วศ 5012223

การสื่อสารดิจิทัล

EN 5012223 Digital Communication

วิศวกรรมการสื่อสารและสารสนเทศ

คณะเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยราชภัฏเทพสตรี

วศ 5012223 **การสื่อสารดิจิทัล** **3(3-0-6)**

EN 5012223 **Digital Communication**

วิชาที่ต้องศึกษาก่อน : ไม่มี

หลักการระบบสื่อสารแบบดิจิทัล เทคนิคการมอดูเลตสัญญาณดิจิทัล การวิเคราะห์ประสิทธิภาพ การซิงโครไนซ์ กล่าวนำเกี่ยวกับทฤษฎีของข้อมูล การเข้ารหัสต้นทาง การเข้ารหัสช่องสัญญาณ ระบบหลายช่องทางและหลายพาท์ เทคนิคการการแผ่สเปกตรัม การจางหายของสัญญาณจากการแพร่หลายเส้นทาง วิศวกรรมทราฟิกและคิวโอเอส การสื่อสารดิจิทัลบรอดแบนด์

ระหว่างภาค 60 คะแนน

สอบเก็บคะแนนครั้งที่ 1 (บทที่ 1) [1]	10 คะแนน
สอบกลางภาค (บทที่ 2, 3, 4) [2]	30 คะแนน
สอบเก็บคะแนนครั้งที่ 2 (บทที่ 5) [3]	10 คะแนน
งานที่ได้รับมอบหมาย	10 คะแนน

ปลายภาค 40 คะแนน

ต่ำกว่า 50 คะแนน -> ไม่ผ่าน (F)

[1] สอบสัปดาห์ที่ 2

[2] สอบสัปดาห์ที่ 8

[3] สอบสัปดาห์ที่ 11

Contents

1.Communications Fundamentals (1)

2.A Review of Some Important Math (2-3)

3.Source Coding and Decoding (4-5)

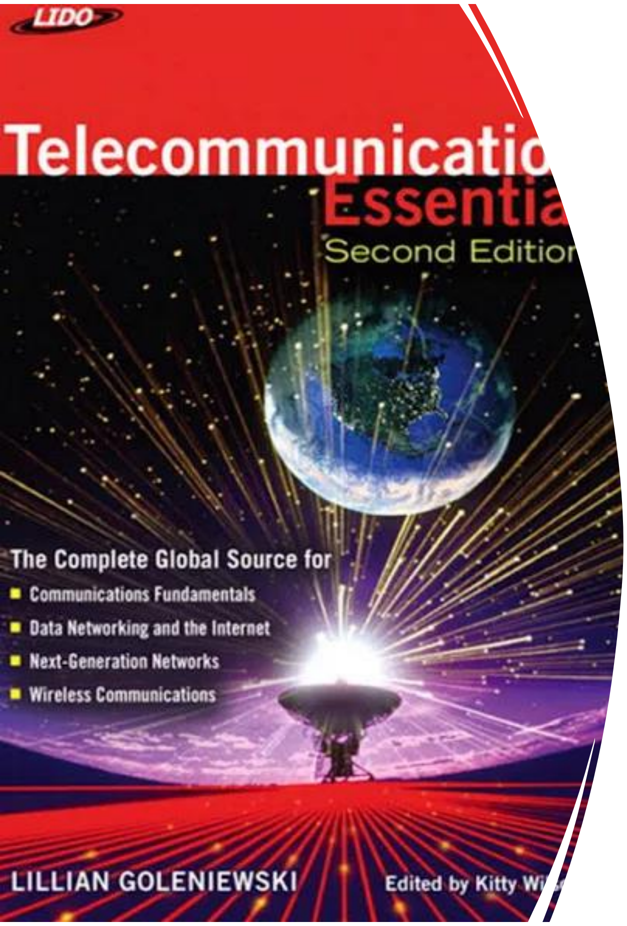
4.Modulators and Demodulators (6-7)

5.Block Coding and Decoding (9-10)

6.Convolutional Coding and Decoding (11-12)

7.Quality of service (13)

8.Broadband Networks (14-15)



1. Communications Fundamentals

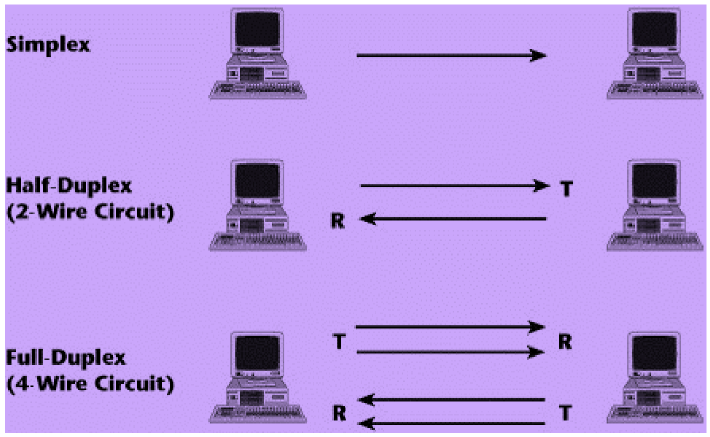
- 1.1 Transmission Lines
- 1.2 Types of Network Connections
- 1.3 The Electromagnetic Spectrum
- 1.4 Analog and Digital Transmission
- 1.5 Multiplexing
- 1.6 Traditional Transmission Media
- 1.7 Establishing Channels
- 1.8 Political and Regulatory Forces

1.1 Transmission Lines

- Two prerequisites must be satisfied to have successful communication.
- The first prerequisite is understandability. The transmitter and receiver must speak the same language.
- The second prerequisite is the capability to detect errors as they occur and to have some procedure for resolving those errors.
- If these two prerequisites understandability and error control are met, then communication can occur. We communicate by using data devices over what is generically termed a *transmission line*. There are five main types of transmission lines: *circuits, channels, lines, trunks,* and *virtual circuit*.

1.1 Transmission Lines

- A *circuit* is the physical path that runs between two or more points. It terminates on a *port* , and that port can be in a host computer, on a multiplexer, on a switch, or in another device.

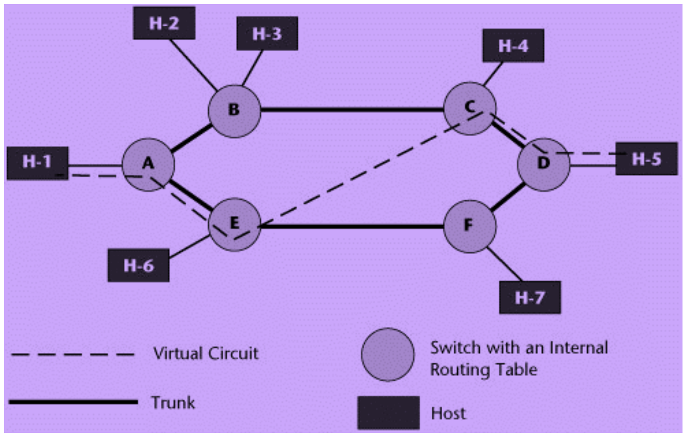


1.1 Transmission Lines

- A *channel* defines a logical conversation path. It is the frequency band, time slot, or wavelength (also referred to as lambda, λ) allocated to a single conversation. In the context of telecommunications, a channel is a child of the digital age because digital facilities enable multiple channels, greatly increasing the carrying capacity of an individual circuit. Because we are becoming more digitalized all the time, people often refer to the number of channels rather than the number of circuits.
- *Lines* and *trunks* are basically the same thing, but they're used in different situations. A *line* is a connection configured to support a normal calling load generated by one individual. A *trunk* is a circuit configured to support the calling loads generated by a group of users; it is the transmission facility that ties together switching systems. A switching system is a device that connects two transmission lines.

1.1 Transmission Lines

- Because of the great interest in and increased use of packet switching, most networks use *virtual circuits*. Unlike a *physical circuit*, which terminates on specific physical ports, a virtual circuit is a series of logical connections between sending and receiving devices.



1.2 Types of Network Connections

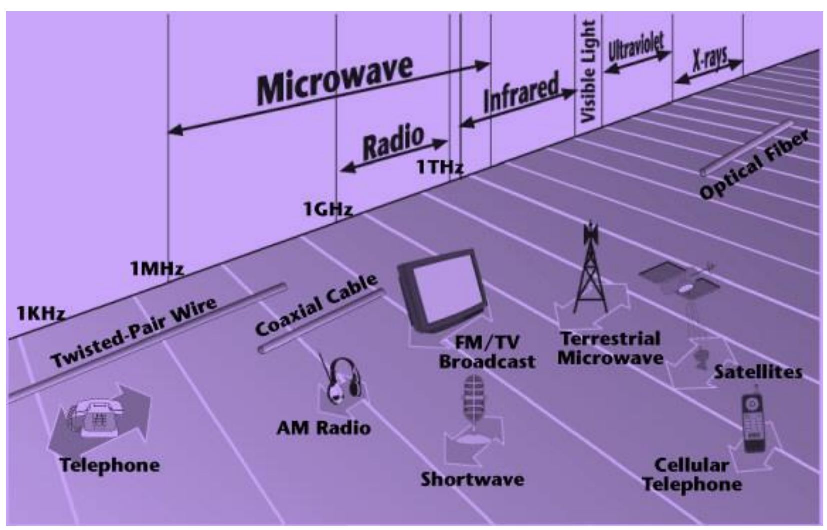
- *Switched network connections* : A switched connection is referred to as a dialup connection. This implies that it uses a series of network switches to establish the connection between the parties.
- *Leased-line network connections* : A leased line is also referred to as a private line. With a leased line, the same locations or the same devices are always connected, and transmission between those locations or devices always occurs on the same path.
- *Dedicated network connections* : In essence, a dedicated line works exactly like a leased line. It is always connected, and it always uses the same path for transmission. However, the end user may own the transmission facility (rather than lease it) such that it is exclusive to that user.

1.3 The Electromagnetic Spectrum

- When electrons move, they create electromagnetic waves that can propagate through free space. *James Maxwell* first predicted the existence of this phenomenon in 1865, and *Heinrich Hertz* first produced and observed it in 1887. All modern communication depends on manipulating and controlling signals within the electromagnetic spectrum.
- The electromagnetic spectrum ranges from extremely low-frequency radio waves of 30Hz, with wavelengths nearly double the earth's diameter, to high-frequency cosmic rays of more than 10 million trillion Hz, with wavelengths smaller than the nucleus of an atom.
- The electromagnetic spectrum is depicted as a logarithmic progression: The scale increases by multiples of 10, so the higher regions encompass a greater span of frequencies than do the lower regions.

1.3 The Electromagnetic Spectrum

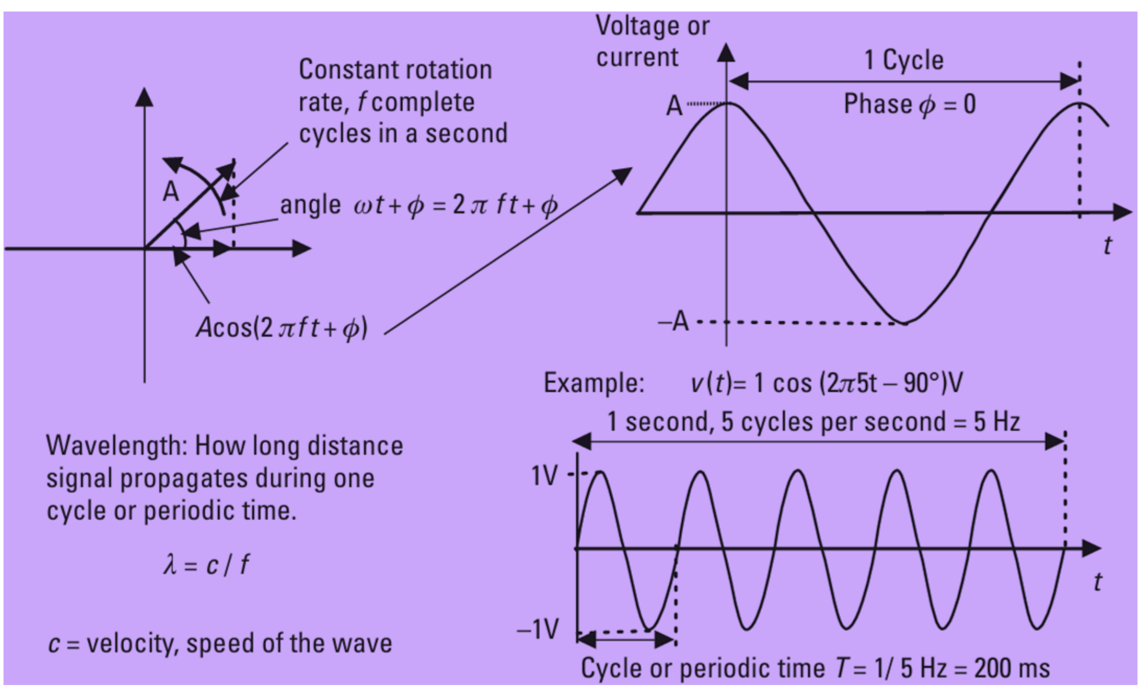
Transmission Media	ITU-Defined Band	Frequency
	Gamma Rays	10^{22}
		10^{19}
	X-rays	10^{16}
	Ultraviolet	
Fiber Optics	Visible Light	10^{14}
	Infrared	
Microwave	THF	10^{12} **1THz**
	EHF	
PCS Microwave	SHF	
FM TV, Cellular Radio	UHF	10^{9} **1GHz**
TV, Coax	VHF	
AM		
Coax	High	10^{6} **1MHz**
Twisted Pair	Medium	
Audio Frequencies	Low	
	Very Low	10^{3} **1KHz**
	Extremely Low	
		10^{0} **1Hz**



1.3 The Electromagnetic Spectrum

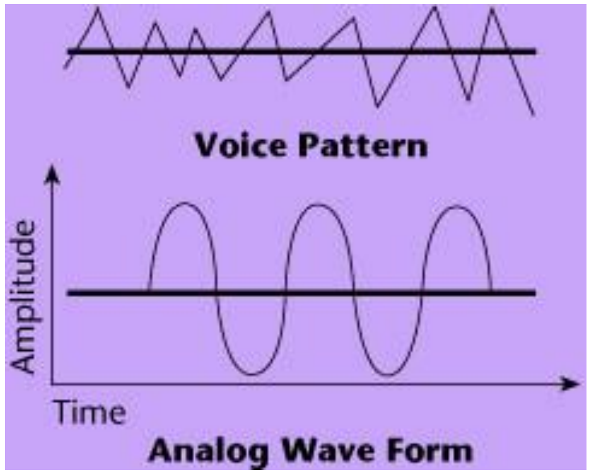
- *Frequency* : The number of oscillations per second of an electromagnetic wave.
- *Hertz* : Frequency is measured in Hertz (Hz).
- *Wavelength* : The distance between two consecutive maxima or minima of the waveform.
- *Amplitude* : The height of the wave, which indicates the strength, or power, of the signal.
- *Phase* : The phase of a wave refers to the angle of the waveform at any given moment; more specifically, it defines the offset of the wave from a reference point.
- Bandwidth : The range of frequencies (i.e., the difference between the lowest and highest frequencies carried).

1.3 The Electromagnetic Spectrum



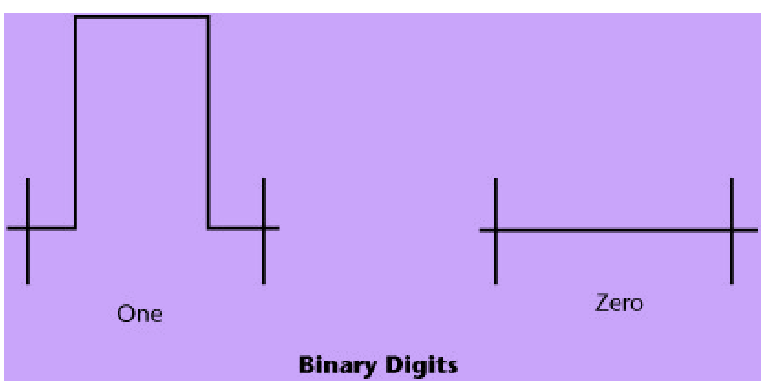
1.4 Analog and Digital Transmission

- *Analog Transmission* : An analog waveform (or signal) is characterized by being continuously variable along amplitude and frequency.

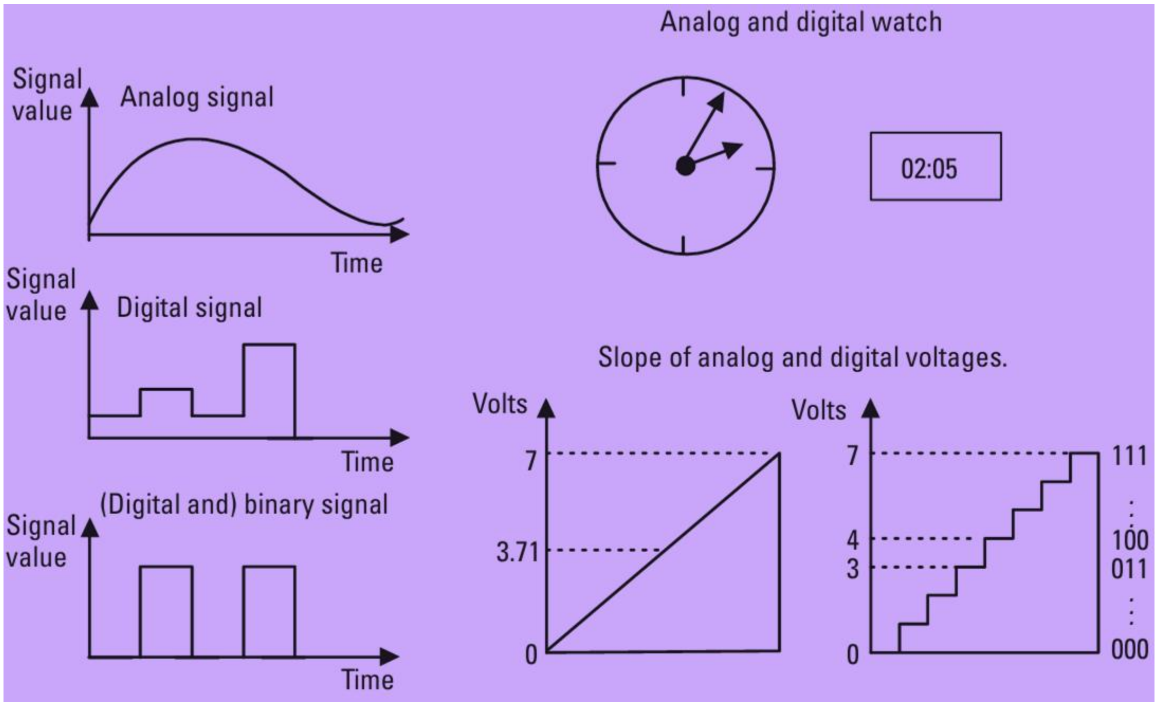


1.4 Analog and Digital Transmission

- *Digital Transmission* : It is a series of discrete pulses, representing one bits and zero bits. In electrical networks, one bits are represented as high voltage, and zero bits are represented as null, or low voltage.

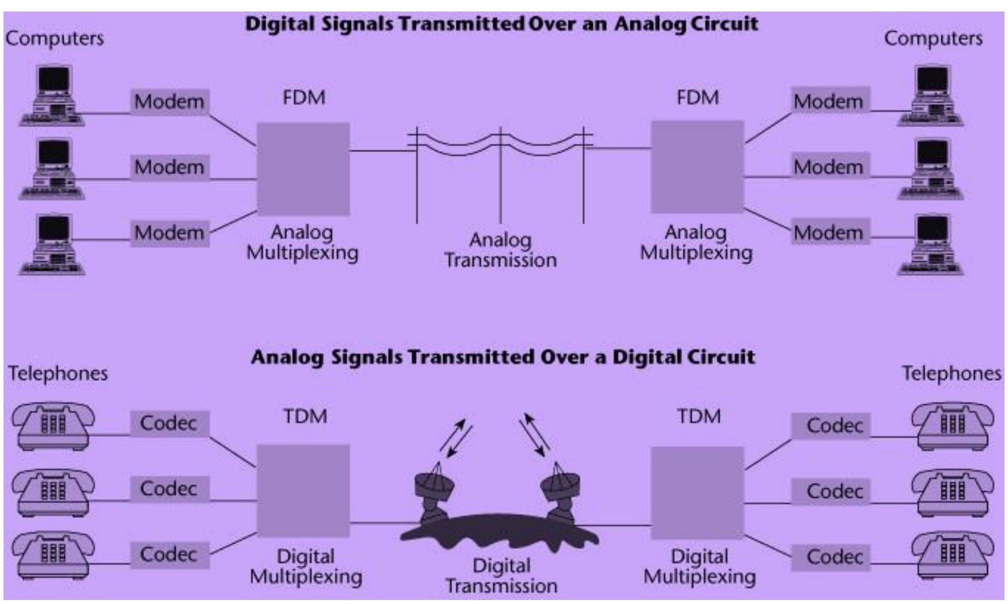


1.4 Analog and Digital Transmission



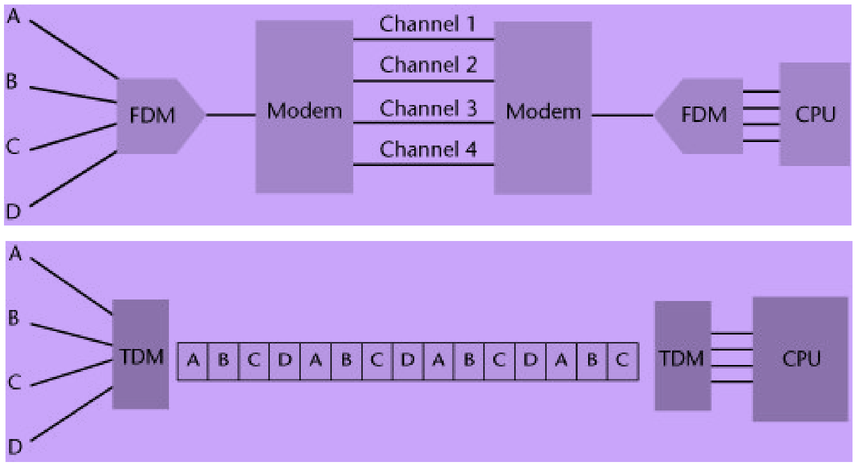
1.4 Analog and Digital Transmission

- Conversion: *Codecs* and *Modems*



1.5 Multiplexing

- *Multiplexers*, often called muxes, are extremely important to communication system. Their main reason for being is to reduce network costs by minimizing the number of communications links needed between two points.



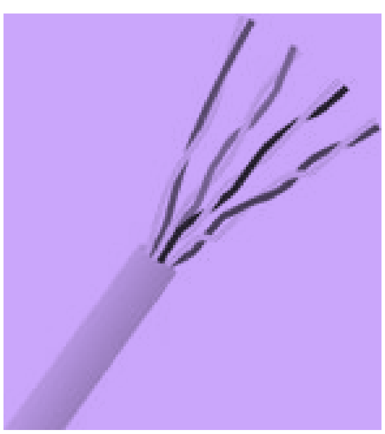
1.6 Traditional Transmission Media

- *Twisted-Pair* - There are two types of twisted-pair: UTP and STP. In STP, a metallic shield around the wire pairs minimizes the impact of outside interference. Most implementations today use UTP.
- The primary applications of twisted-pair are in premises distribution systems, telephony, private branch exchanges (PBXs) between telephone sets and switching cabinets, LANs, and local loops, including both analog telephone lines and broadband DSL.
- Twisted-pair is used in traditional analog subscriber lines, also known as the telephony channel or 4KHz channel. Digital twisted-pair takes the form of Integrated Services Digital Network (ISDN) and the new-generation family of DSL standards, collectively referred to as xDSL

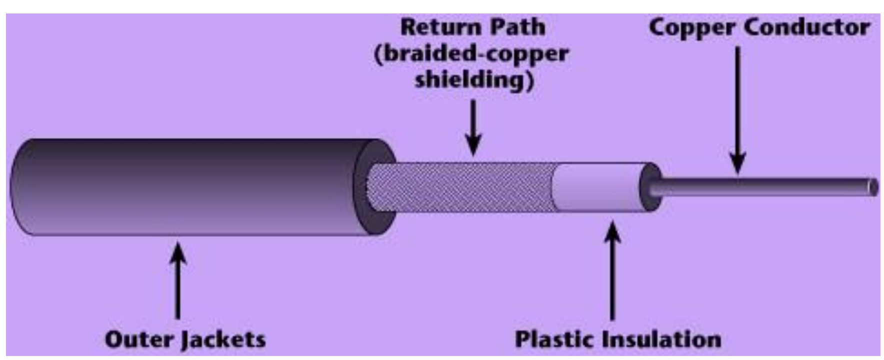
1.6 Traditional Transmission Media

- *Coaxial Cable* - In the mid-1920s, coax was applied to telephony networks as interoffice trunks.
- The next major use of coax in telecommunications occurred in the 1950s, when it was deployed as submarine cable to carry international traffic.
- It was then introduced into the data-processing realm in the mid- to late 1960s. Early computer architectures required coax as the media type from the terminal to the host.
- LANs were predominantly based on coax from 1980 to about 1987.
- Coax has been used in cable TV and in the local loop, in the form of HFC architectures.

