EE2-4 Communication Systems

Dr. Deniz Gündüz

Reader in Information Theory and Communications

d.gunduz@imperial.ac.uk

www.imperial.ac.uk/ipc-lab

Contents

Chapter 1

Introduction

1.1 Background

Communication involves the transfer of information from one point to another (in time or space). In general, a communication system can be represented by the model shown in Fig. 1.1.

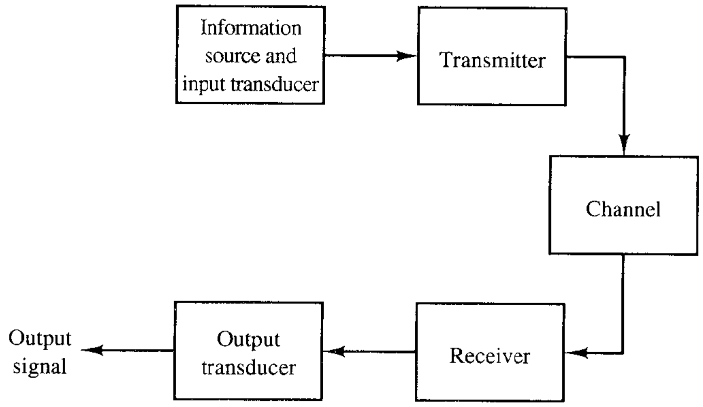


Figure 1.1: Block diagram of a communication system. [Proakis & Salehi, Fig. 1.1]

The information generated by the source (which might be text, voice, image, etc.) is converted into an electrical signal. The transmitter then converts (through the process of modulation) this signal into a form that is suitable for transmission. The communication channel is the physical medium that is used to send the signal from transmitter to receiver. Finally, the function of the receiver is to recover the message signal, and a transducer then converts it into a form that is suitable for the user.

From the point of view of this course, the most important aspect is that in its transmission from source to user, the message signal is corrupted by noise. Although we may know basically how communication systems work, and understand different forms of modulation, we also need to consider how these systems behave when subjected to noise. The basic question that motivates most of the material in this course is: *How do communication systems perform in the presence of noise and how is noise to be coped with?*

1.2 Some Definitions

Signal

A *signal* is a single-valued function of time that conveys information. In other words, at every point in time there is a unique value of the function. This value may either be a real number, giving a real-valued signal, or a complex number, giving a complex-valued signal.

Deterministic and Random Signals

A *deterministic* signal can be modelled as a completely specified function of time. In other words, there is no uncertainty about its value at any time. For example, the sinusoid signal $A\cos(2\pi f_c t + \theta)$ is deterministic if A, f_c and θ are known constants.

A *random* (or stochastic) signal cannot be completely specified as a function of time and must be modelled probabilistically. Random signals will be extremely important in this course, as we will use them to model noise.

Analog and Digital Signals

An *analog* signal is a continuous function of time, for which the amplitude is also continuous. Analog signals arise whenever a physical waveform (e.g., a speech wave) is converted to an electrical signal.

A *digital* signal is a discrete function of time, for which the amplitude can only have a finite set of values. Sometimes a distinction is also made of discrete-time signals—these are signals that are a discrete function of time, but the amplitude may take on a continuum of values. In this course we will be primarily concerned with analog signals in Chapter 3 and digital signals in Chapter 4.

Power

The *instantaneous power* of a voltage or current signal is given by

$$p = v(t)i(t)$$

Then with the help of Ohm's law

$$i(t) = \frac{v(t)}{R}$$

where R is the resistance, we get

$$p = \frac{|v(t)|^2}{R} \quad \text{or} \quad p = |i(t)|^2 R \tag{1.1}$$

The convention is to normalise the power using a 1Ω resistor, so

$$p = |g(t)|^2 \tag{1.2}$$

where $g(t)$ is either a voltage or a current signal.

For a periodic signal, the *average power* is defined as

$$P \equiv \frac{1}{T}\int_{-T/2}^{T/2} |g(t)|^2 \, dt \tag{1.3}$$

where T is the period of the signal. For a non-periodic signal, the *average power* is defined as

$$P \equiv \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} |g(t)|^2 \, dt \tag{1.4}$$

Example 1.1 – *Average power of a sinusoidal signal*

Consider the deterministic signal $x(t) = A\cos(2\pi f t + \theta)$, where A is the amplitude, f is the frequency, and θ is the phase. By definition, the average power is

$$\begin{aligned} P &= \frac{1}{T} \int_{-T/2}^{T/2} A^2 \cos^2(2\pi f t + \theta) \, dt \\ &= \frac{1}{T} \int_{-T/2}^{T/2} \frac{A^2}{2} \left[1 + \cos(4\pi f t + 2\theta)\right] \, dt \\ &= \frac{A^2 T}{2T} + \frac{A^2}{8\pi f T} \Big[\sin(4\pi f t + 2\theta) \Big]_{-T/2}^{T/2} \\ &= \frac{A^2}{2} \end{aligned} \tag{1.5}$$

Note that we have used the identity $\cos^2 x = \frac{1}{2}(1 + \cos 2x)$ in the second line above.

