011110</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>0110</td>
<td>001101001000</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>0111</td>
<td>001100100110</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>111110000000</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>1001</td>
<td>111110111110</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>1010</td>
<td>111101111000</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>1011</td>
<td>111101000110</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>1100</td>
<td>110001100000</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>1101</td>
<td>110001011110</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>1110</td>
<td>110010011000</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>1111</td>
<td>110010100110</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 8-4:** When the probability of bit error is less than 1/2, maximum likelihood decoding boils down to finding the message whose parity bit sequence, when transmitted, has the smallest Hamming distance to the received sequence. Ties may
be broken arbitrarily. Unfortunately, for an $N$-bit transmit sequence, there are $2^N$ possibilities, which makes it hugely intractable to simply go through in sequence because of the sheer number. For instance, when $N = 256$ bits (a really small packet), the number of possibilities rivals the number of atoms in the universe!

coded BER.)

The straightforward approach of simply going through the list of possible transmit sequences and comparing Hamming distances is horribly intractable. The reason is that a transmit sequence of $N$ bits has $2^N$ possible strings, a number that is simply too large for even small values of $N$, like 256 bits. We need a better plan for the receiver to navigate this unbelievable large space of possibilities and quickly determine the valid message with smallest Hamming distance. We will study a powerful and widely applicable method for solving this problem, called Viterbi decoding, in the next lecture. This decoding method uses a special structure called the trellis, which we describe next.

8.5 The Trellis and Decoding the Message

The trellis is a structure derived from the state machine that will allow us to develop an efficient way to decode convolutional codes. The state machine view shows what happens

![Trellis Diagram](image)

Figure 8-5: The trellis is a convenient way of viewing the decoding task and understanding the time evolution of the state machine.

at each instant when the sender has a message bit to process, but doesn’t show how the system evolves in time. The trellis is a structure that makes the time evolution explicit. An example is shown in Figure 8-5. Each column of the trellis has the set of states; each state in a column is connected to two states in the next column—the same two states in the state diagram. The top link from each state in a column of the trellis shows what gets
transmitted on a “0”, while the bottom shows what gets transmitted on a “1”. The picture shows the links between states that are traversed in the trellis given the message 101100.

We can now think about what the decoder needs to do in terms of this trellis. It gets a sequence of parity bits, and needs to determine the best path through the trellis—that is, the sequence of states in the trellis that can explain the observed, and possibly corrupted, sequence of received parity bits.

The Viterbi decoder finds a maximum likelihood path through the Trellis. We will study it in the next lecture.