1.6 Traditional Transmission Media



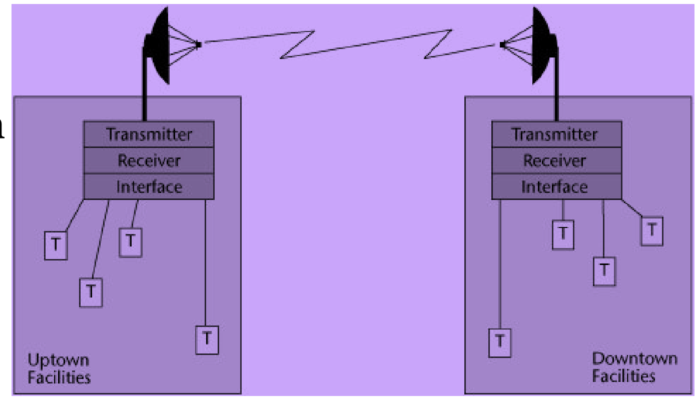
Twisted-Pair



Coaxial Cable

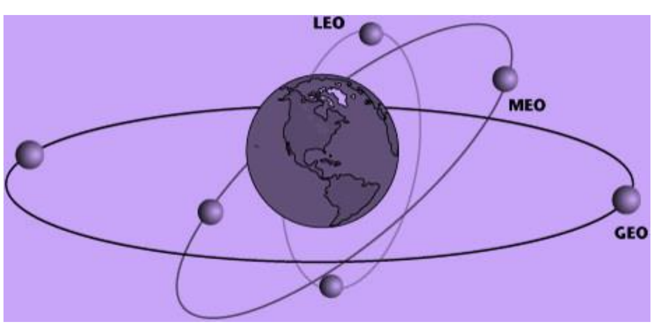
1.6 Traditional Transmission Media

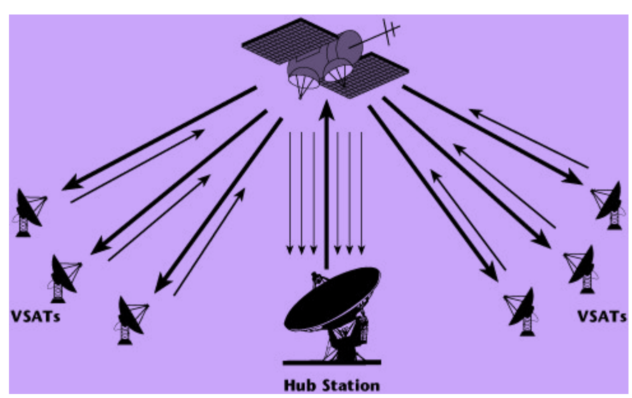
- twisted-pair and coax both face limitations because of the frequency spectrum.
- *microwave* promises to have a much brighter future than twisted-pair or coax.
- Many locations cannot be cost-effectively cabled by using wires and this is where microwave can shine.
- In addition, the microwave spectrum is the workhorse of the wireless world.



1.6 Traditional Transmission Media

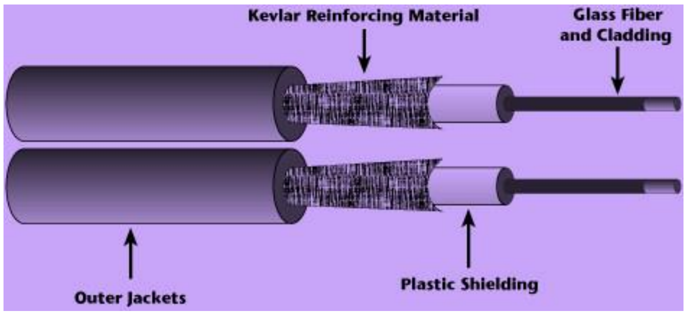
- *Satellite* - In descriptions of satellite services, three abbreviations relate to the applications that are supported: FSS Fixed satellite services, BSS Broadcast satellite services and MSS Mobile satellite services

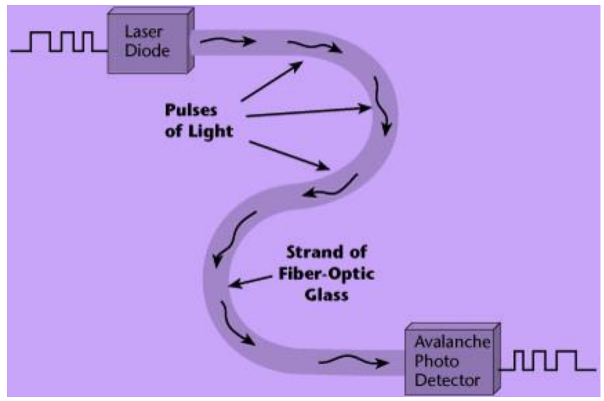




1.6 Traditional Transmission Media

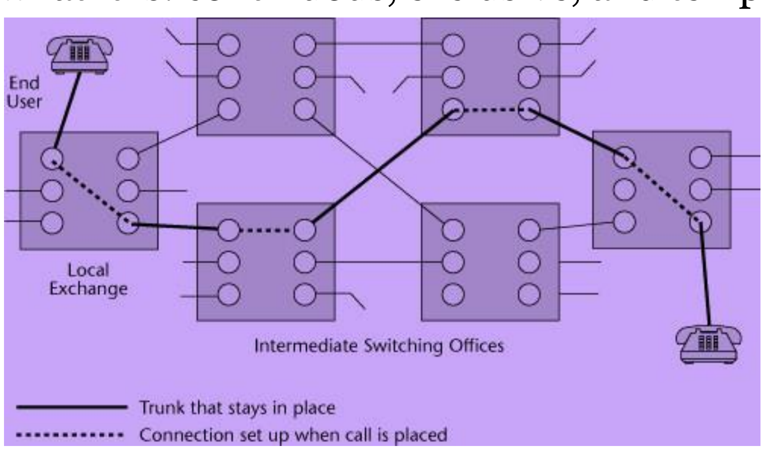
- *Fiber Optics* - Fiber optics operates in the visible light spectrum. Wavelength is a measure of the width of the waves being transmitted. Different fiber-optic materials are optimized for different wavelengths. The EIA/TIA standards currently support three wavelengths for fiber-optic transmission: 850, 1,300, and 1,550 nanometers (nm).





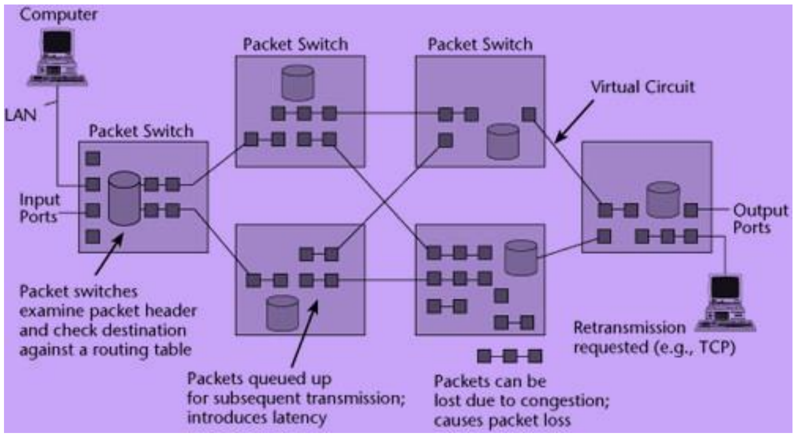
1.7 Establishing Channels

- There are two switching modes: circuit switching and packet switching.
- *Circuit switching* has been the basis of voice networks worldwide for many years. You can apply three terms to the nature of a circuit-switched call to help remember what it is: continuous, exclusive, and temporary.



1.7 Establishing Channels

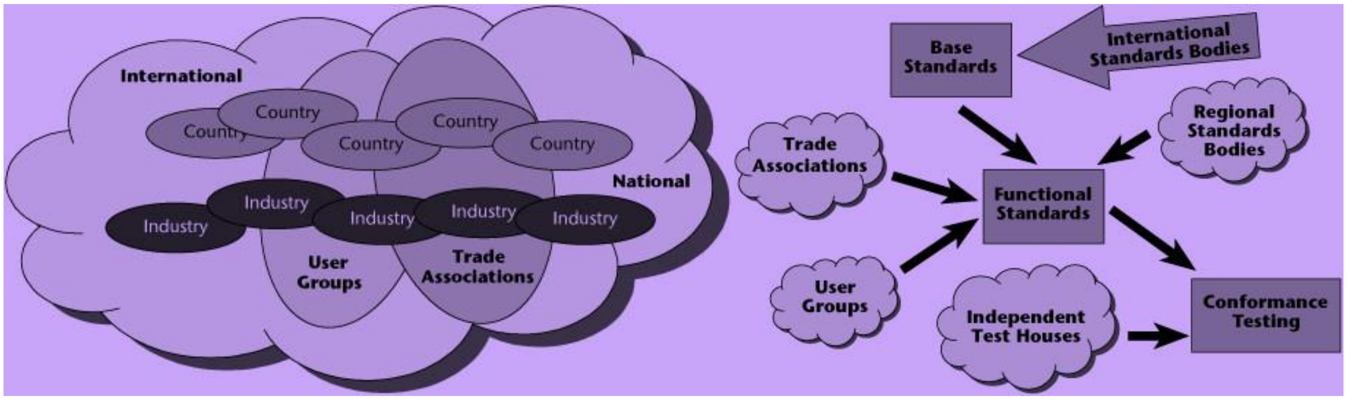
- *Packet switching* has its origin in data communications. In fact, packet switching was developed specifically as a solution for the communications implications of a form of data processing called interactive processing.



1.7 Establishing Channels

Characteristic	Circuit Switching	Packet Switching
Origin	Voice telephony	Data networking
Connectionless or connection oriented	Connection oriented	Both
Key applications	Real-time voice, streaming media, videoconferencing, video-on-demand, and other delay- and loss-sensitive traffic applications	Bursty data traffic that has long connect times but low data volumes; applications that are delay and loss tolerant
Latency/delay/jitter	Low latency and minimal delays	Subject to latency, delay, and jitter because of its store-and-forward nature
Network intelligence	Centralized	Decentralized
Bandwidth efficiency	Low	High
Packet loss	Low	Low to high, depending on the network

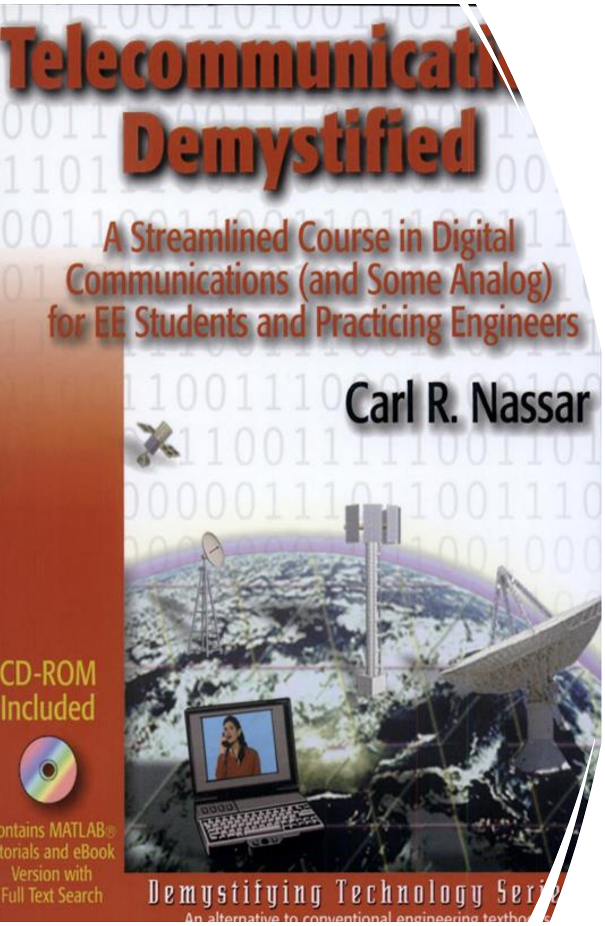
1.8 Political and Regulatory Forces



Standards-making groups & The standards-making process

1.8 Political and Regulatory Forces

Region	Standards Organizations
International	ITU, ITU-T, ITU-R, ITU-D, IEC, ISO
Australia	ACA, ACC, AIIA, ATUG
Europe	AFNOR (France), CEN and CENLEC, CEPT, DIN (Germany), DTI (UK), ETSI, European Community (EU)
Japan	JISC, TTC
New Zealand	ITANZ
North America	ANSI (USA), EIA, FCC (USA), IEEE, NIST, SCC (Canada)



2. A Review of Some Important Math

2.1 Random Variables

- A *random event*, *A*, refers simply to an event with an unknown outcome.
- An example of a random event is tomorrow's weather.
- A *random variable*, x, is a number whose value is determined by a random event, A.
- For example, it may be tomorrow's outdoor temperature.

2.1 Random Variables

- One way to fully characterize our random variable x is by a function called the *probability distribution function*, $F_x(X)$.
- The function $F_x(X)$ is defined in words as follows: $F_x(X)$ is the likelihood that the random variable x is less than the number X.
- In a nice, neat equation, $F_x(X)$ is defined as

$$F_x(X) = P(x \leq X)$$

- where P(___) is shorthand for the words "probability that ___ happens."
- The probability distribution function is also known as the *cumulative distribution function (CDF)*.

2.1 Random Variables

- Example 1: Suppose there are 6 balls in a bag. The random variable X is the weight of a ball (in kg) selected at random. Balls 1, 2, and 3 weighs 0.5 kg; Balls 4 and 5 weighs 0.25 kg; and Ball 6 weighs 0.3 kg. Write the Probability for X.

weight of a ball (*X*)	Probability *P(x)*
0.25	2/6
0.30	
0.50	

2.1 Random Variables

- Example 2: Suppose we toss two dice. Make a table of the probabilities for the sum of the dice.

X	1	2	3	4	5	6	7	8	9	10	11	12	13
P(x)													
$F_x(X)$													

2.1 Random Variables

- Here are four simple properties of $F_x(X)$:

(1) $0 < F_x(X) < 1$: that is, since $F_x(X)$ represents the probability that $x < X$, it, like all probabilities, must be between 0 (never happens) and 1 (always happens).

(2) $F_x(-\infty) = 0$: that is, $F_x(-\infty) = \mathrm{P}(x < -\infty) =$ (the probability that x is less than $-\infty$) = 0 (since no number can be smaller than $-\infty$).

(3) $F_x(\infty) = 1$: that is, $F_x(\infty) = \mathrm{P}(x < \infty) =$ (the probability that x is less than ∞) = 1 (since every value must be smaller than ∞).

(4) $F_x(\mathrm{x}_1) \geq F_x(\mathrm{x}_2)$ if $\mathrm{x}_1 > \mathrm{x}_2$: that is, for example, the probability that x is less than 20 is at least as big as the probability that x is less than 10 .

2.1 Random Variables

- Example 3: The number of old people living in houses on a randomly selected city block is described by the following probability distribution.

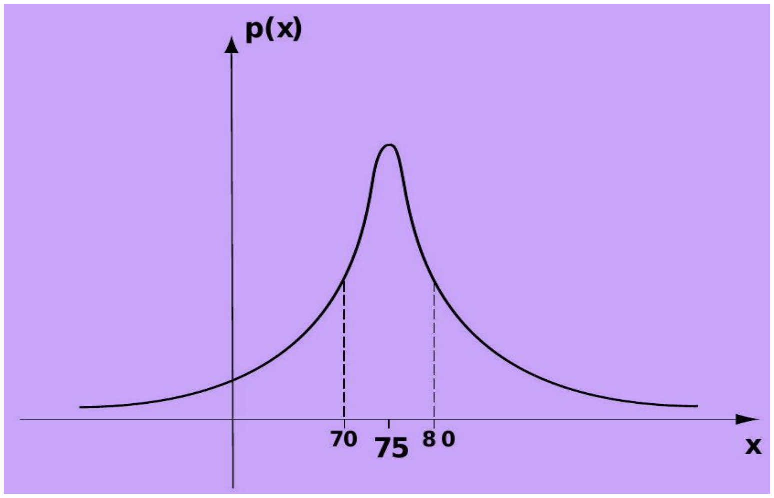
Number of adults (X)	Probability $P(x)$	$F_x(X)$
3	0.50	
4	0.25	
5	0.10	
6		

2.1 Random Variables

- A second way to describe our random variable x is to use a different function called the *probability density function* (*pdf* for short).
- The *pdf* for this variable is denoted $p_x(x)$ or $p(x)$

$$P(x_1 \le x \le x_2) = \int_{x_1}^{x_2} p(x)dx$$

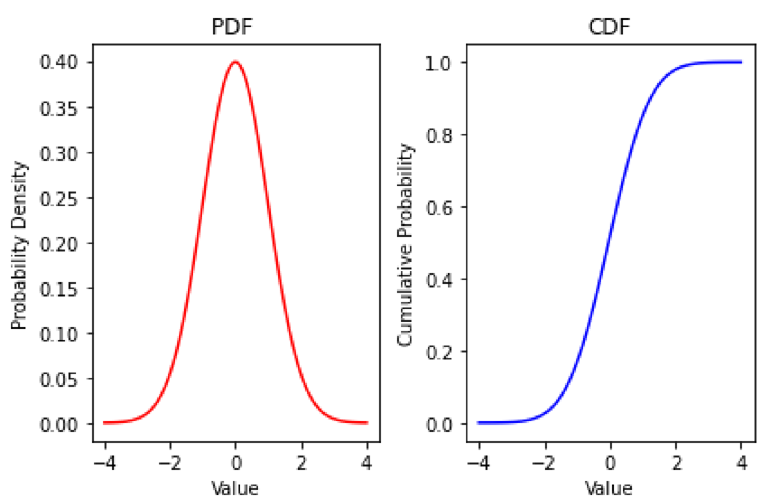
2.1 Random Variables



- if you want to know how likely it is that tomorrow's temperature x will be between 70 degrees and 80 degrees, all you have to do is integrate $p(x)$ over the range of 70 to 80 .
- *p(x)* at *x*=70 gives you an idea how likely it is that tomorrow's temperature will be about 70 degrees.

2.1 Random Variables

- CDF is the integral of the PDF, and the PDF is the derivative of the CDF.



2.1 Random Variables

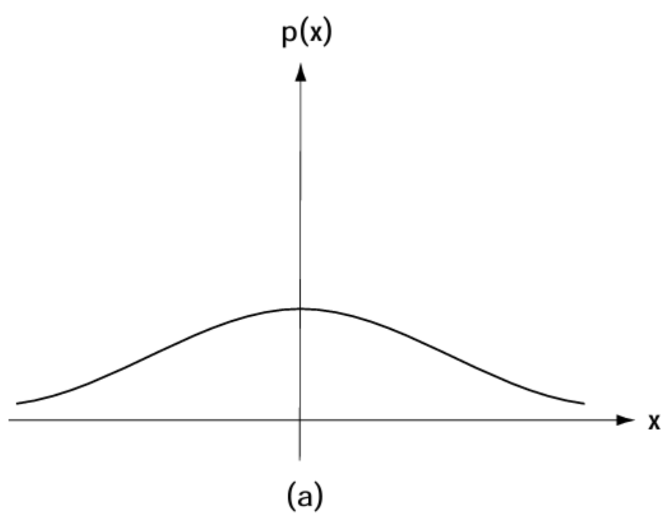
- The mean, x_m (also known as E(x)) : One thing you can tell me is the average (or mean) value of x

$$x_m = \int_{-\infty}^{\infty} xp(x)dx$$

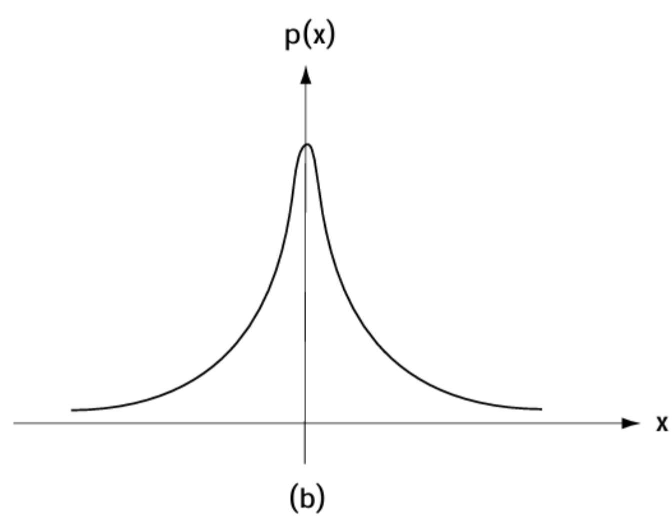
- The variance, σ_n^2 : Another important piece of information about the random variable x is how much x varies.

$$\sigma_n^2 = \int_{-\infty}^{\infty} (x - x_m)^2 p(x)dx$$

2.1 Random Variables



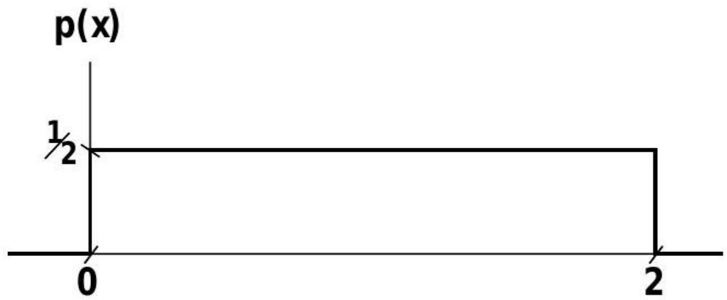
Random variable x with large variance



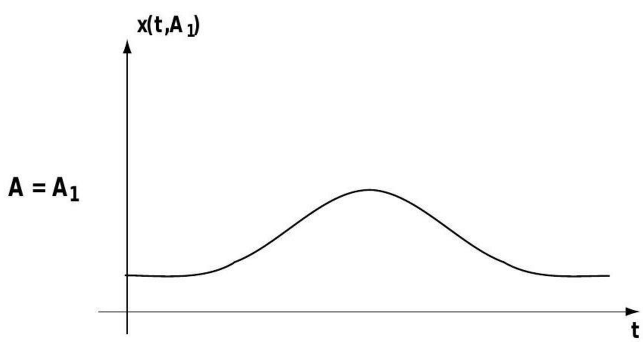
Random variable x with small variance

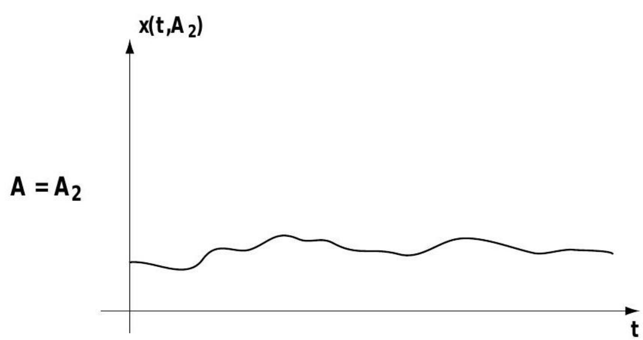
2.1 Random Variables

- Example 4: Given a random variable x and told that x has a probability distribution function $p(x)$, determine its mean and its variance.



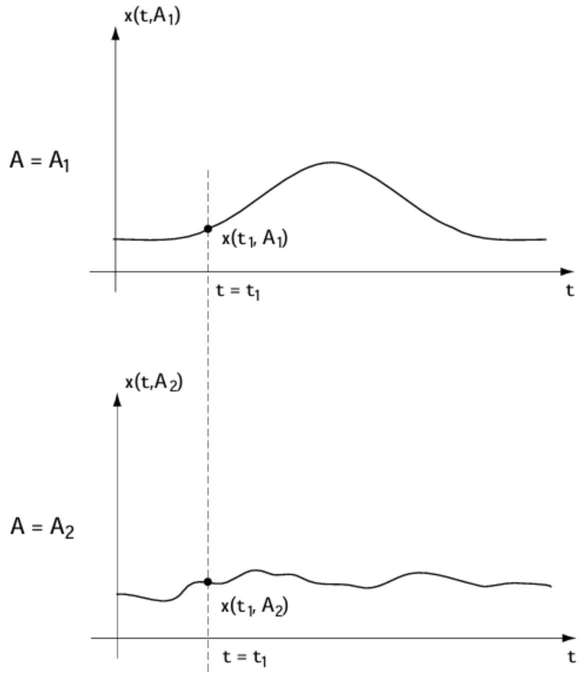
2.2 Random Processes





- A random process, $x(t)$, is a function of time t, where the exact function that occurs depends on a random event A.
- For example, let $x(t)$ be tomorrow's temperature as it changes with time over the day; the values of $x(t)$ will depend on A (whether it's sunny or not).
- So, we can write $x(t) = x(t, A)$ to indicate that the time function depends on the random event A. Here, $x(t, A)$ is a random process.

2.2 Random Processes



- There's one very important thing to note about a random process $x(t, A)$.
- At time $t = t_1$, we have $x(t_1, A)$, which is a number whose exact value depends on A.
- That's just a random variable! So, the sample of a $A = A_1$ random process $x(t, A)$ at $t = t_1$ is a random variable.
- We'll call it $x_1 = x_1(A)$.

2.2 Random Processes

- the mean of $x(t_1, A) = x_1$: this number (which may be different at different times t_1) tells you the average value of $x(t, A)$ at $t = t_1$. This value can be generated using the equation

$$x_m(t_1) = \int_{-\infty}^{\infty} x_1 p(x_1) dx_1$$

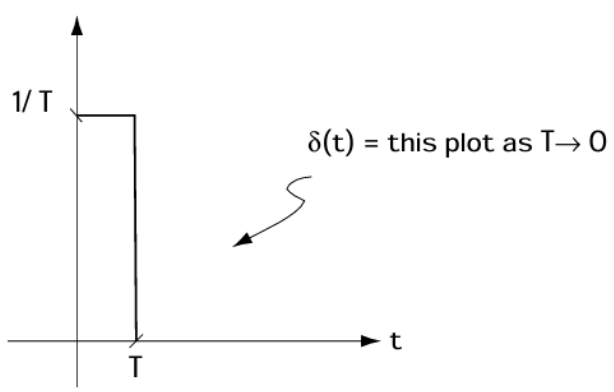
2.2 Random Processes

- the autocovariance, $R_x(t_1, t_2)$: this number (which may be different for different t_1 and t_2 values) describes the relationship between the random variable $x(t_1, A) = x_1$ and the random variable $x(t_2, A) = x_2$.

$$R_x(t_1, t_2) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left(x_1 - x_m(t_1)\right)\left(x_2 - x_m(t_2)\right)p(x_1, x_2)dx_1dx_2$$

- The larger this number, the more closely related $x(t_1, A)$ is to $x(t_2, A)$. This value can be generated mathematically through the equation

2.3 Signals and Systems



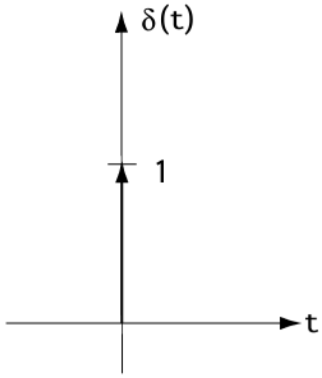
- $\delta(t)$, which is called the impulse function (or delta function, or just impulse)
- $\delta(t)$ is infinitely tall;
- $\delta(t)$ is infinitely skinny; and
- the area under the $\delta(t)$ function is 1

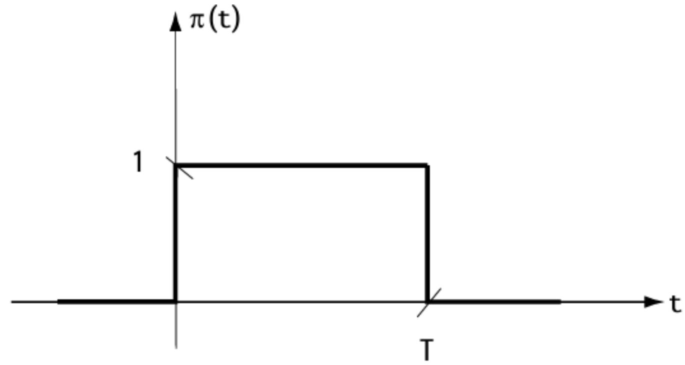
$$\int_{-\infty}^{\infty}\delta(t)dt = 1$$

2.3 Signals and Systems

- A square wave function, $\pi(t)$.
- This function is of height 1 and duration T.

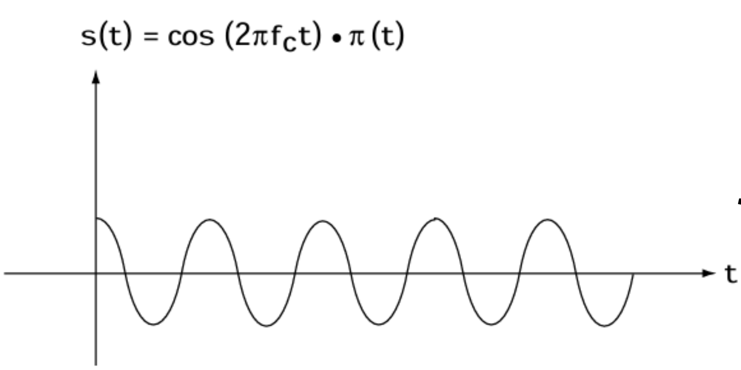
$$\pi(t)=\begin{cases}1, & 0 \le t \le \mathrm{T}\\ 0, & \text{else}\end{cases}$$





2.3 Signals and Systems

- To describe signals, discovered by a fellow named Fourier.
- Any time signal $s(t)$ can be created by adding together cosine and sine waveforms with different frequencies.
- You can describe $s(t)$ by indicating how much of each frequency f you need to put together to make your signal $s(t)$.

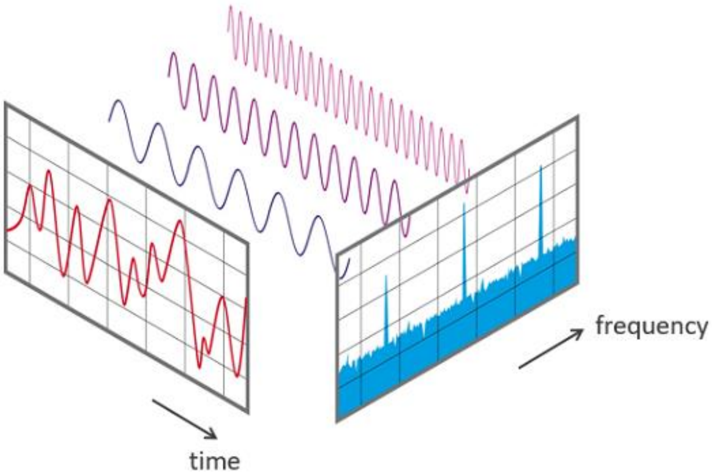


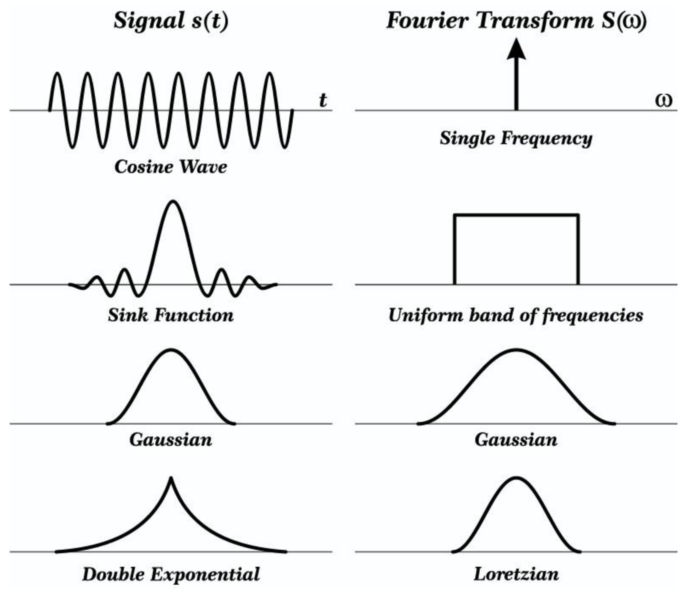
$$s(t) = \cos(2\pi f_c t) \cdot \pi(t)$$

$$S(f) = F\{s(t)\} = \int_{-\infty}^{\infty} s(t)e^{-j2\pi t}dt$$

2.3 Signals and Systems

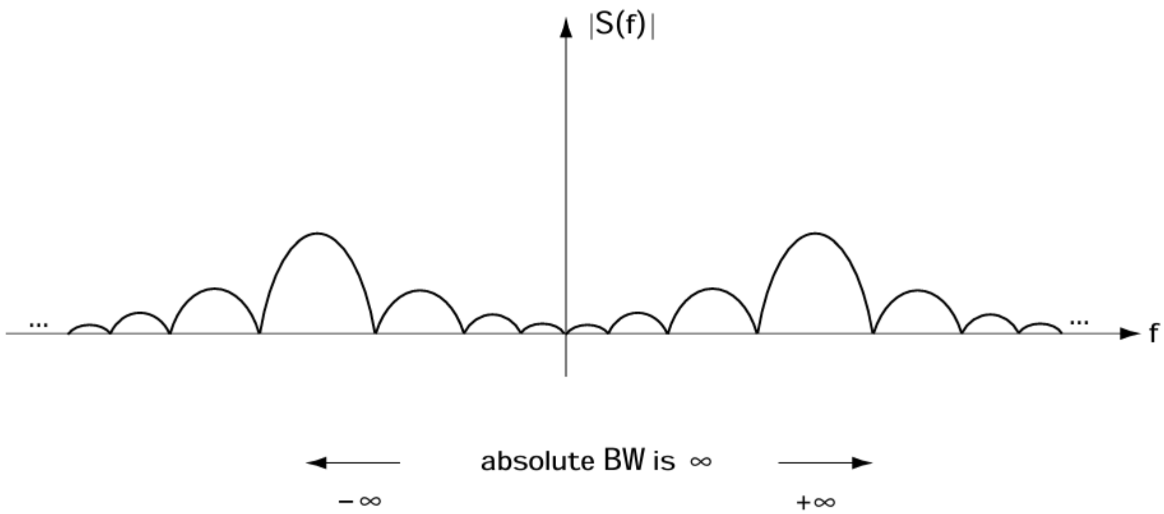
- Fourier Transform





2.3 Signals and Systems

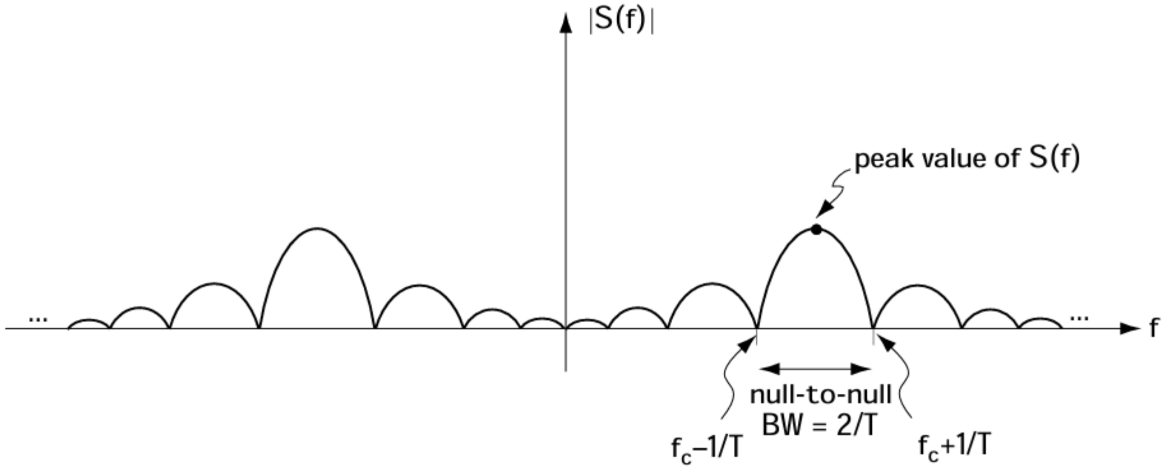
- Bandwidth



(a)

2.3 Signals and Systems

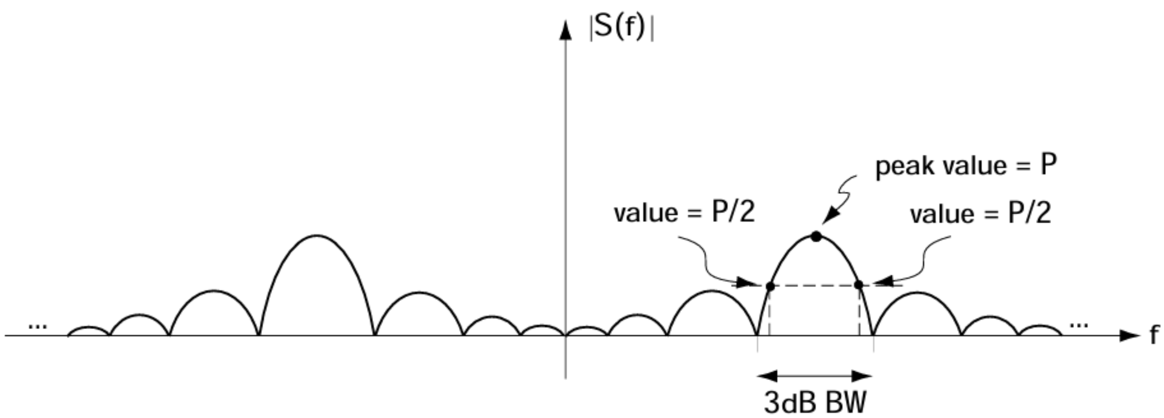
- Bandwidth



(b)

2.3 Signals and Systems

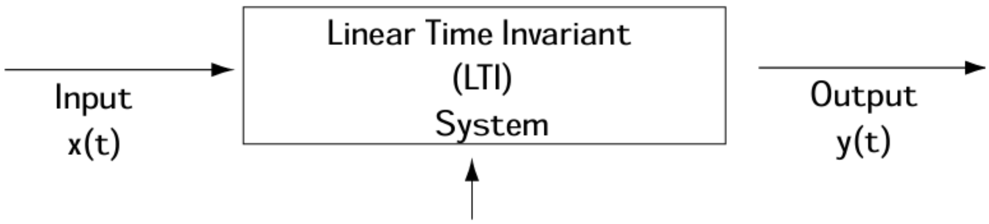
- Bandwidth



(c)

2.3 Signals and Systems

- Linear Time Invariant (LTI) System

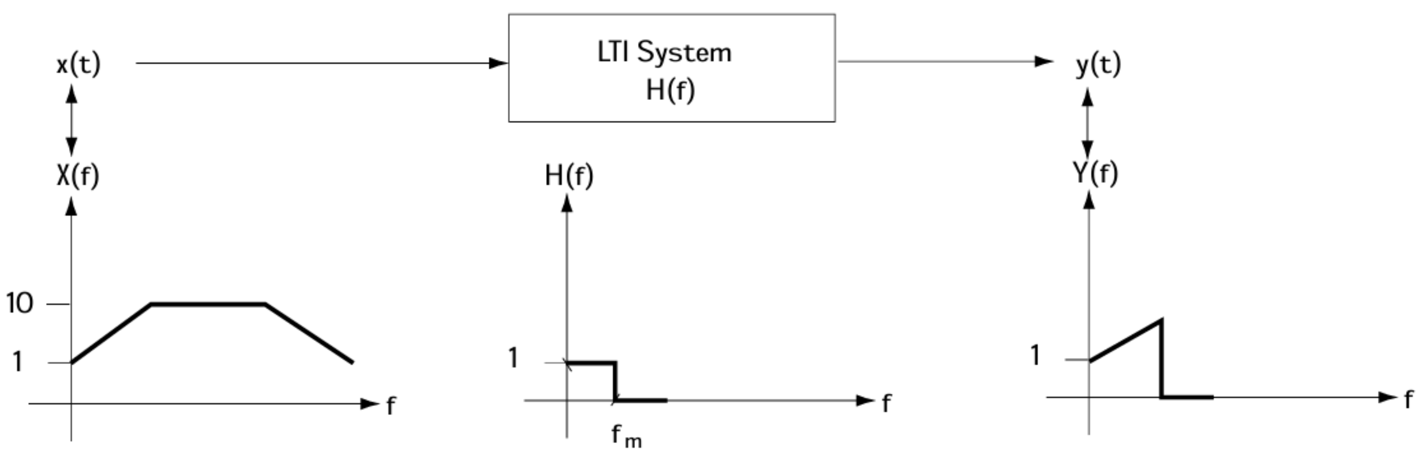


$$y(t) = h(t) * x(t) = \int_{-\infty}^{\infty} h(\tau)x(t-\tau)d\tau$$

$$Y(f) = H(f)X(f)$$

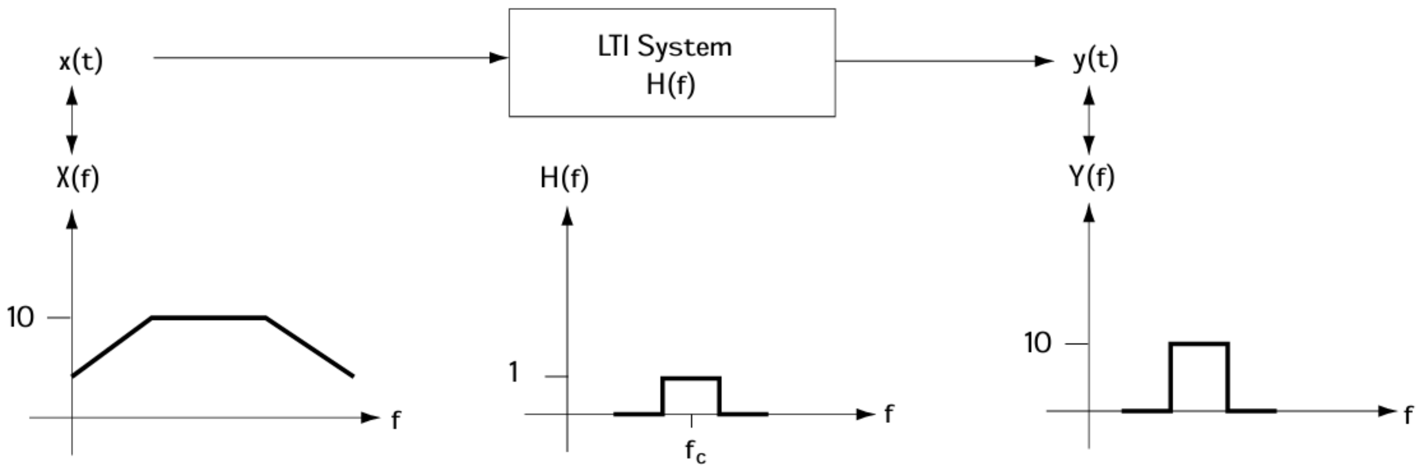
2.3 Signals and Systems

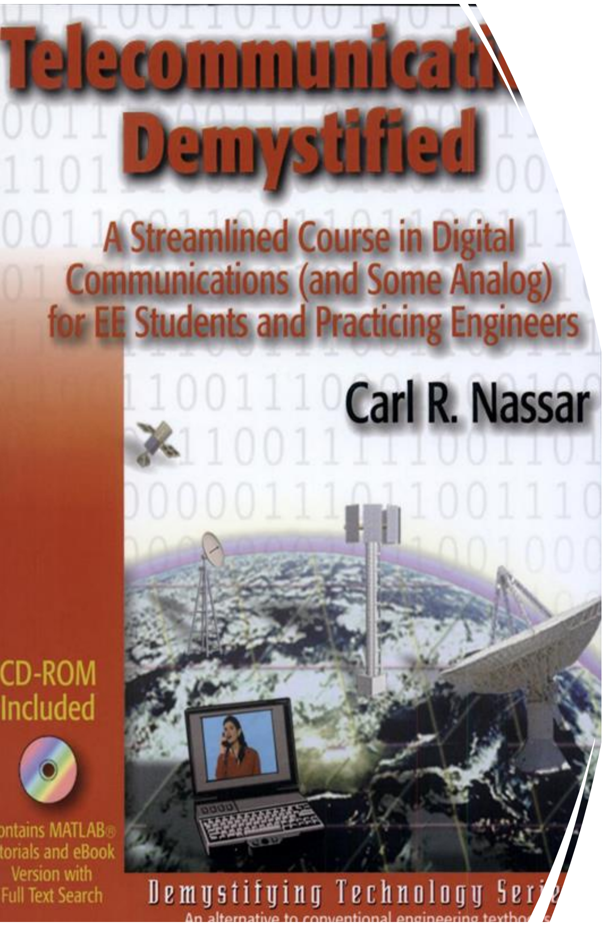
- Linear Time Invariant (LTI) System



2.3 Signals and Systems

- Linear Time Invariant (LTI) System



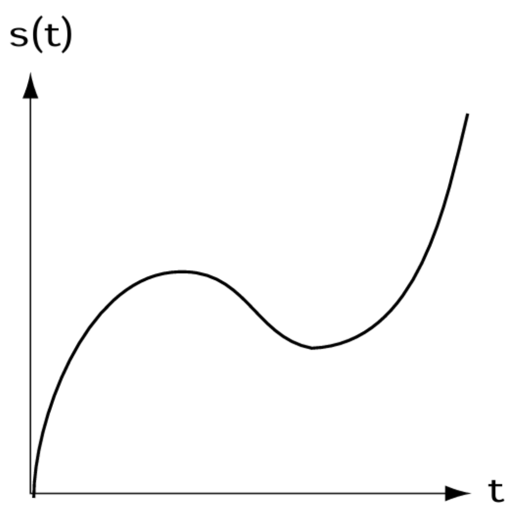


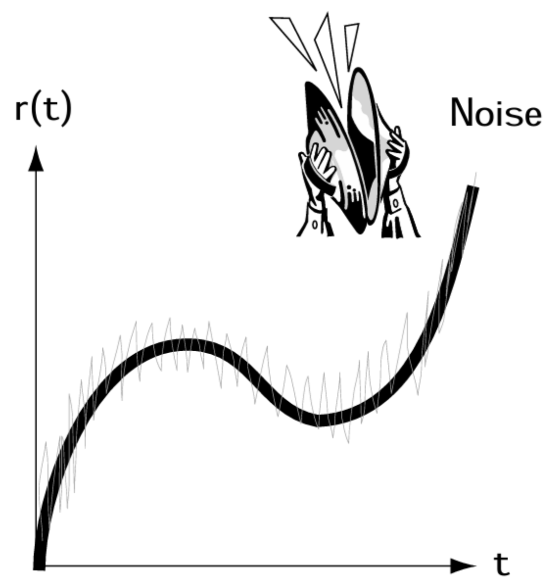
3. Source Coding and Decoding

3.1 Sampling
3.2 Quantization
3.3 Source Coding

3. Source Coding and Decoding

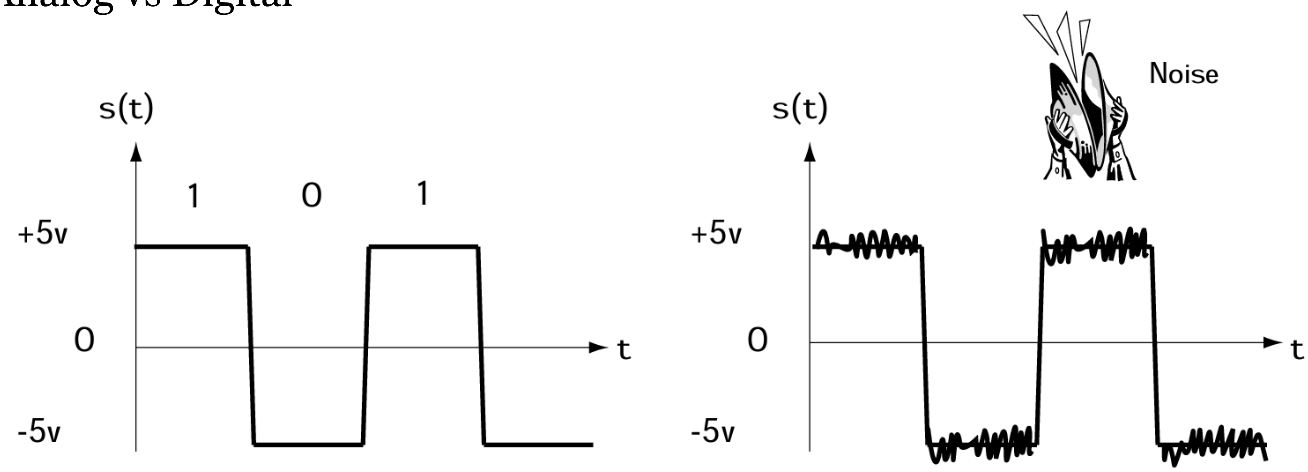
- Analog vs Digital





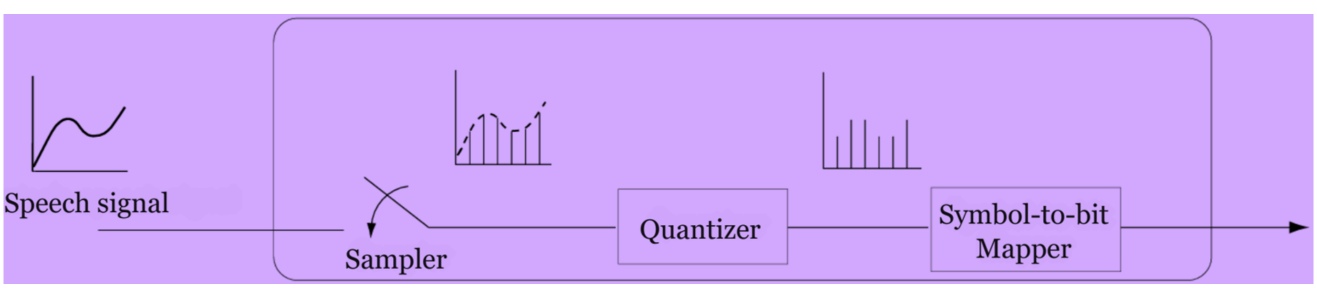
3. Source Coding and Decoding

- Analog vs Digital



3. Source Coding and Decoding

- This chapter talks about how to turn an analog signal into a digital one, a process called source coding.
- Before going on, just a brief reminder about why we want to turn analog signals to digital. So many naturally occurring sources of information are analog (human speech, for example), and we want to make them digital signals so we can use a digital communication system.



3.1 Sampling

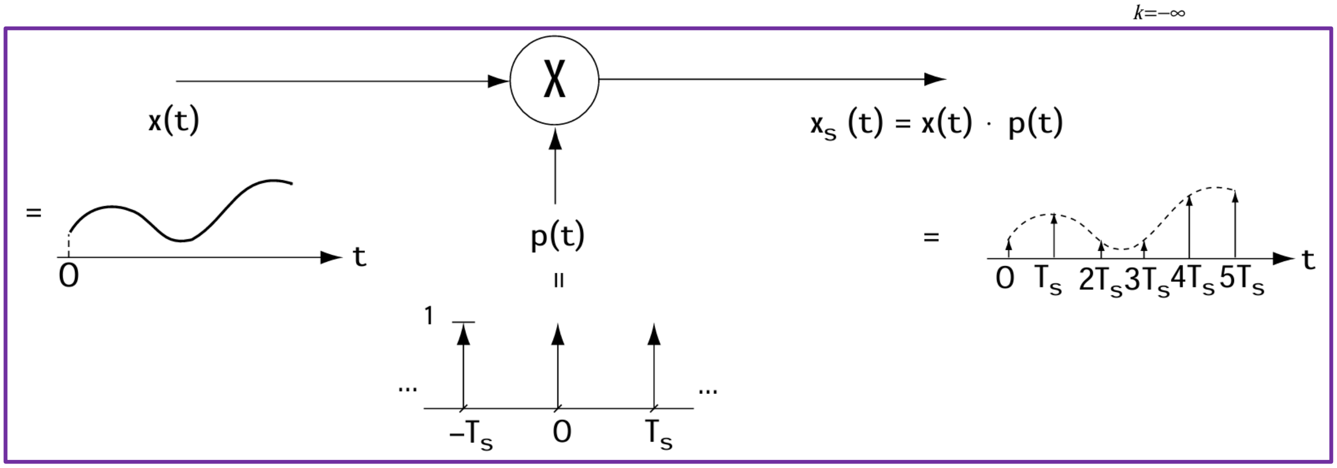
- The key first step in turning any analog signal to a digital one is called sampling. *Sampling* is the changing of an analog signal to samples (or pieces) of itself.
- There are three methods of sampling that we'll look at together, Ideal Sampling, Zero-order Hold Sampling, Natural Sampling

3.1 Sampling

- Ideal Sampling

$$p(t) = \sum_{k=-\infty}^{\infty} \delta(t - kT_s)$$

$$x_s(t) = x(t) \cdot p(t) = x(t) \cdot \sum_{k=-\infty}^{\infty} \delta(t - kT_s)$$



3.1 Sampling

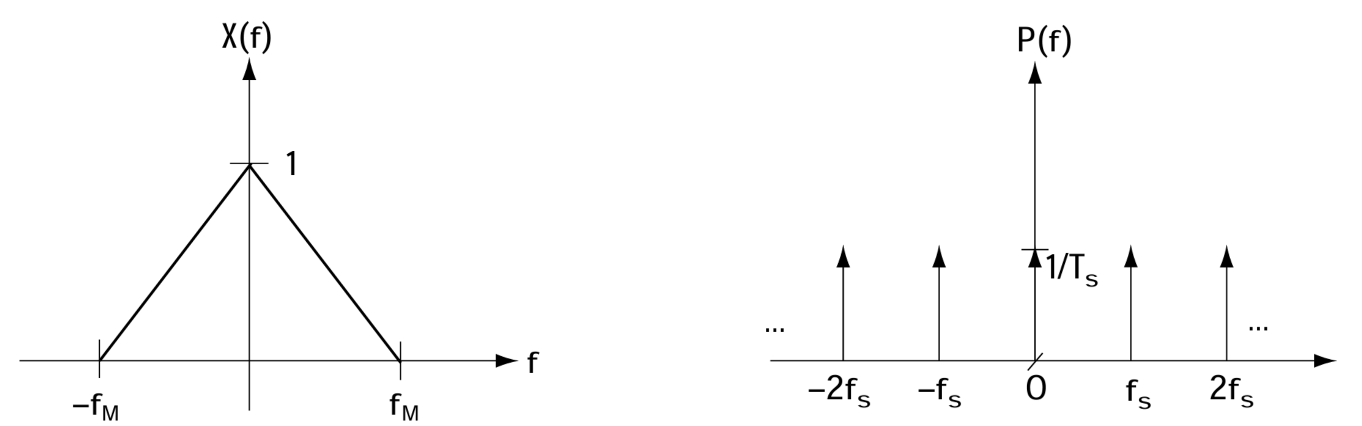
- Ideal Sampling

$$p(t) = \sum_{k=-\infty}^{\infty} \delta(t - kT_s) \xrightarrow[\text{transform}]{\text{Fourier}} P(f) = \frac{1}{T_s} \sum_{k=-\infty}^{\infty} \delta(f - k f_s)$$

- where $f_s = 1/T_s$ and f_s is called the *sampling rate*.

3.1 Sampling

- Ideal Sampling

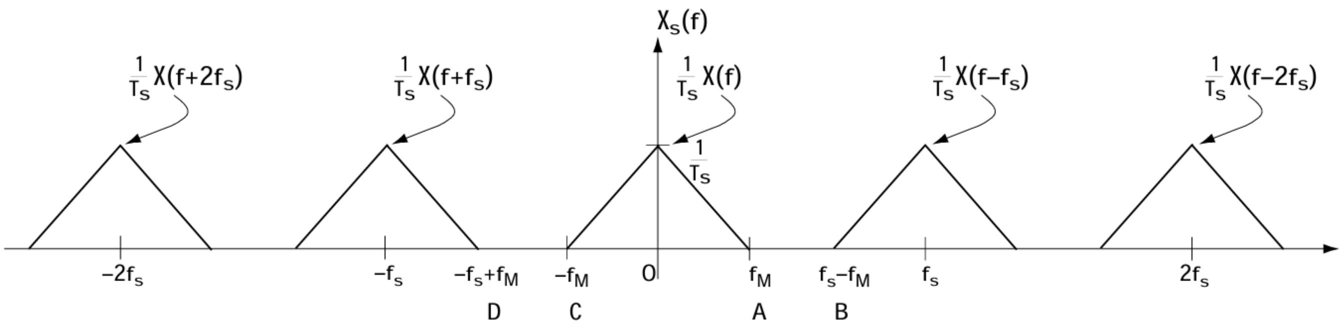


Fourier transform of the input signal x(t)

Fourier transform of the impulse train p(t)

3.1 Sampling

- Ideal Sampling

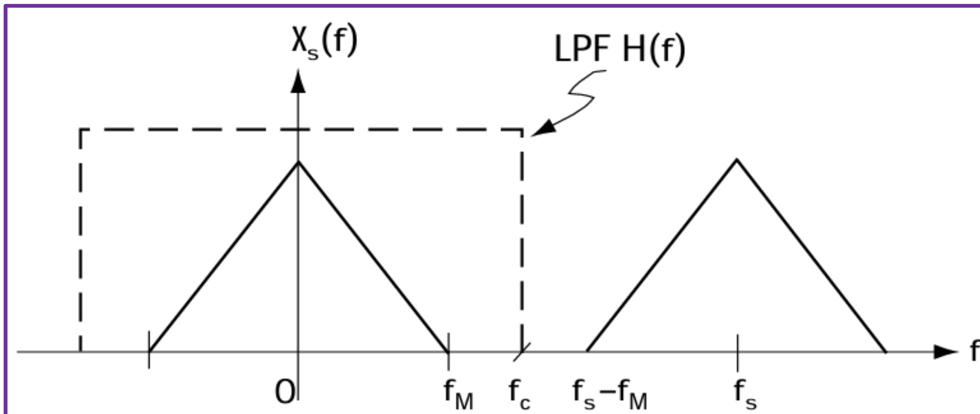


Fourier transform of the ideal sampling output $x_s(t)$

- The sampling theorem simply states that a signal can be recovered from its samples as long as it is sampled at $f_s > 2f_m$.

3.1 Sampling

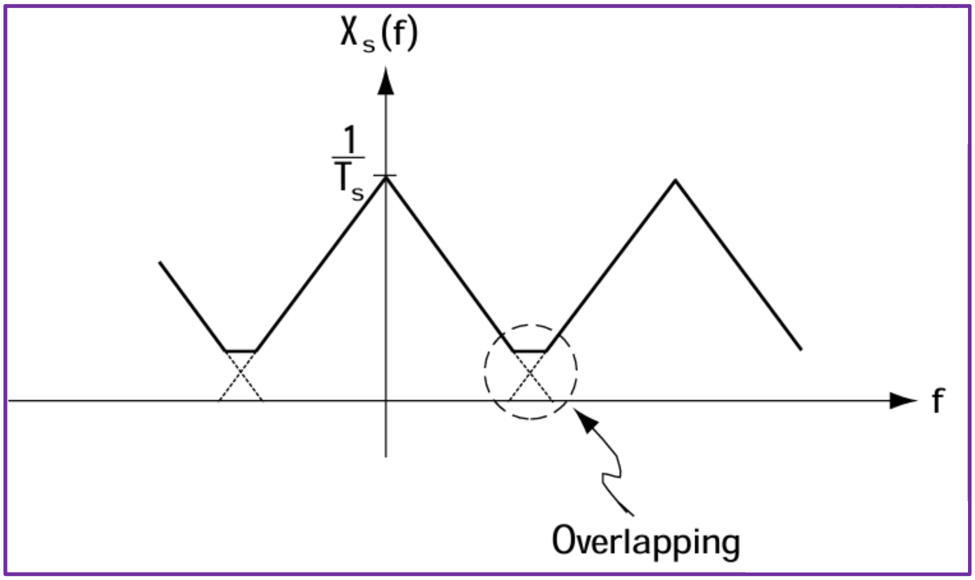
- Ideal Sampling



- The *Nyquist rate* is the smallest *sampling rate* f_s that can be used if you want to recover the original signal from its samples. From what we just saw, we know that $f_N = 2f_M$.

3.1 Sampling

- Ideal Sampling



3.1 Sampling

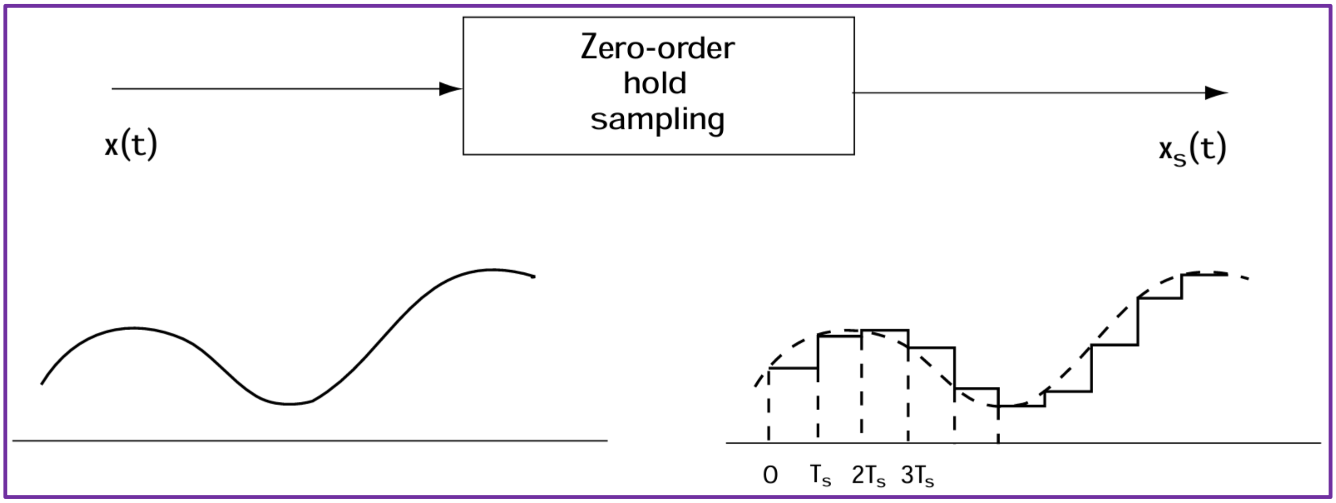
- Example 1: Determine the Nyquist sampling rate for the following signals.

(a) $x(t) = \dfrac{\sin(4000\pi\, t)}{\pi\, t}$

(b) $x(t) = \dfrac{\sin^2(4000\pi\, t)}{\pi^2\, t^2}$

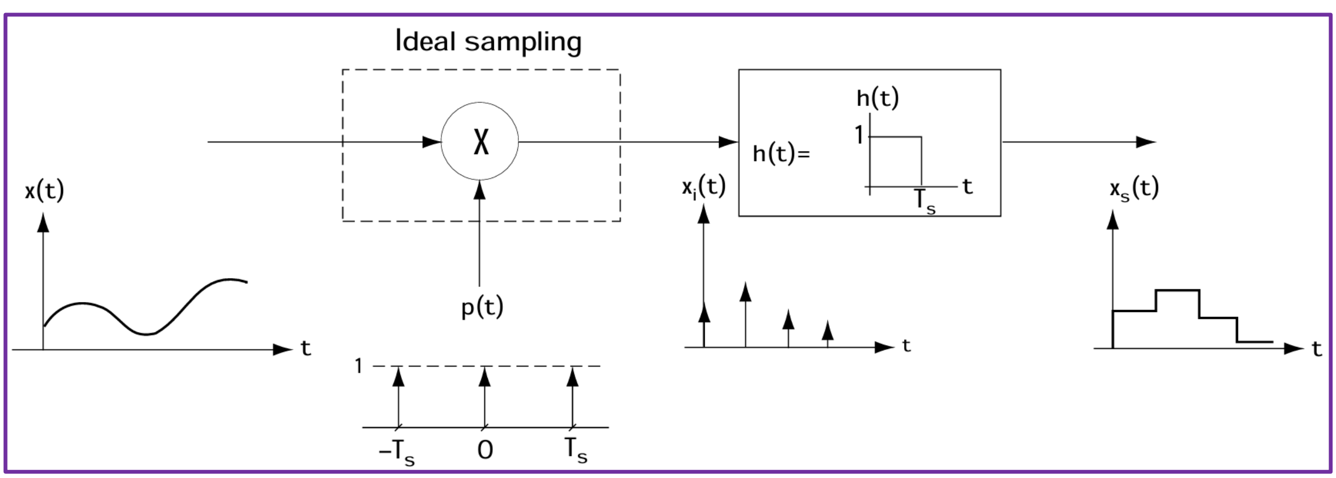
3.1 Sampling

- Zero-order Hold Sampling



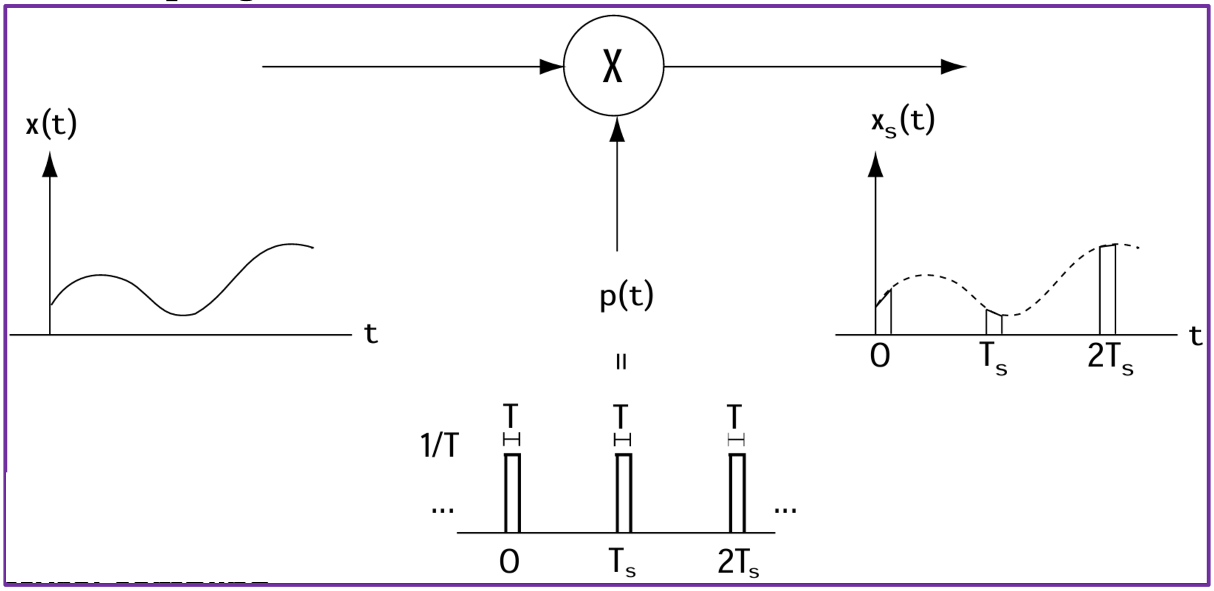
3.1 Sampling

- Zero-order Hold Sampling

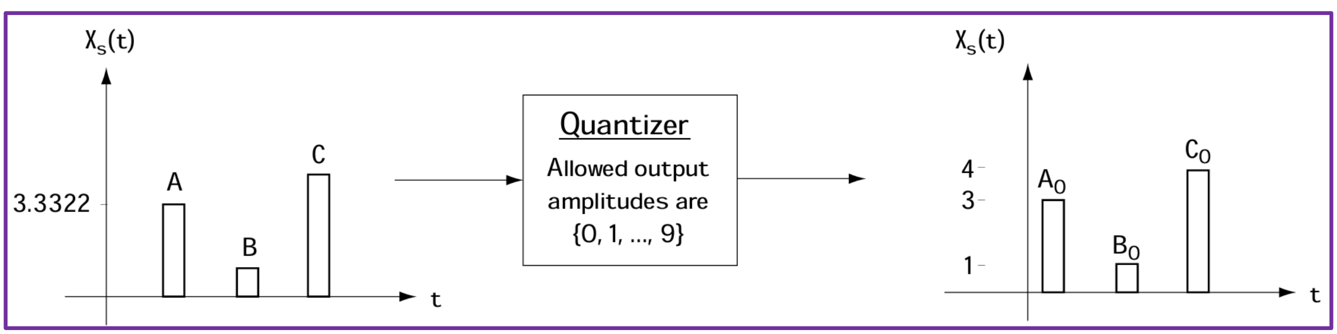


3.1 Sampling

- Natural Sampling



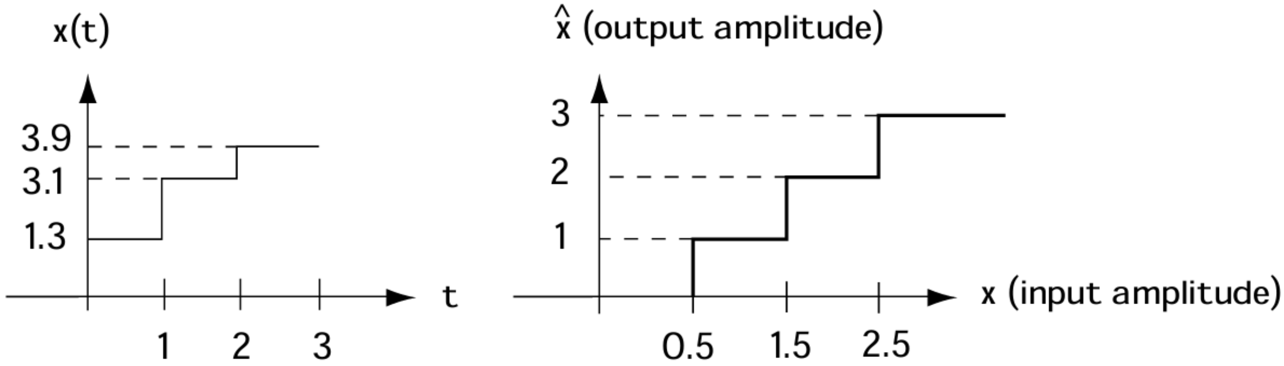
3.2 Quantization



- The operation carried out in source coding is called *quantization*, and the device which does it is called a *quantizer*.
- It is actually just an "*amplitude changer*"

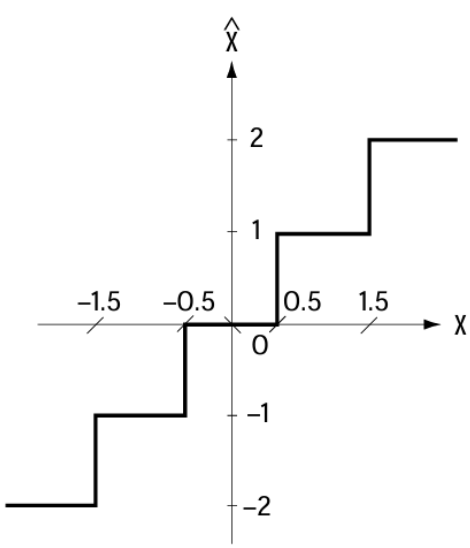
3.2 Quantization

- Example 2: Consider the quantizer with the input shown in Figure (a) and with an input amplitude–output amplitude relationship drawn in Figure (b). Draw a plot of its output.

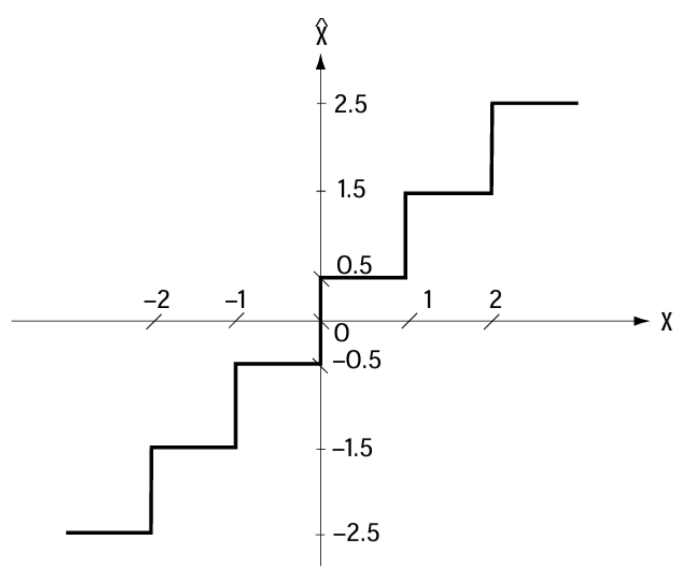


(a) (b)

3.2 Quantization

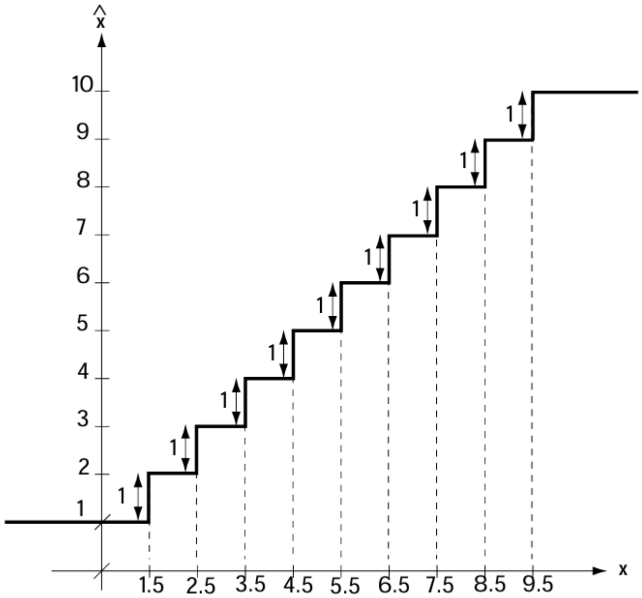


mid-tread quantizer

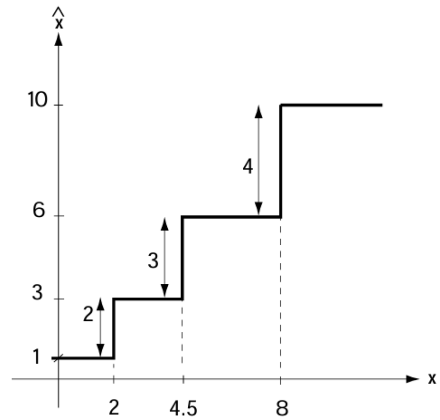


mid-riser quantizer

3.2 Quantization



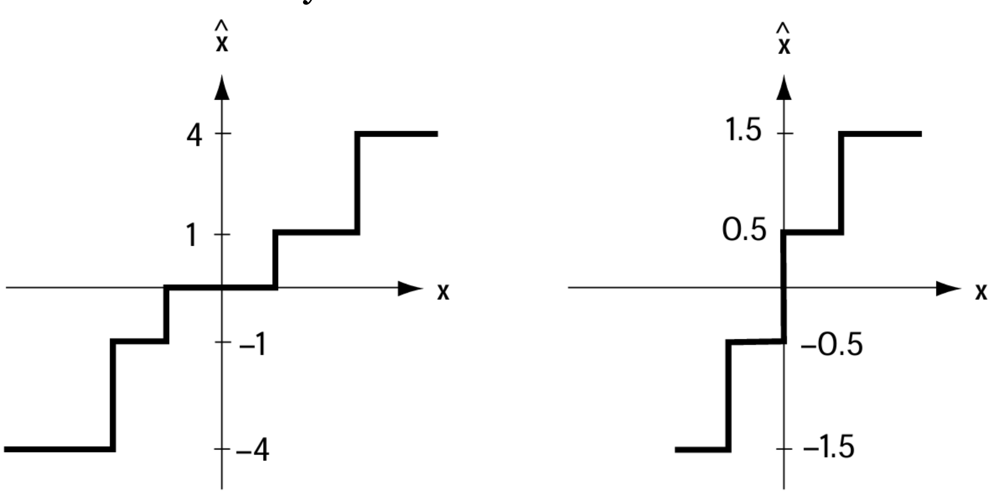
uniform quantizer



non-uniform quantizer

3.2 Quantization

- Example 3: Looking at the quantizers in Figure, determine if they are mid-tread or mid rise and if they are uniform or non-uniform.



(a) (b)

3.2 Quantization

• Measures of Performance

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

$$mse = E\left[(x - \hat{x})^2\right] = \int_{-\infty}^{\infty} (x - \hat{x})^2 p_x(x)\,dx$$

$$SQNR = \frac{P_s}{P_e} = \frac{\int_{-\infty}^{\infty} (x - x_m)^2 p_x(x)\,dx}{mse}$$

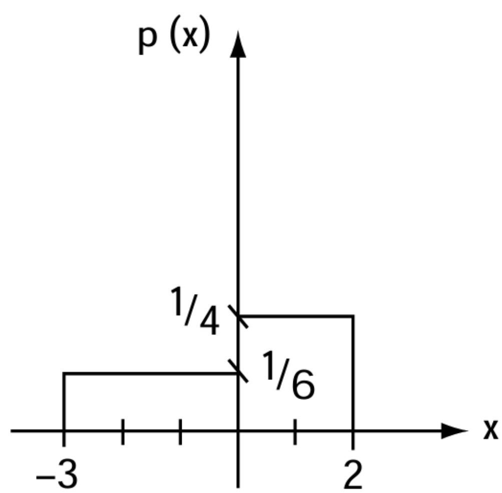
3.2 Quantization

• Example 4: Consider a quantizer with an input described in Figure

(a) Draw a quantizer with 7 levels. Make it mid-tread, let it have –3 as its smallest output value, and make sure that the step size is 1.

(b) Evaluate the *mse* of your quantizer given the input.

(c) Evaluate the SQNR.



3.3 Source Coding

- In this section we'll put samplers and quantizers together to build a source coder.
- Because there are other ways to build source coders, as we'll see later.
- This source coder is given a very particular name—the *pulse code modulator (PCM)*.

$$\text{bit rate} = \text{symbol rate} \times \#\ \text{of bits}/\text{symbol}$$

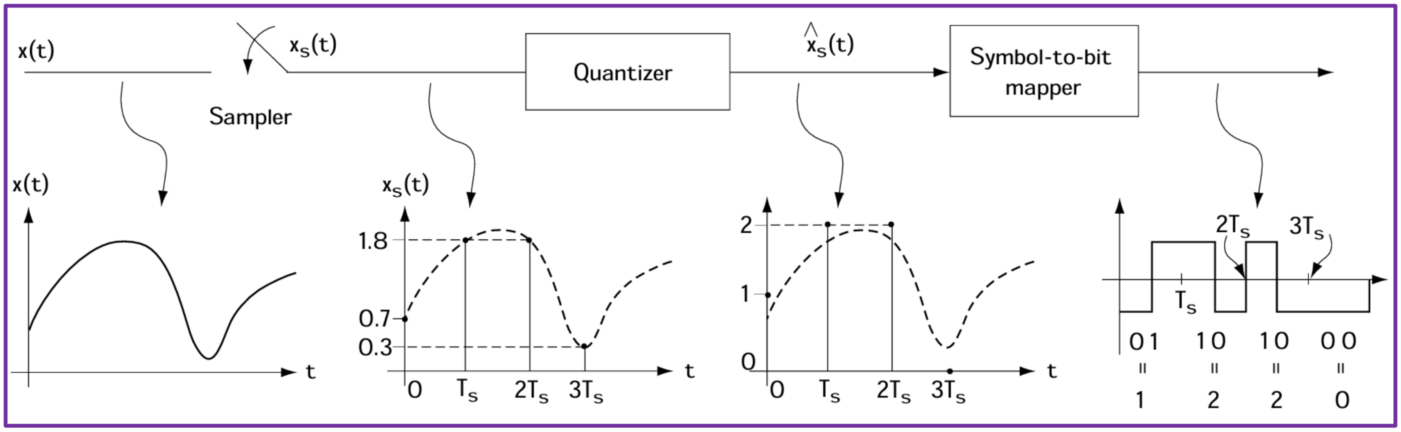
3.3 Source Coding

- Example 5: A computer sends:
    - 100 letters every 4 seconds
    - 8 bits to represent each letter
    - the bits enter a special coding device that takes in a set of bits and puts out one of 32 possible symbols.

What is the bit rate and what is the symbol rate out of the special coding device?

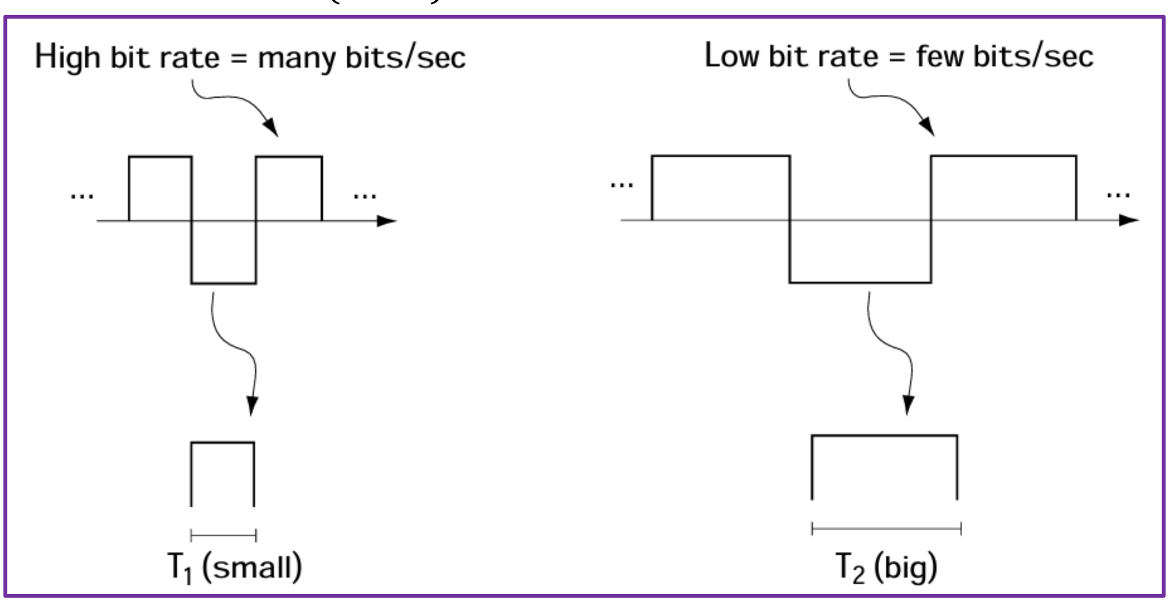
3.3 Source Coding

• Pulse Code Modulator (PCM).



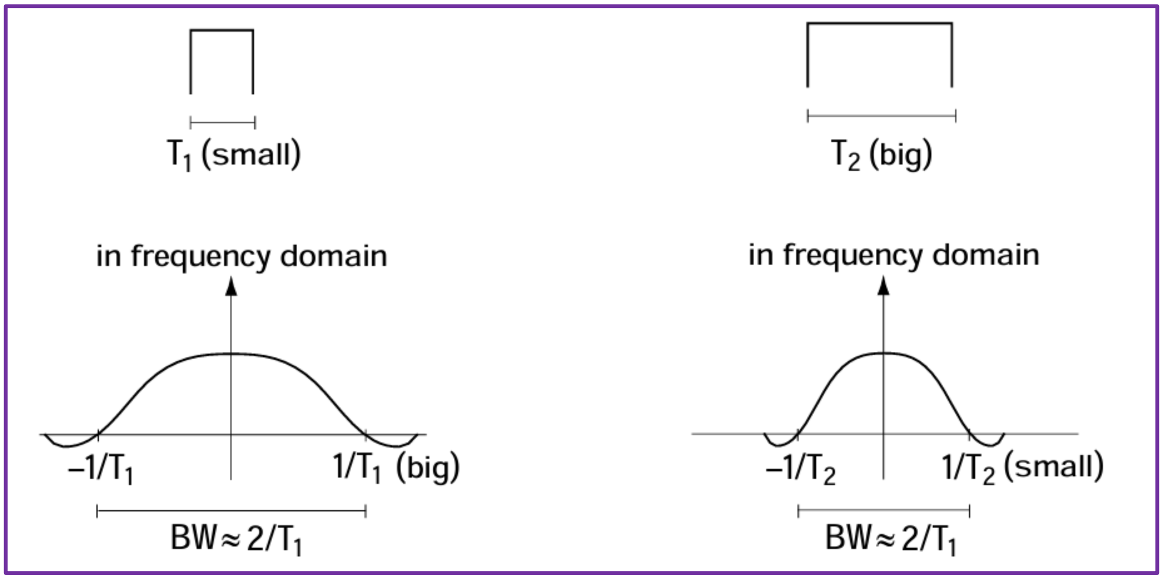
3.3 Source Coding

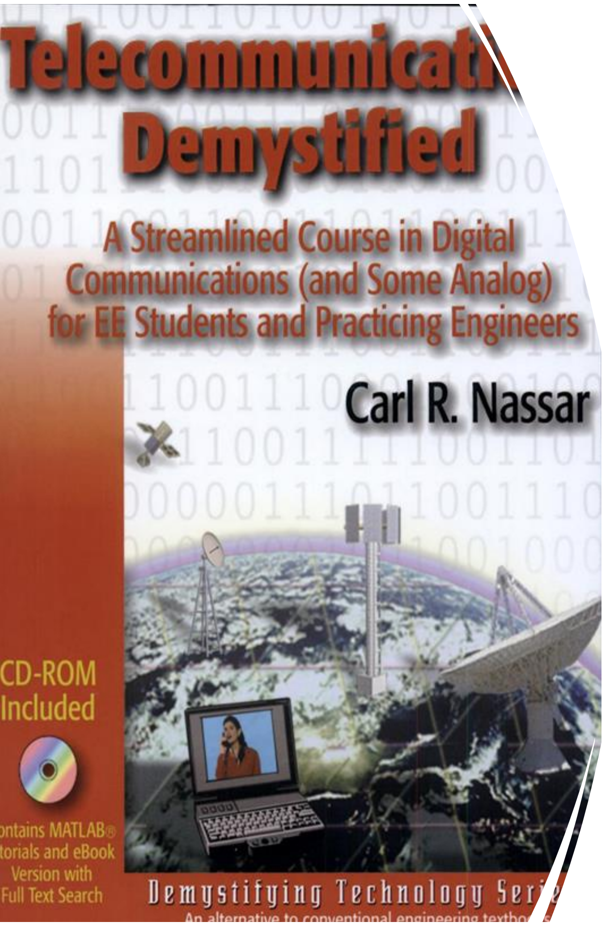
• Pulse Code Modulator (PCM).



3.3 Source Coding

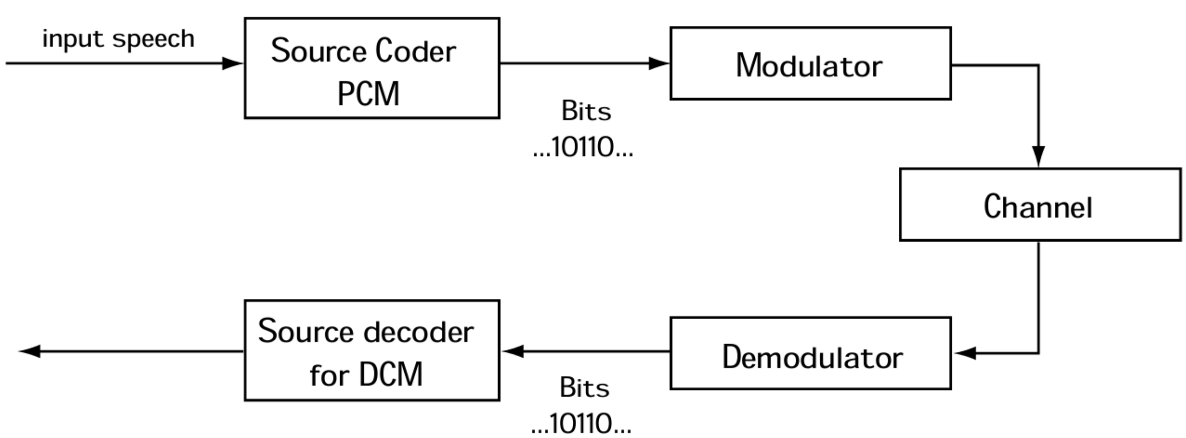
- Pulse Code Modulator (PCM).





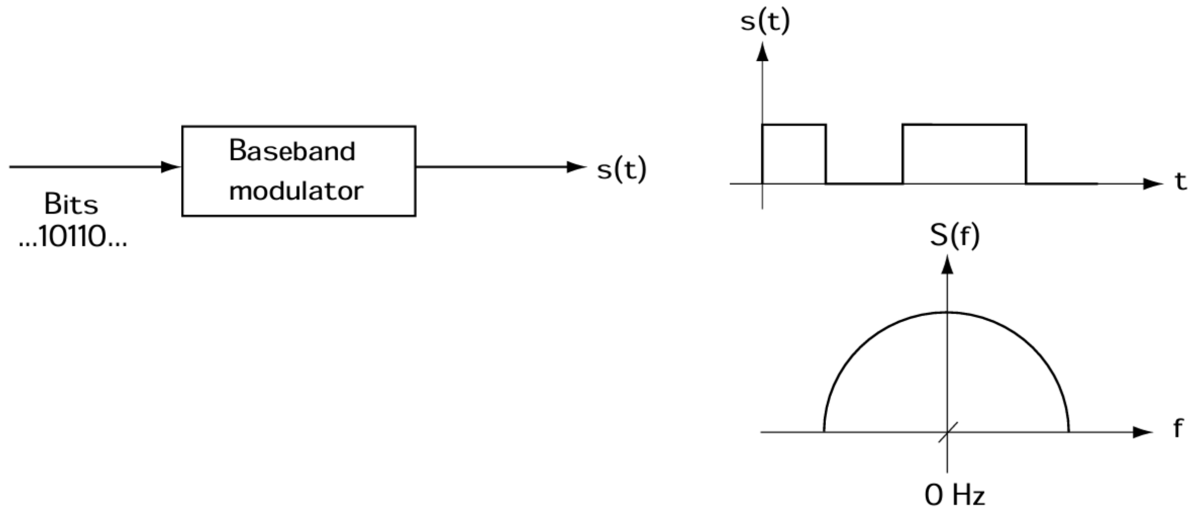
4. Modulators and Demodulators

4. Modulators and Demodulators



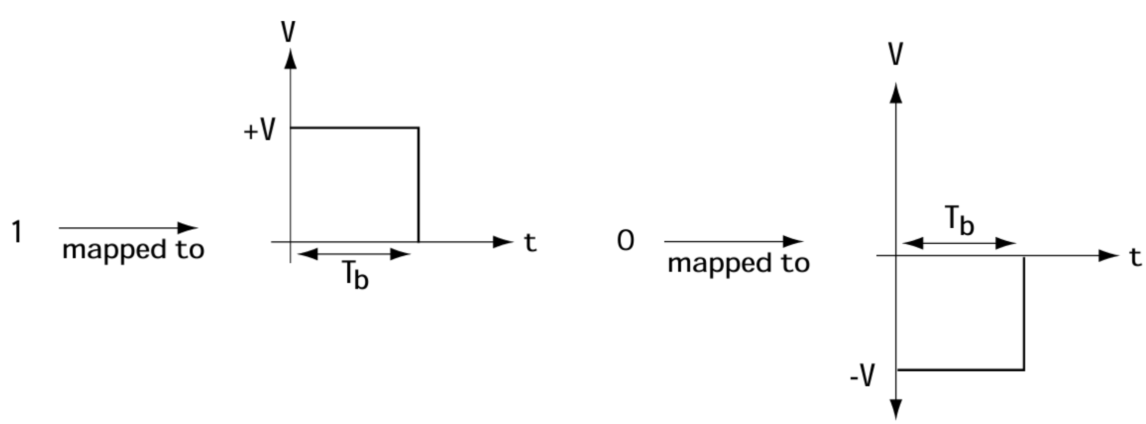
4.1 Baseband Modulator

- *Baseband modulators* are devices that turn your bit stream into a waveform centered around 0 Hz



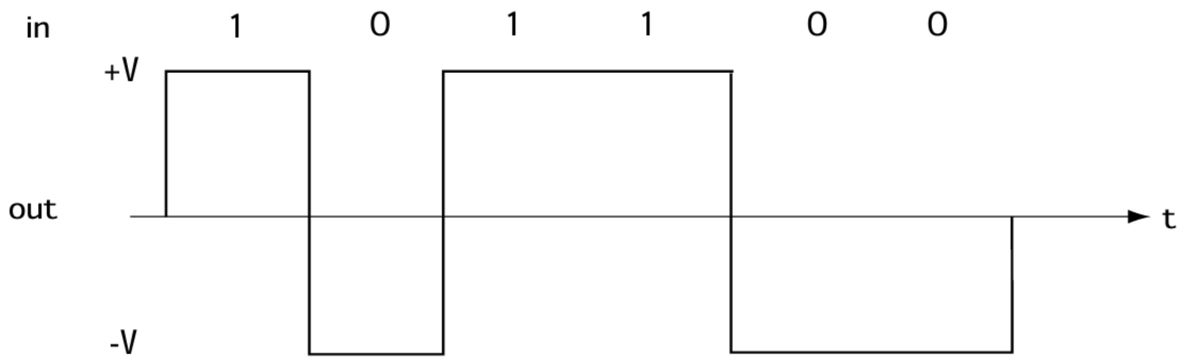
4.1 Baseband Modulator

- NRZ modulator (NRZ-L)



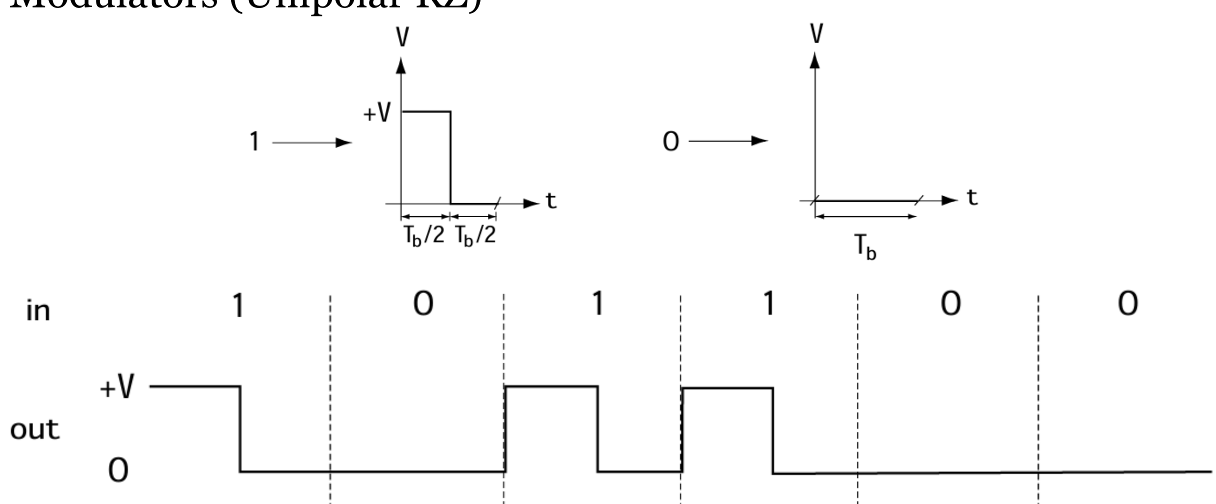
4.1 Baseband Modulator

- NRZ modulator (NRZ-L)



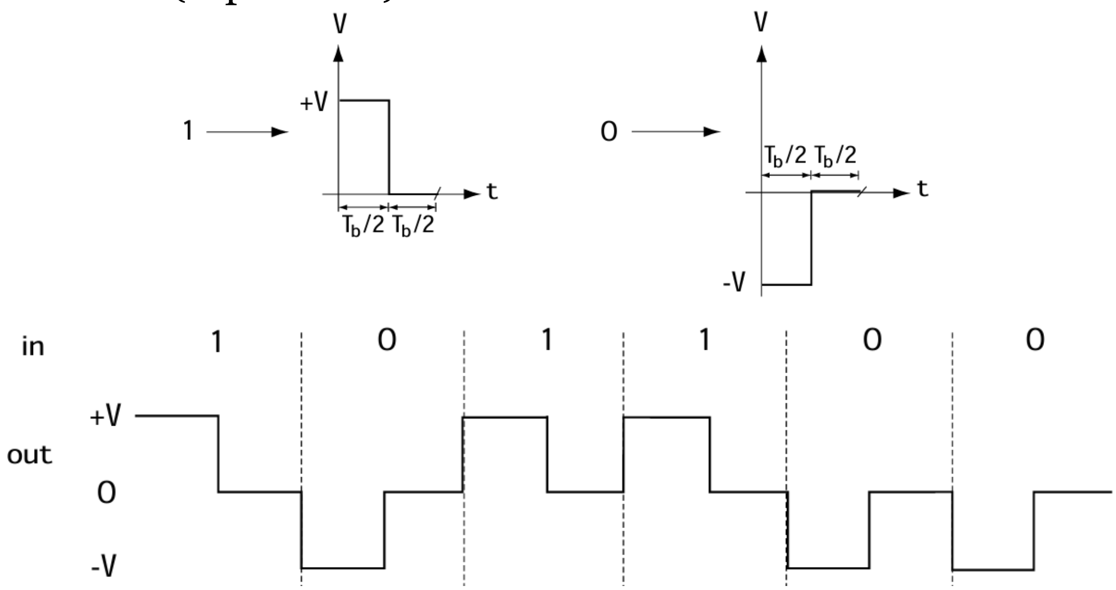
4.1 Baseband Modulator

- RZ Modulators (Unipolar RZ)



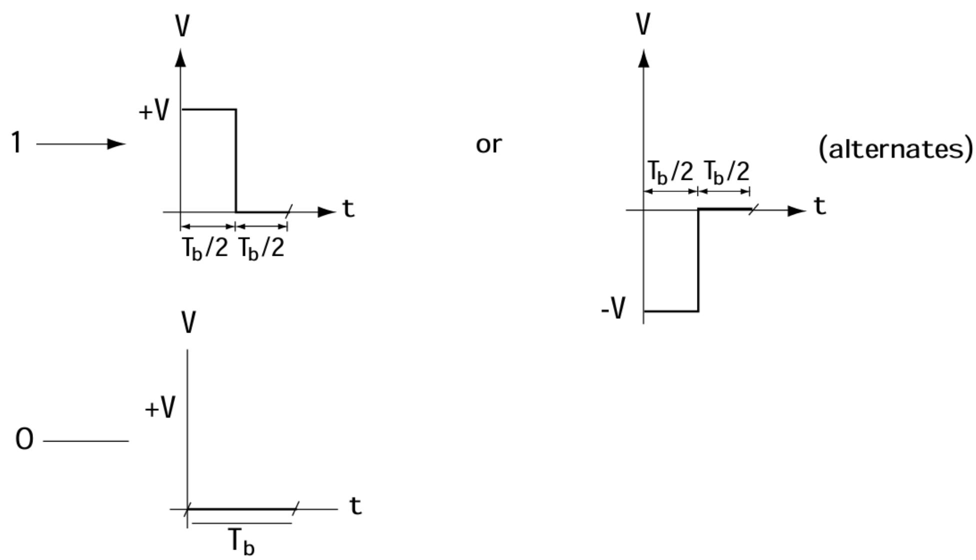
4.1 Baseband Modulator

- RZ Modulators (Bipolar RZ)



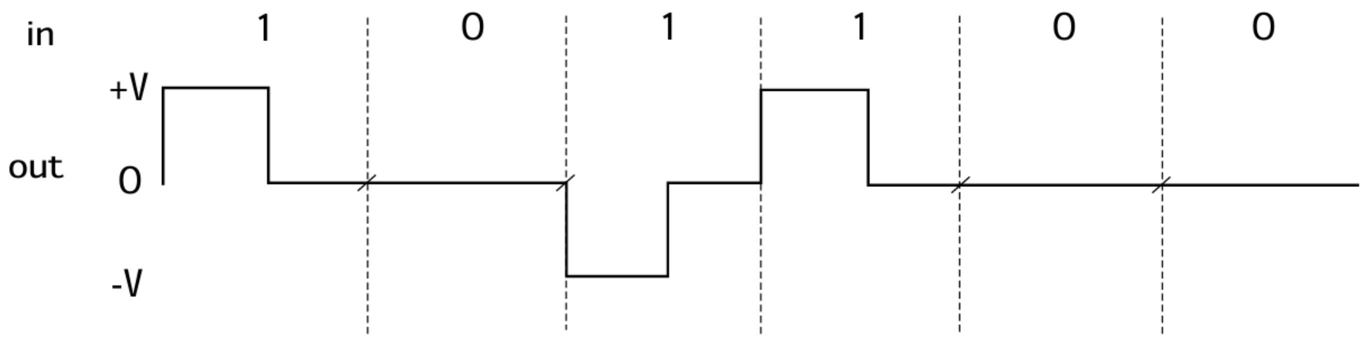
4.1 Baseband Modulator

- RZ Modulators (RZ-AMI)



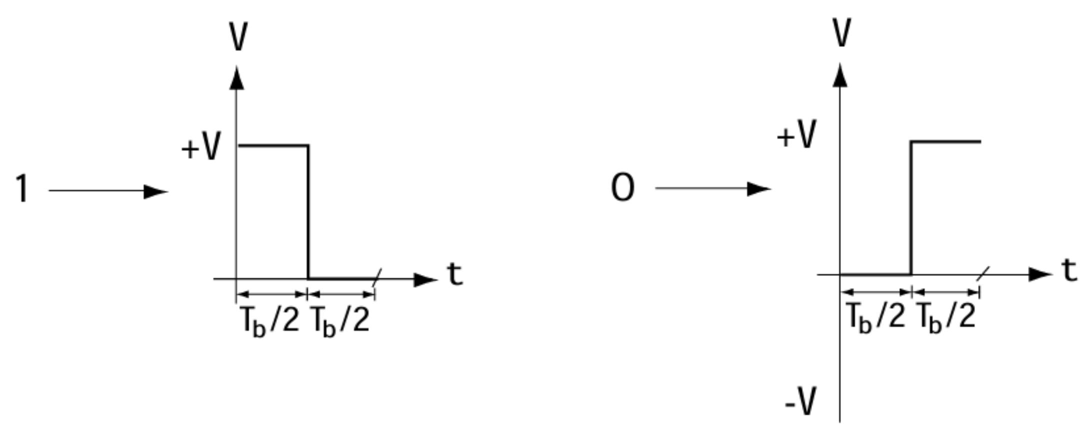
4.1 Baseband Modulator

- RZ Modulators (RZ-AMI)



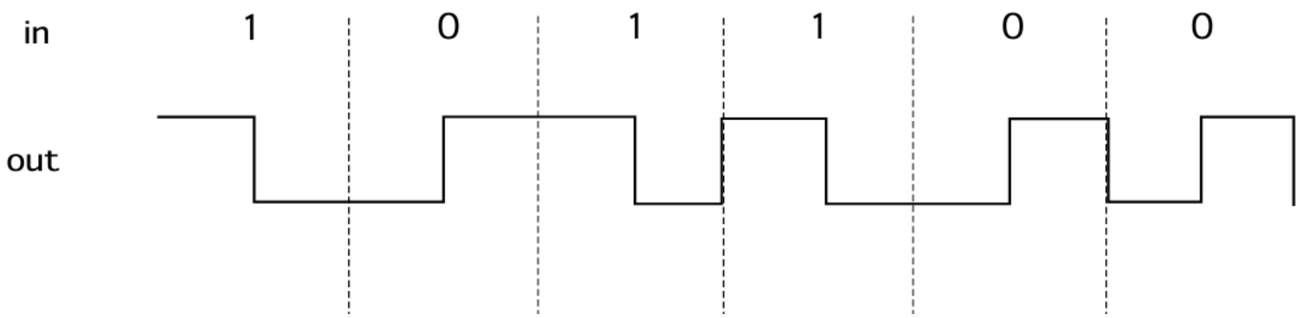
4.1 Baseband Modulator

- Manchester Coding



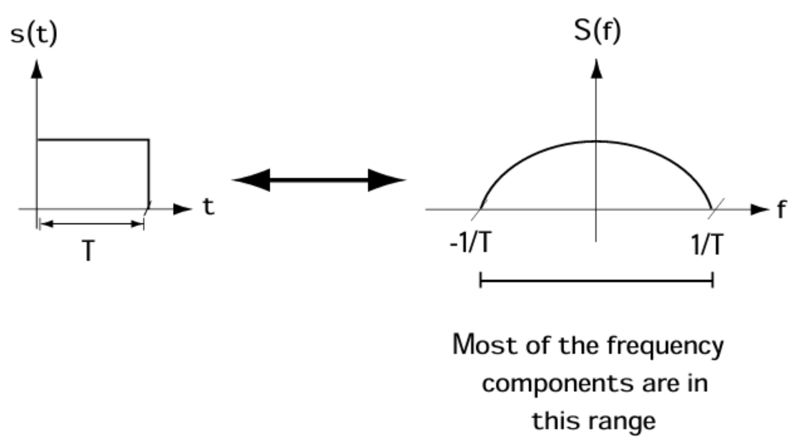
4.1 Baseband Modulator

- Manchester Coding



4.1 Baseband Modulator

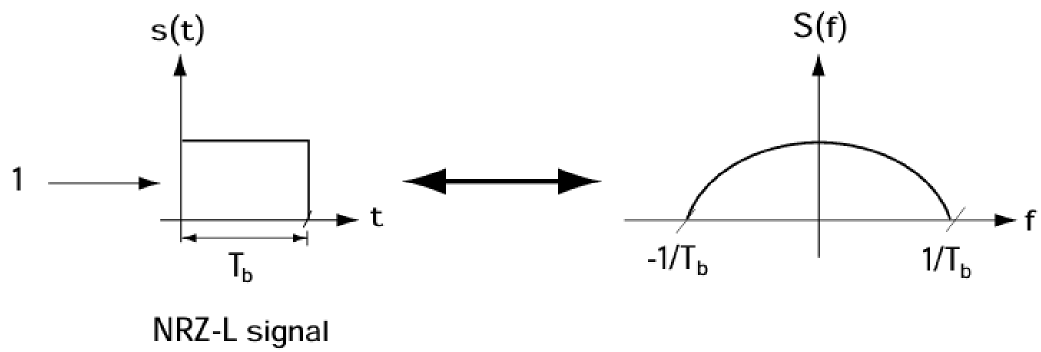
- Bandwidth considerations



(a)

4.1 Baseband Modulator

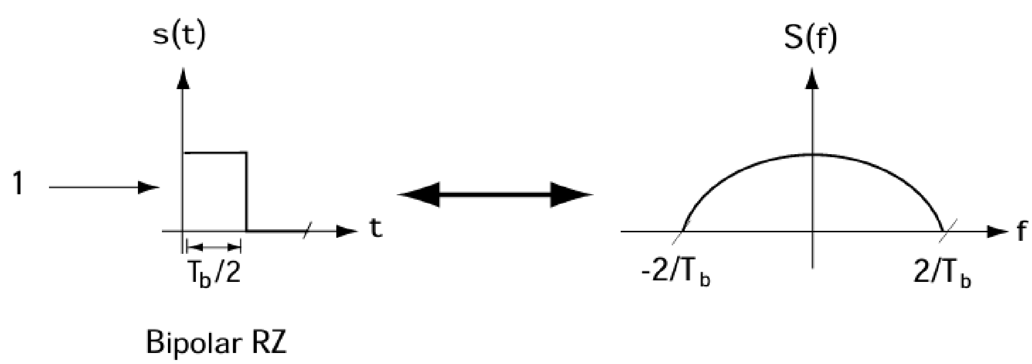
- Bandwidth considerations



(b)

4.1 Baseband Modulator

- Bandwidth considerations



(c)

4.2 Bandpass Modulator

- A bandpass modulator takes incoming bits and outputs a waveform centered around frequency ω_c
- This is the modulator you want to use when your communication channel will provide safe passage to frequencies around ω_c

$$s(t) = A\cos(\omega t + \theta)$$

4.2 Bandpass Modulator

- Amplitude Shift-Keying modulators (ASK)
- This refers to the modulators that, given the input bits, create the waveform

$$s(t) = A\cos(\omega t + \theta)$$

where the input bits are stuffed in the amplitude (A).

- We'll start with the simplest of the ASK modulators, called *Binary ASK* or *B-ASK* for short.

4.2 Bandpass Modulator

- 4-ASK

Input bits	Output waveform	Output waveform (shorthand form)
00	$s_0(t) = -3A \cos\omega_c t,\ iT \le t < (i+1)T$	$-3A \cos \omega_c t \cdot \pi\ (t-iT)$
01	$s_1(t) = -A \cos \omega_c t,\ iT \le t < (i+1)T$	$-A \cos \omega_c t \cdot \pi\ (t-iT)$
10	$s_2(t) = A \cos \omega_c t,\ iT \le t < (i+1)T$	$A \cos \omega_c t \cdot \pi\ (t-iT)$
11	$s_3(t) = 3A \cos \omega_c t,\ iT \le t < (i+1)T$	$3A \cos \omega_c t \cdot \pi\ (t-iT)$

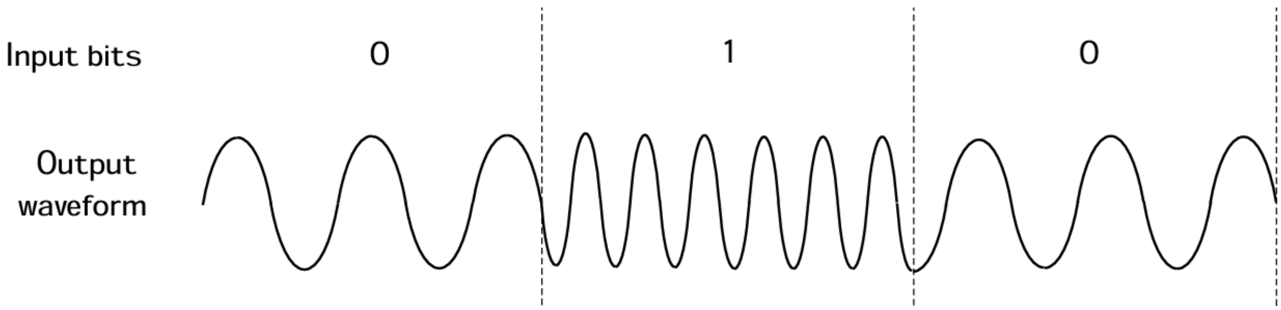
4.2 Bandpass Modulator

- 8-ASK

Input bits	Output waveform	Output waveform (shorthand form)
000	$s_0(t) = -7A \cos \omega_c t,\ iT \le t < (i+1)T$	$-7A \cos \omega_c t \cdot \pi\ (t-iT)$
001	$s_1(t) = -5A \cos \omega_c t,\ iT \le t < (i+1)T$	$-5A \cos \omega_c t \cdot \pi\ (t-iT)$
010	$s_2(t) = -3A \cos \omega_c t,\ iT \le t < (i+1)T$	$-3A \cos \omega_c t \cdot \pi\ (t-iT)$
011	$s_3(t) = -A \cos \omega_c t,\ iT \le t < (i+1)T$	$-A \cos \omega_c t \cdot \pi\ (t-iT)$
100	$s_4(t) = A \cos \omega_c t,\ iT \le t < (i+1)T$	$A \cos \omega_c t \cdot \pi\ (t-iT)$
101	$s_5(t) = 3A \cos \omega_c t,\ iT \le t < (i+1)T$	$3A \cos \omega_c t \cdot \pi\ (t-iT)$
110	$s_6(t) = 5A \cos \omega_c t,\ iT \le t < (i+1)T$	$5A \cos \omega_c t \cdot \pi\ (t-iT)$
111	$s_7(t) = 7A \cos \omega_c t,\ iT \le t < (i+1)T$	$7A \cos \omega_c t \cdot \pi\ (t-iT)$

4.2 Bandpass Modulator

- Frequency Shift-Keying modulators (FSK)
- As the name suggests, here we stuff the information bits into the frequency.
- We'll look at the simplest first, which is *BFSK* (*binary FSK*).



4.2 Bandpass Modulator

- Frequency Shift-Keying modulators (FSK)

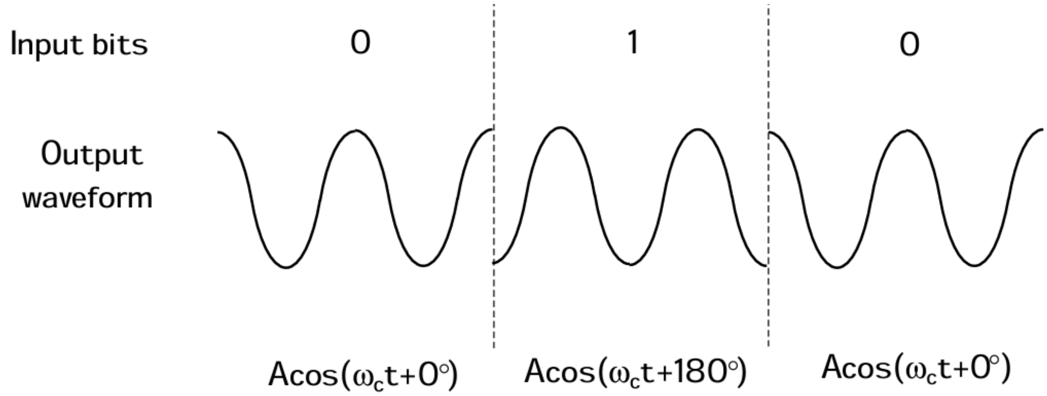
	Input bits	Output waveform	Output waveform (shorthand)
BFSK	0	$s_0(t) = A\cos((\omega_c+\Delta\omega_0)t),\ iT \le t < (i+1)T$	$A\cos((\omega_c+\Delta\omega_0)t)\cdot\pi(t-iT)$
	1	$s_1(t) = A\cos((\omega_c+\Delta\omega_1)t),\ iT \le t < (i+1)T$	$A\cos((\omega_c+\Delta\omega_1)t)\cdot\pi(t-iT)$
4-FSK	00	$s_0(t) = A\cos((\omega_c+\Delta\omega_0)t),\ iT \le t < (i+1)T$	$A\cos((\omega_c+\Delta\omega_0)t)\cdot\pi(t-iT)$
	01	$s_1(t) = A\cos((\omega_c+\Delta\omega_1)t),\ iT \le t < (i+1)T$	$A\cos((\omega_c+\Delta\omega_1)t)\cdot\pi(t-iT)$
	10	$s_2(t) = A\cos((\omega_c+\Delta\omega_2)t),\ iT \le t < (i+1)T$	$A\cos((\omega_c+\Delta\omega_2)t)\cdot\pi(t-iT)$
	11	$s_3(t) = A\cos((\omega_c+\Delta\omega_3)t),\ iT \le t < (i+1)T$	$A\cos((\omega_c+\Delta\omega_3)t)\cdot\pi(t-iT)$

4.2 Bandpass Modulator

- Phase Shift-Keying modulators (PSK)
- With these, input bits are mapped into output waveforms of the form

$$s(t)=A\cos(\omega t+\theta)$$

and the information bits are stuffed in the phase θ.



4.2 Bandpass Modulator

- Phase Shift-Keying modulators (PSK)

	Input bits	Output waveform	Output waveform (shorthand form)
BPSK	0	$s_0(t) = A\cos(\omega_c t+0°),\ iT \le t < (i+1)T$	$A\cos(\omega_c t+0°) \cdot \pi(t-iT)$
	1	$s_1(t) = A\cos(\omega_c t+180°),\ iT \le t < (i+1)T$	$A\cos(\omega_c t+180°) \cdot \pi(t-iT)$
4-PSK	00	$s_0(t) = A\cos(\omega_c t+0°),\ iT \le t < (i+1)T$	$A\cos(\omega_c t+0°) \cdot \pi(t-iT)$
	01	$s_1(t) = A\cos(\omega_c t+90°),\ iT \le t < (i+1)T$	$A\cos(\omega_c t+90°) \cdot \pi(t-iT)$
	10	$s_2(t) = A\cos(\omega_c t+180°),\ iT \le t < (i+1)T$	$A\cos(\omega_c t+180°) \cdot \pi(t-iT)$
	11	$s_3(t) = A\cos(\omega_c t+270°),\ iT \le t < (i+1)T$	$A\cos(\omega_c t+270°) \cdot \pi(t-iT)$

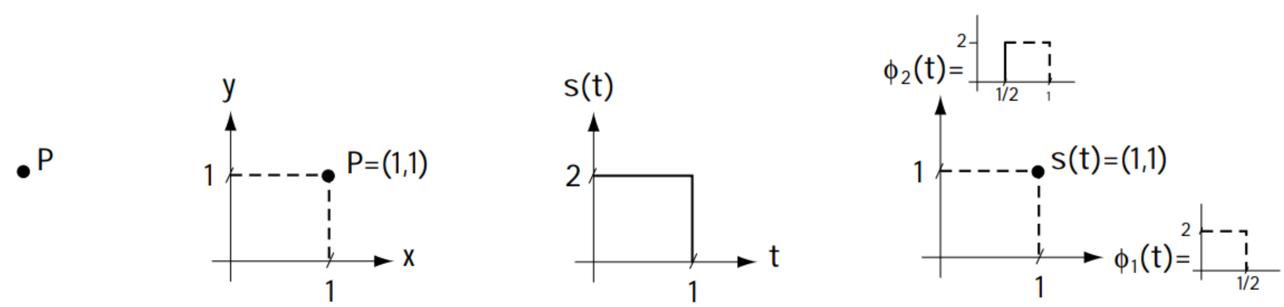
4.2 Bandpass Modulator

- Phase Shift-Keying modulators (PSK)

	Input bits	Output waveform	Output waveform (shorthand form)
8-PSK	000	$s_0(t) = A\cos(\omega_c t + 0°),\ iT \le t < (i+1)T$	
	001	$s_1(t) = A\cos(\omega_c t + 45°),\ iT \le t < (i+1)T$	
	010	$s_2(t) = A\cos(\omega_c t + 90°),\ iT \le t < (i+1)T$	
	011	$s_3(t) = A\cos(\omega_c t + 135°),\ iT \le t < (i+1)T$	
	100	$s_4(t) = A\cos(\omega_c t + 180°),\ iT \le t < (i+1)T$	
	101	$s_5(t) = A\cos(\omega_c t + 225°),\ iT \le t < (i+1)T$	
	110	$s_6(t) = A\cos(\omega_c t + 270°),\ iT \le t < (i+1)T$	
	111	$s_7(t) = A\cos(\omega_c t + 315°),\ iT \le t < (i+1)T$	

4.3 Modulator Signal

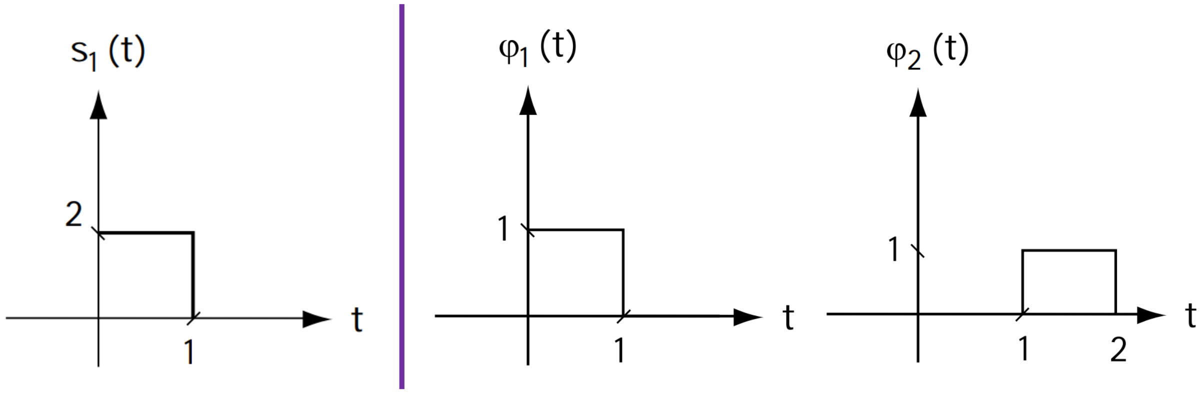
- Representing points and signals



(a) (b) (c) (d)

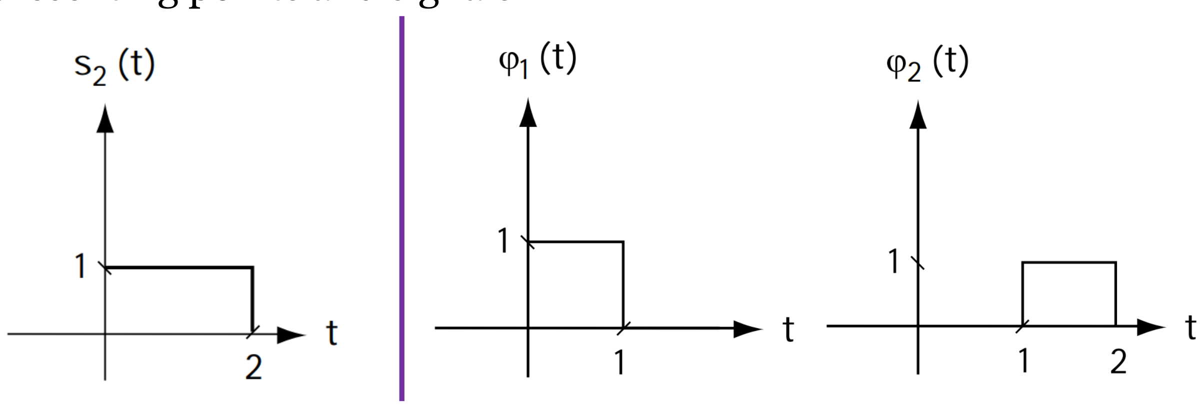
4.3 Modulator Signal

- Representing points and signals



4.3 Modulator Signal

- Representing points and signals



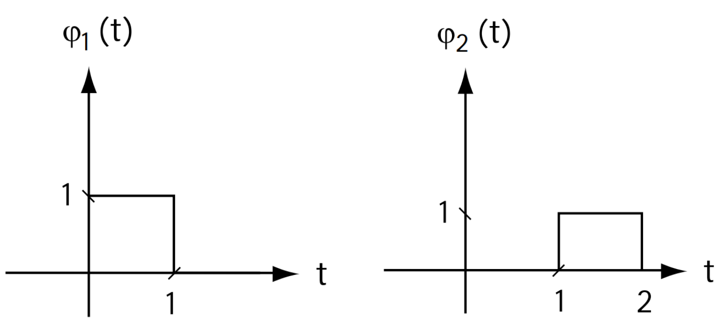
4.3 Modulator Signal

- orthonormal basis

$$\left.\begin{aligned} s_1(t) &= s_{11}\varphi_1(t) + s_{12}\varphi_2(t) + \ldots + s_{1N}\varphi_N(t) \\ &\ldots \\ s_M(t) &= s_{M1}\varphi_1(t) + s_{M2}\varphi_2(t) + \ldots + s_{MN}\varphi_N(t) \end{aligned}\right\} \quad \begin{aligned} &\int_{-\infty}^{\infty} \varphi_i(t)\varphi_j(t)\,dt = 0, i \neq j \\ &\int_{-\infty}^{\infty} \varphi_j(t)\varphi_j(t)\,dt = 1 \\ &s_{ij} = \int_{-\infty}^{\infty} s_i(t)\varphi_j(t)\,dt \end{aligned}$$

4.3 Modulator Signal

- orthonormal basis

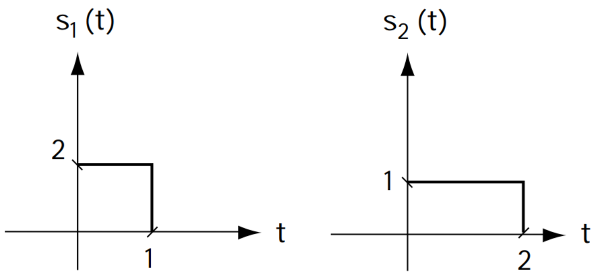


$$\int_{-\infty}^{\infty} \varphi_i(t)\varphi_j(t)\,dt = 0, i \neq j$$

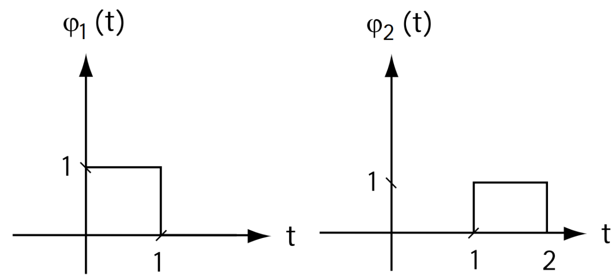
$$\int_{-\infty}^{\infty} \varphi_j(t)\varphi_j(t)\,dt = 1$$

4.3 Modulator Signal

- orthonormal basis



signals



orthonormal basis

$$s_1(t) = s_{11}\varphi_1(t) + s_{12}\varphi_2(t) + \ldots + s_{1N}\varphi_N(t)$$

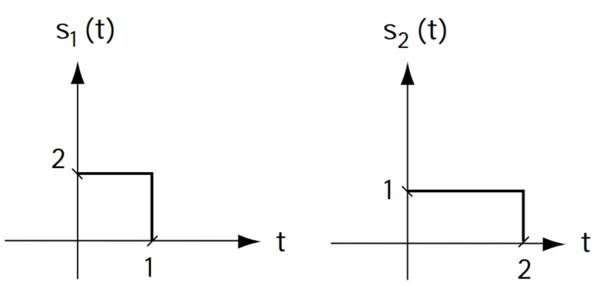
$$\ldots$$

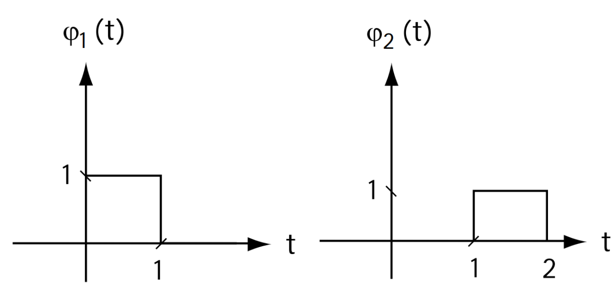
$$s_M(t) = s_{M1}\varphi_1(t) + s_{M2}\varphi_2(t) + \ldots + s_{MN}\varphi_N(t)$$

$$s_{ij} = \int_{-\infty}^{\infty} s_i(t)\varphi_j(t)\,dt$$

4.3 Modulator Signal

- orthonormal basis

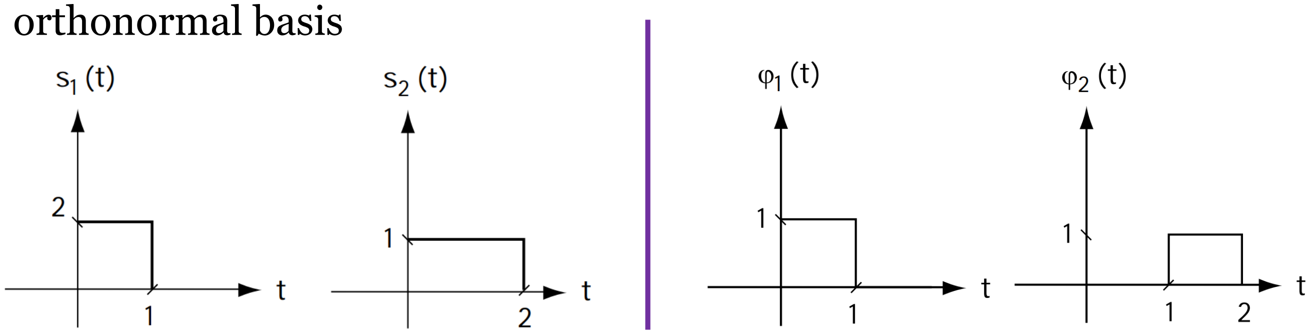




$$s_1(t) = 2\phi_1(t) + 0\phi_2(t) \quad \begin{cases} s_{11} = \int_{-\infty}^{\infty} s_1(t)\phi_1(t)dt = 2 \\ s_{12} = \int_{-\infty}^{\infty} s_1(t)\phi_2(t)dt = 0 \end{cases}$$

4.3 Modulator Signal

- orthonormal basis



$$s_2(t) = 1(t) + 1\phi_2(t) \quad \begin{cases} s_{21} = \int_{-\infty}^{\infty} s_2(t)\phi_1(t)dt = 1 \\ \\ s_{22} = \int_{-\infty}^{\infty} s_2(t)\phi_2(t)dt = 1 \end{cases}$$

4.3 Modulator Signal

- Gram-Schmidt orthogonalization procedure

(1) To get $\phi_1(t)$, just compute

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

where $E_1 = \int_{-\infty}^{\infty} s_1(t)s_1(t)dt$.

(2) To get $\phi_2(t)$, compute

$$\phi_2(t) = \frac{\theta_2(t)}{\sqrt{E_{\theta 2}}}$$

where $\theta_2(t) = s_2(t) - s_{21}\phi_1(t)$ and $E_{\theta 2} = \int_{-\infty}^{\infty} \theta_2(t)\theta_2(t)dt$.

4.3 Modulator Signal

- Gram-Schmidt orthogonalization procedure

(3) To get $\phi_3(t)$, just compute

$$\phi_3(t) = \frac{\theta_3(t)}{\sqrt{E_{\theta 3}}}$$

where $\theta_3(t) = s_3(t) - s_{31}\phi_1(t) - s_{32}\phi_2(t)$

and $E_{\theta 3} = \int_{-\infty}^{\infty} \theta_3(t)\theta_3(t)dt$.

(4) Keep going, up to $\phi_M(t)$, and if you get $\phi_k(t) = 0$

along the way, just throw that one out, because you don't need it.

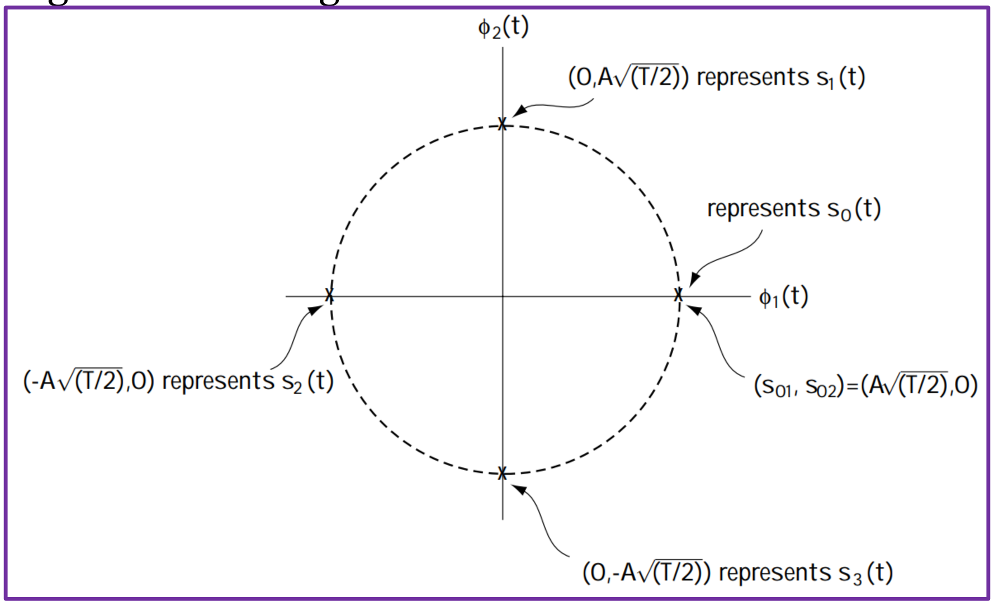
4.3 Modulator Signal

- Representing Modulated Signals : 4 - PSK

	Input bits	Output waveforms				
4-PSK	00	$s_0(t) = A\cos(\omega_c t + 0°)\pi(t-iT)$	=	$A\cos(\omega_c t)\cdot\pi(t-iT)$	+	0
	01	$s_1(t) = A\cos(\omega_c t + 90°)\cdot\pi(t-iT)$	=	0		$-A\sin(\omega_c t)\cdot\pi(t-iT)$
	10	$s_2(t) = A\cos(\omega_c t + 180°)\cdot\pi(t-iT)$	=	$-A\cos(\omega_c t)\cdot\pi(t-iT)$	+	0
	11	$s_3(t) = A\cos(\omega_c t + 270°)\cdot\pi(t-iT)$	=	0		$+A\sin(\omega_c t)\cdot\pi(t-iT)$

4.3 Modulator Signal

- Representing Modulated Signals : 4 - PSK



4.3 Modulator Signal

- Representing Modulated Signals : 8 - PSK

	Input bits	Output waveforms				
8-PSK	000	$s_0(t) = A\cos(\omega_c t + 0^\circ)\cdot\pi(t-iT)$	=	$A\cos(\omega_c t)\cdot\pi(t-iT)$	+	0
	001	$s_1(t) = A\cos(\omega_c t + 45^\circ)\cdot\pi(t-iT)$	=	$\frac{A}{\sqrt{2}}\cos(\omega_c t)\cdot\pi(t-iT)$	$-$	$\frac{A}{\sqrt{2}}\sin(\omega_c t)\cdot\pi(t-iT)$
	010	$s_2(t) = A\cos(\omega_c t + 90^\circ)\cdot\pi(t-iT)$	=	0		$-A\sin(\omega_c t)\cdot\pi(t-iT)$
	011	$s_3(t) = A\cos(\omega_c t + 135^\circ)\cdot\pi(t-iT)$	=	$\frac{-A}{\sqrt{2}}\cos(\omega_c t)\cdot\pi(t-iT)$	$-$	$\frac{A}{\sqrt{2}}\sin(\omega_c t)\cdot\pi(t-iT)$

4.3 Modulator Signal

- Representing Modulated Signals : 8 - PSK

100	$s_4(t) = A\cos(\omega_c t+180°)\cdot\pi(t-iT)$	$=$	$-A\cos(\omega_c t)\cdot\pi(t-iT)$	$+\ 0$
101	$s_5(t) = A\cos(\omega_c t+225°)\cdot\pi(t-iT)$	$=$	$\frac{-A}{\sqrt{2}}\cos(\omega_c t)\cdot\pi(t-iT)$	$+\ \frac{A}{\sqrt{2}}\sin(\omega_c t)\cdot\pi(t-iT)$
110	$s_6(t) = A\cos(\omega_c t+270°)\cdot\pi(t-iT)$	$=$	0	$+\ A\sin(\omega_c t)\cdot\pi(t-iT)$
111	$s_7(t) = A\cos(\omega_c t+315°)\cdot\pi(t-iT)$	$=$	$\frac{A}{\sqrt{2}}\cos(\omega_c t)\cdot\pi(t-iT)$	$+\ \frac{A}{\sqrt{2}}\sin(\omega_c t)\cdot\pi(t-iT)$

4.3 Modulator Signal

- Representing Modulated Signals : ASK

	Output waveform	Output waveform represented on orthonormal basis
BASK	$s_0(t)$	$\mathbf{s}_0 = s_{01} = -A\sqrt{½}$
	$s_1(t)$	$\mathbf{s}_1 = s_{11} = A\sqrt{½}$
4-ASK	$s_0(t)$	$\mathbf{s}_0 = s_{01} = -3A\sqrt{½}$
	$s_1(t)$	$\mathbf{s}_1 = s_{11} = -A\sqrt{½}$
	$s_2(t)$	$\mathbf{s}_2 = s_{21} = A\sqrt{½}$
	$s_3(t)$	$\mathbf{s}_3 = s_{31} = 3A\sqrt{½}$

4.3 Modulator Signal

- Representing Modulated Signals : ASK

8-ASK	$s_0(t)$	$\mathbf{s}_0 = s_{01} = -7A\sqrt{T/2}$
	$s_1(t)$	$\mathbf{s}_1 = s_{11} = -5A\sqrt{T/2}$
	$s_2(t)$	$\mathbf{s}_2 = s_{21} = -3A\sqrt{T/2}$
	$s_3(t)$	$\mathbf{s}_3 = s_{31} = -A\sqrt{T/2}$
	$s_4(t)$	$\mathbf{s}_4 = s_{41} = A\sqrt{T/2}$
	$s_5(t)$	$\mathbf{s}_5 = s_{51} = 3A\sqrt{T/2}$
	$s_6(t)$	$\mathbf{s}_6 = s_{61} = 5A\sqrt{T/2}$
	$s_7(t)$	$\mathbf{s}_7 = s_{71} = 7A\sqrt{T/2}$

4.3 Modulator Signal

- Representing Modulated Signals : QAM
- The information bits are stuffed into both the phase (θ) and the amplitude (A) of the cosine waveform. That is, a typical output waveform for QAM looks like this

$$s_j(t) = A_j \cos(\omega_c t + \theta_j) \cdot \pi(t - iT)$$

- we can rewrite this as

$$s_j(t) = A_j \cos(\theta_j)\cos(\omega_c t) \cdot \pi(t - iT) - A_j \sin(\theta_j)\sin(\omega_c t) \cdot \pi(t - iT)$$

4.3 Modulator Signal

- Representing Modulated Signals : QAM
- Using the orthonormal basis as

$$s_j(t) = s_{j1}\phi_1(t) + s_{j2}\phi_2(t)$$

where

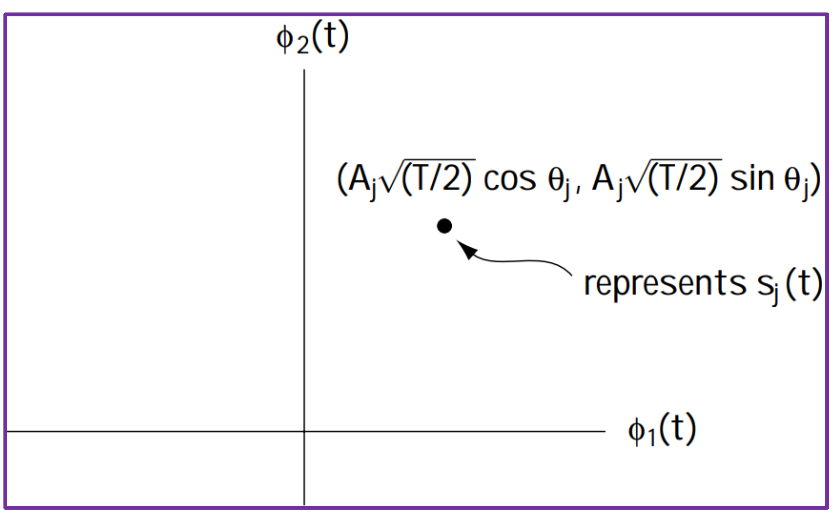
$$\phi_1(t) = +\sqrt{2/T}\cos(\omega_c t)\cdot\pi(t-iT)$$

$$\phi_2(t) = -\sqrt{2/T}\sin(\omega_c t)\cdot\pi(t-iT)$$

4.3 Modulator Signal

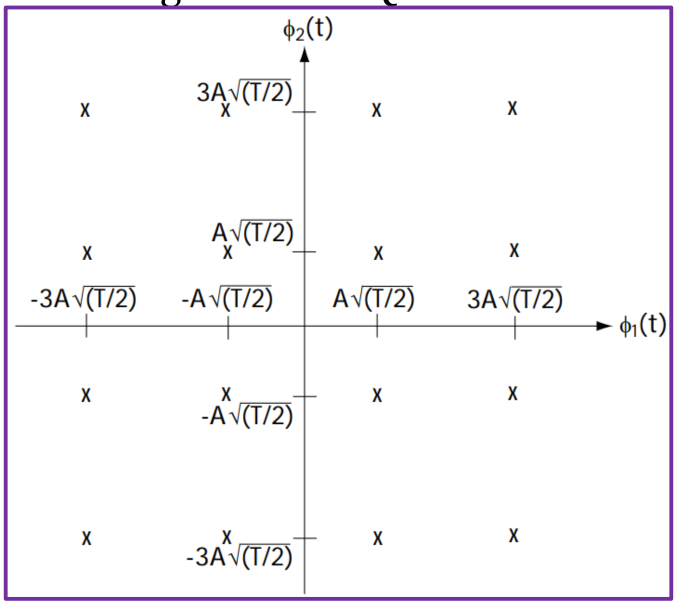
- Representing Modulated Signals : QAM

$$s_j(t) \leftrightarrow \underline{s}_j = (s_{j1}, s_{j2}) = \left(A_j\sqrt{T/2}\cos\theta_j, A_j\sqrt{T/2}\sin\theta_j\right)$$



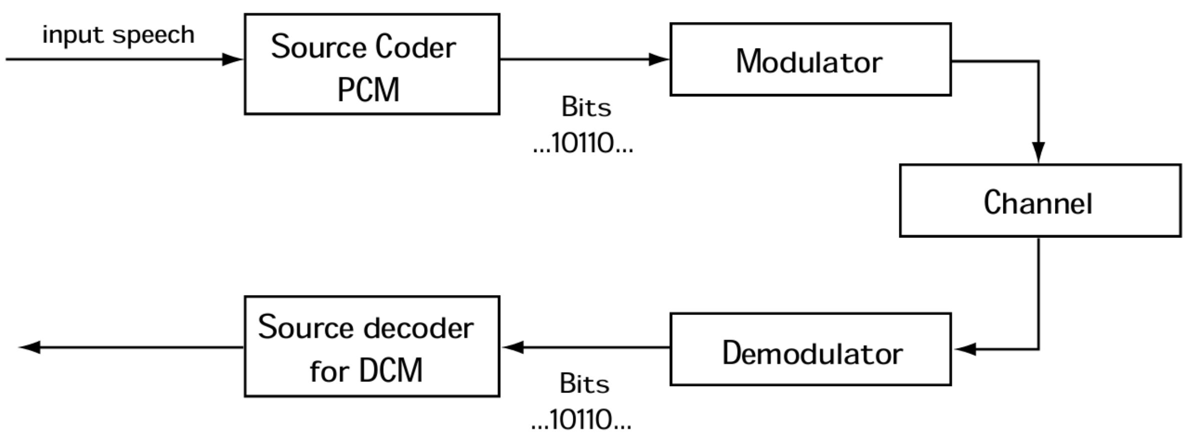
4.3 Modulator Signal

- Representing Modulated Signals : 16 - QAM

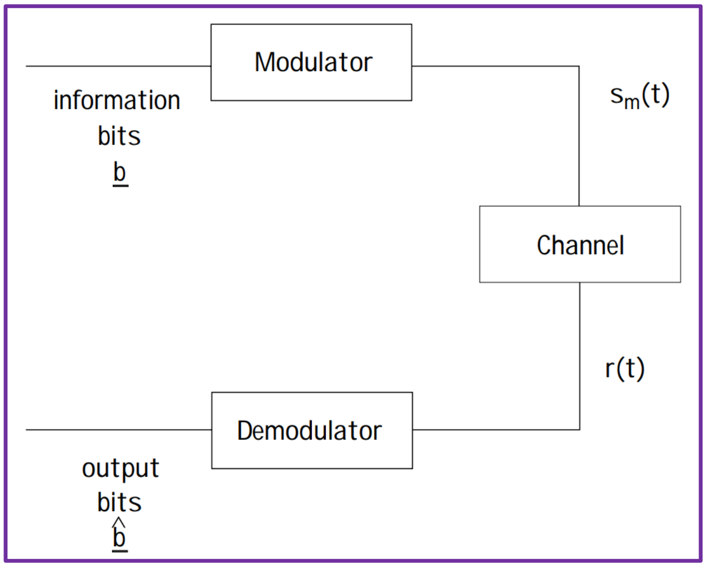


4.4 Demodulator

- The demodulator is a device that gets the signal sent across the channel and turns it back into bits.

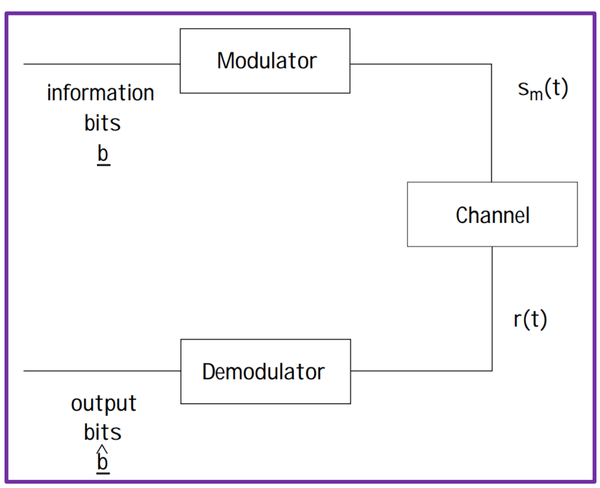


4.4 Demodulator



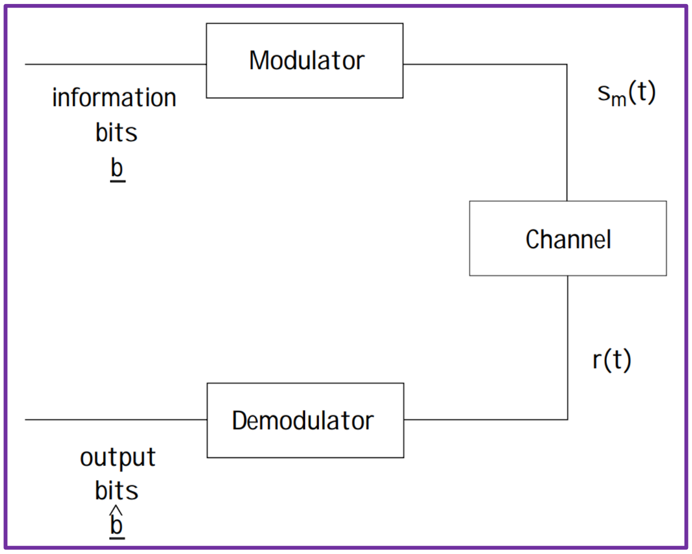
- The key to building a good demodulator is to minimize the effects of noise and give the highest probability of guessing the correct sent signal.
- This will make sure that the bits that leave the demodulator (receiver side) are as close as possible to the bits that come into the modulator (transmitter side).

4.4 Demodulator



- $r(t) = s_m(t) + \eta(t)$
- $r(t) = r_1\phi_1(t) + r_2\phi_2(t) + \cdots + r_N\phi_N(t)$

4.4 Demodulator



$$r_1 = \int r(t)\varphi_1(t)dt$$

$$r_1 = \int (s_m(t)+\eta(t))\varphi_1(t)dt$$

$$r_1 = \int s_m(t)\varphi_1(t)dt + \int \eta(t)\varphi_1(t)dt$$

$$r_1 = s_{m1} + \eta_1$$

4.4 Demodulator

$$r_2 = s_{m2} + \eta_2$$

$$r_3 = \int r(t)\varphi_3(t)dt$$

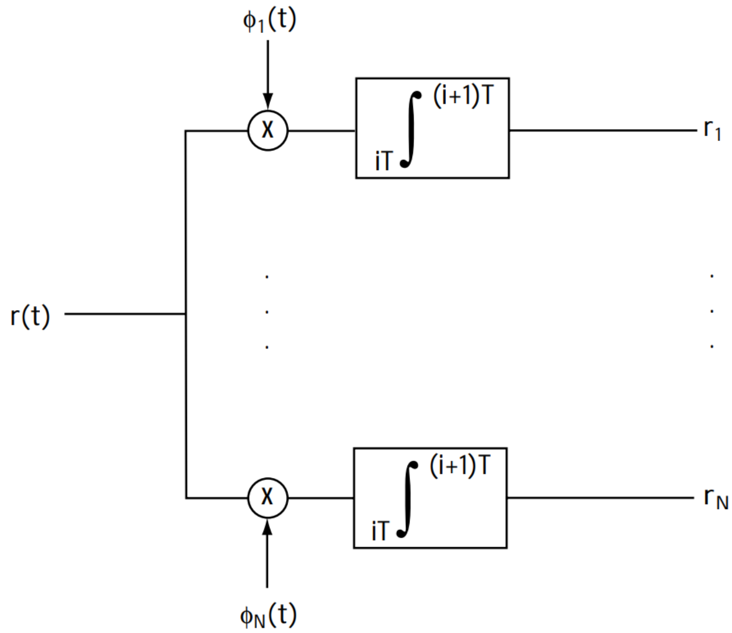
$$r_3 = \int (s_m(t)+\eta(t))\varphi_3(t)dt$$

$$r_3 = \int s_m(t)\varphi_3(t)dt + \int \eta(t)\varphi_3(t)dt$$

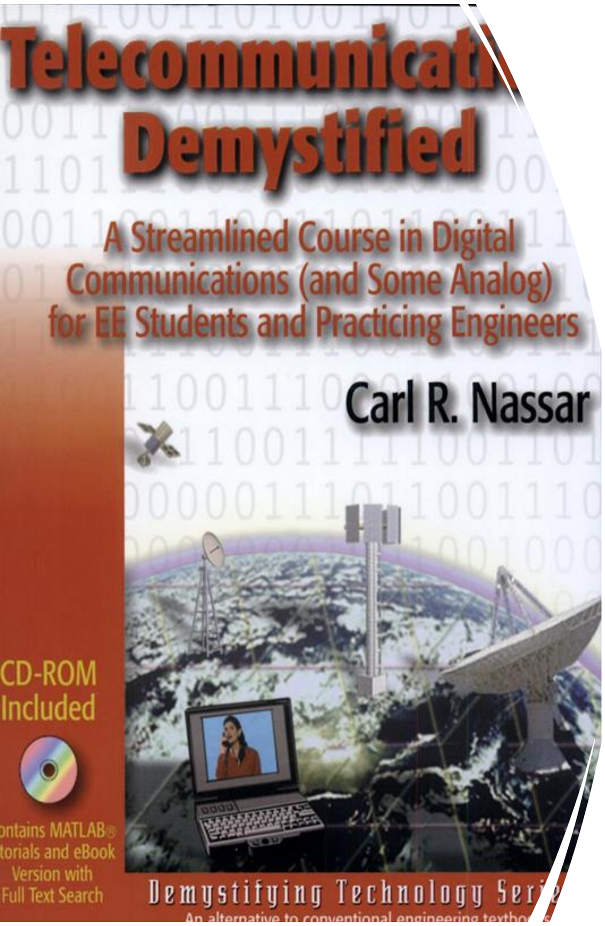
$$r_3 = \int (s_{m1}\varphi_1(t)+s_{m2}\varphi_2(t))\varphi_3(t)dt + \int \eta(t)\varphi_3(t)dt$$

$$r_3 = s_{m1}\int \varphi_1(t)\varphi_3(t)dt + s_{m2}\int \varphi_2(t)\varphi_3(t)dt + \int \eta(t)\varphi_3(t)dt$$

4.4 Demodulator

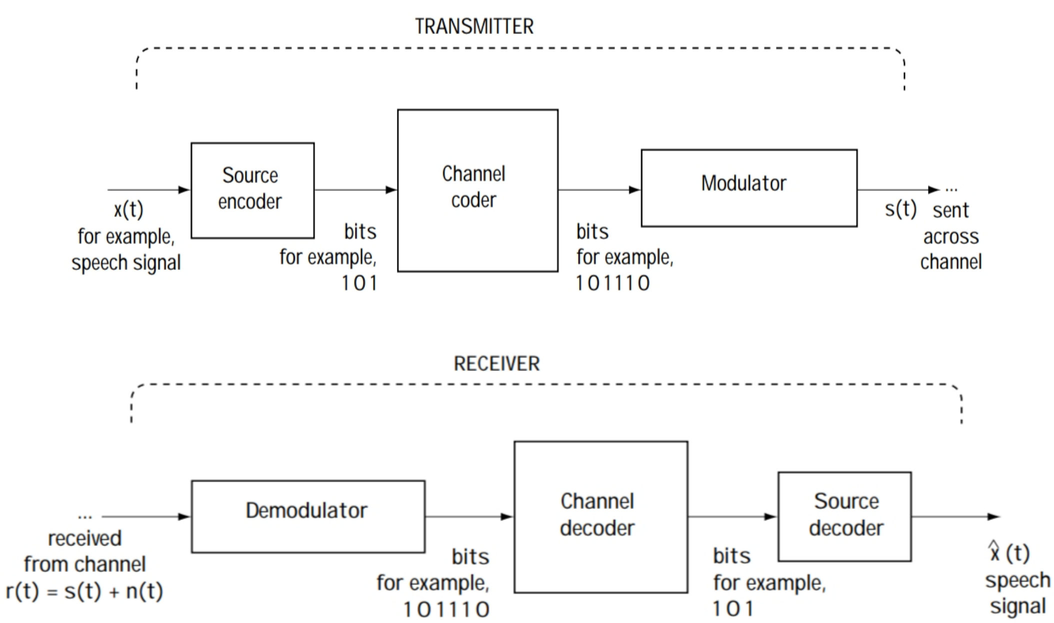


Correlator receiver front end



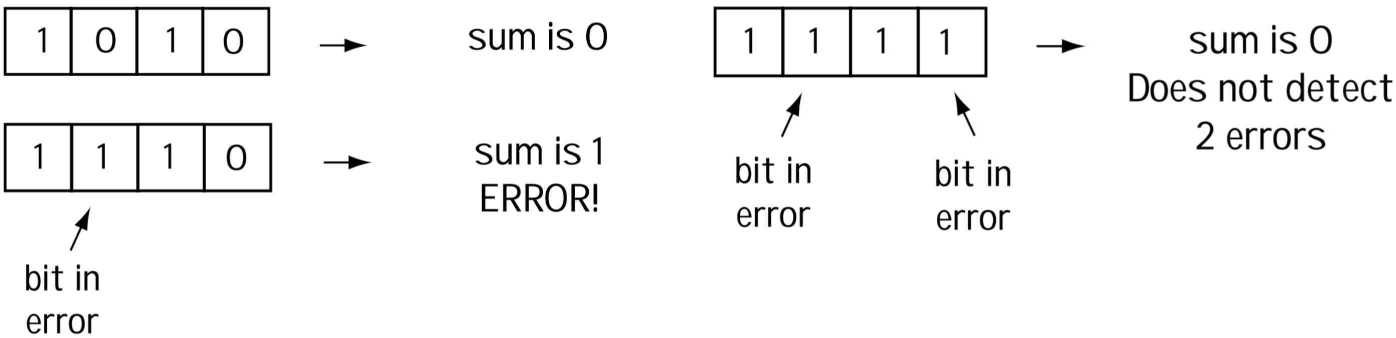
5. Block Coding and Decoding

5. Block Coding and Decoding



5.1 Simple Block Coding

- The Single Parity Check Bit Coder - Channel coding where one bit is added to create a total sum of 0 is called *even parity*. You can instead add one more bit so that the total when adding all bits is 1, and this is called *odd parity*.

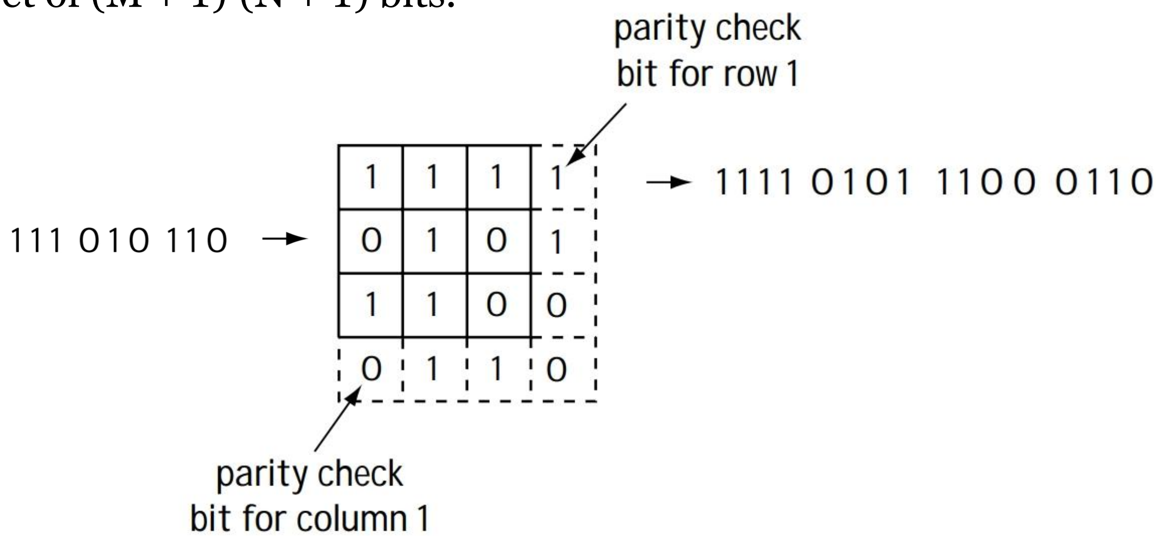


5.1 Simple Block Coding

- Consider a block code that maps each incoming set of k bits to an outgoing set of n bits:
  1. First, in shorthand notation, this block code will be called an (n,k) code.
  2. This block code is said to have a code rate of k/n.
  3. This block code is also said to have a redundancy of (n–k)/k.
  4. And finally, this code is said to have (n–k) redundant bits (that is, check bits or parity bits), which refer to the added (n–k) bits.

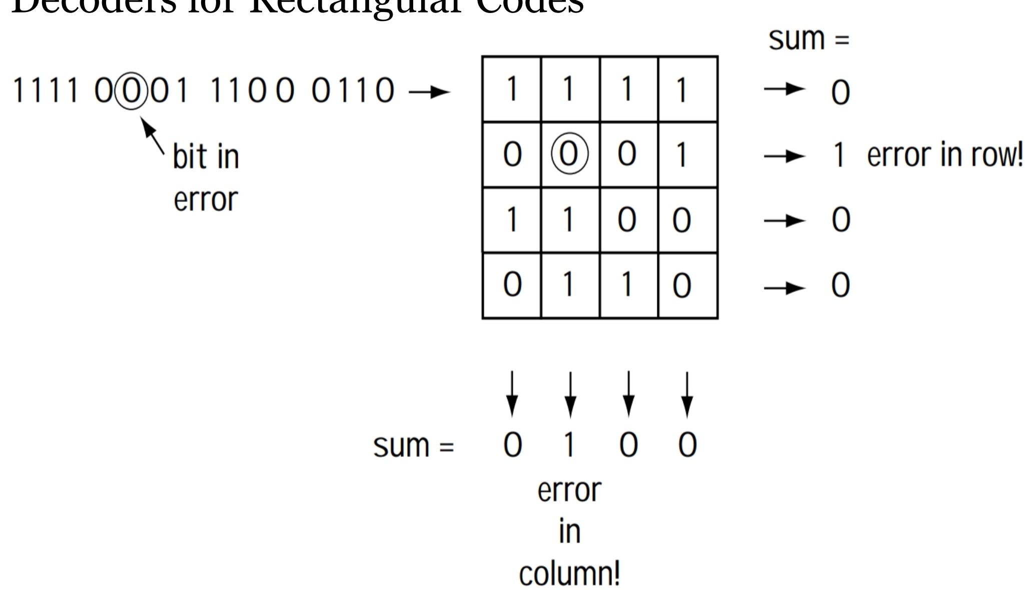
5.1 Simple Block Coding

- Rectangular Codes - In rectangular codes, each set of M·N bits are mapped to a set of (M + 1)·(N + 1) bits.



5.1 Simple Block Coding

- Channel Decoders for Rectangular Codes



5.2 Linear block codes

- Linear block coders are a group of block coders that follow a special set of rules when choosing which set of outputs to use. The rules are as follows, using a (6,3) code for illustrative purposes:

  V_n = the set of all possible 64 6-bit sequences

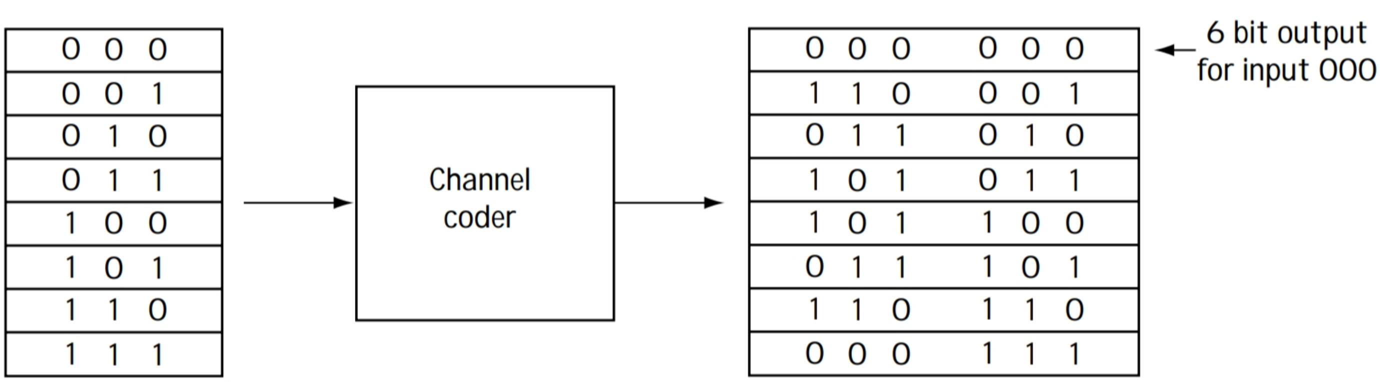
  U = the set of eight 6-bit sequences output at the channel coder
- Using this notation, the rule is this:

  U must be a subspace of V_n.
- This means two very simple things:

  1. U must contain {000000}

  2. Adding (modulo 2) any two elements in U must create another element in U.

5.2 Linear block codes



5.2 Linear block codes

input bits	output bits
0 0	0 0 0 0
0 1	0 1 0 1
1 0	1 0 1 0
1 1	1 1 1 1

- A linear block code?

5.2 Linear block codes

1 1 0

Channel coder : uses look-up table

Input	Output
0 0 0	0 0 0 0 0 0
0 0 1	1 1 0 0 0 1
0 1 0	0 1 1 0 1 0
0 1 1	1 0 1 0 1 1
1 0 0	1 0 1 1 0 0
1 0 1	0 1 1 1 0 1
1 1 0	1 1 0 1 1 0
1 1 1	0 0 0 1 1 1

1 1 0 1 1 0

- vector **m** (1 0 1)
- vector **u** = (0 1 1 1 0 1)
- **u** = **m** G

5.2 Linear block codes

Channel coder : uses look-up table

110 → 110110

Input	Output
0 0 0	0 0 0 0 0 0
0 0 1	1 1 0 0 0 1
0 1 0	0 1 1 0 1 0
0 1 1	1 0 1 0 1 1
1 0 0	1 0 1 1 0 0
1 0 1	0 1 1 1 0 1
1 1 0	1 1 0 1 1 0
1 1 1	0 0 0 1 1 1

$$G = \left(P_{3\times 3} \vdots I_{3\times 3}\right)$$

$$= \begin{pmatrix} P_{11} & P_{12} & P_{13} & \vdots & 1 & 0 & 0 \\ P_{21} & P_{22} & P_{23} & \vdots & 0 & 1 & 0 \\ P_{31} & P_{32} & P_{33} & \vdots & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{u} = \mathbf{m}G$$

$$= \begin{pmatrix} 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 1 \end{pmatrix}$$

5.2 Linear block codes

input bits	output bits
0 0	0 0 0 0
0 1	0 1 0 1
1 0	1 0 1 0
1 1	1 1 1 1

$$G = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}$$

$$\mathbf{u}_{\mathrm{m}=(0\ \ 0)} = \mathbf{m}G = \begin{pmatrix} 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{u}_{\mathrm{m}=(0\ \ 1)} = \mathbf{m}G =$$

$$\mathbf{u}_{\mathrm{m}=(1\ \ 0)} = \mathbf{m}G =$$

$$\mathbf{u}_{\mathrm{m}=(1\ \ 1)} = \mathbf{m}G =$$

5.2 Linear block codes

- The Decoding

$$G\,H = \mathbf{0}$$

$$G = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \qquad H = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \hline 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$$

5.2 Linear block codes

- The Decoding

$$\mathbf{v}\,H = \mathbf{u}\,H = \mathbf{m}\,G\,H = \mathbf{m}\,\mathbf{0} = \mathbf{0}$$

$$\begin{aligned} \mathbf{v} &= \mathbf{u} + \mathbf{e} \\ \mathbf{v}\,H &= \mathbf{u}\,H + \mathbf{e}\,H \\ &= \mathbf{m}\,G\,H + \mathbf{e}\,H \\ &= \mathbf{m}\,\mathbf{0} + \mathbf{e}\,H \\ &= \mathbf{0} + \mathbf{e}\,H \\ &= \mathbf{e}\,H \end{aligned}$$

5.2 Linear block codes

- Mapping errors e to syndromes **S**

e	**S** = **e** H
0 0 0 0 0 0	0 0 0
0 0 0 0 0 1	1 0 1
0 0 0 0 1 0	0 1 1
0 0 0 1 0 0	1 1 0
0 0 1 0 0 0	0 0 1
0 1 0 0 0 0	0 1 0
1 0 0 0 0 0	1 0 0
0 1 0 0 0 1	1 1 1

5.3 Performance of the Block Coders

- Performances of single parity check bit coders/decoders

$$P_m = \sum_{\substack{j=2 \\ j \in even}}^{n} P(j,n) \qquad P(j,n) = \binom{n}{j} p^j (1-p)^{n-j}$$

- Performance of rectangular codes

$$P = \sum_{j=2}^{n} P(j,n)$$

- Performance of linear block codes

$$t = \left\lfloor \frac{d_{\min} - 1}{2} \right\rfloor \qquad P = \sum_{j=t+1}^{n} P(j,n)$$

5.3 Performance of the Block Coders

- Benefits and Costs of Block Coders