□

Energy

The signal energy is defined as

$$E = \int_{-\infty}^{\infty} |g(t)|^2 \, dt. \tag{1.6}$$

Bandwidth

The *bandwidth* of a signal provides a measure of the extent of significant spectral content of the signal for *positive* frequencies. When the signal is strictly band limited the bandwidth is well defined. However, the meaning of "significant" is mathematically imprecise when the signal is not strictly band limited. Several engineering definitions of bandwidth are commonly in use, including:

Null-to-null bandwidth: range of frequencies between zeros in the magnitude spectrum.

3-dB bandwidth: range of frequencies where the magnitude spectrum falls no lower than $1/\sqrt{2}$ of the maximum value of the magnitude spectrum.

Equivalent noise bandwidth: width of a fictitious rectangular spectrum such that the power in the rectangular band is equal to the power associated with the actual spectrum over positive frequencies.

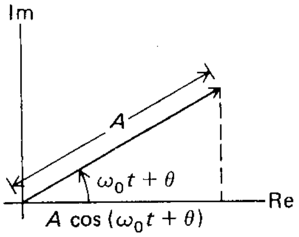


Figure 1.2: Projection of a rotating phasor onto the real axis. [Ziemer & Tranter, Fig. 2.2]

Phasors

To develop the notion of the frequency content of signals, we will consider a phasor representation. Phasors are useful in circuit analysis for representing sinusoidal signals. For example, consider the general sinusoid

$$x(t) = A\cos(2\pi f_0 t + \theta) \tag{1.7}$$

You have probably already come across the phasor representation

$$x(t) = \mathfrak{Re}\left\{Ae^{j\theta}\, e^{j2\pi f_0 t}\right\} \tag{1.8}$$

where the term in brackets is viewed as a rotating vector in the complex plane, as shown in Fig. 1.2. The phasor has length A, rotates anti-clockwise at a rate of f_0 revolutions per second, and at time $t = 0$ makes an angle of θ with respect to the positive real axis. The waveform $x(t)$ can then be viewed as a projection of this vector onto the real axis.

References

- Lathi, chapter 1, sections 2.1, 2.2.
- Couch, sections 1-2, 1-3, 2-1, and 2-9.

Chapter 2

Noise

2.1 Background

The performance of any communication system, such as that shown in Fig. 1.1, is ultimately limited by two factors: (i) the transmission bandwidth, and (ii) the noise. Bandwidth is a resource that must be conserved as much as possible, since only a finite electromagnetic spectrum is allocated for any transmission medium.[a] Whatever the physical medium of the channel, the transmitted signal is corrupted in a random manner by a variety of possible mechanisms as it propagates though the system. The term *noise* is used to denote the unwanted waves that disturb the transmission of signals, and over which we have incomplete control. As we will see throughout this course, bandwidth and noise are intimately linked. In this chapter our aim is to develop a model for the noise, with the ultimate goal of using this model in later chapters to assess the performance of various modulation schemes when used in the presence of noise.

2.2 A Model of Noise

2.2.1 Sources of noise

In a practical communication system, there are many sources of noise. These source may be external to the system (e.g., atmospheric,[b] galactic,[c] and synthetic noise[d]) or internal to the system. Internal noise arises due to spontaneous fluctuation of current or voltage in electrical circuits, and consists of both *shot noise* and *thermal noise*.

Shot noise arises in electronic devices and is caused by the random arrival of electrons at the output of semiconductor devices. Because the electrons are discrete and are not moving in a continuous steady flow, the current is randomly fluctuating. The important characteristic of shot noise is that it is *Gaussian* distributed with zero mean (i.e, it has the Gaussian probability density

[a] Spread-spectrum schemes, such as code-division multiple access (CDMA), actually use a transmission bandwidth that is far greater than necessary (and is independent of the bandwidth of the message signal). However, this is done primarily as a means of reducing the deleterious effect of noise, specifically noise caused by multipath propagation in mobile communications.

[b] Atmospheric noise is naturally occurring electrical disturbances that originate within the Earth's atmosphere, caused by conditions such as lightning.

[c] Galactic noise originates from outside the Earth's atmosphere, and includes solar noise and cosmic noise (background radiation in the universe).

[d] The predominant sources of synthetic noise are spark-producing mechanisms, as well as RF interference.

function shown in Fig. 2.2). This follows from the *central limit theorem*, which states that the sum of n independent random variables approaches a Gaussian distribution as $n \to \infty$. In practice, engineers and statisticians usually accept that the theorem holds when $n \gtrsim 6$.

Thermal noise is associated with the rapid and random motion of electrons within a conductor due to thermal agitation. It is also referred to as Johnson noise, since it was first studied experimentally in 1928 by Johnson,[e] who showed that the average power in a conductor due to thermal noise is

$$P_{\text{thermal}} = kTB \tag{2.1}$$

where k is Boltzman's constant (1.38×10^{-23}), T is the absolute temperature in Kelvin, and B is the bandwidth in hertz.[f] Again, because the number of electrons in a conductor is very large, and their random motions are statistically independent, the central limit theorem indicates that thermal noise is Gaussian distributed with zero mean.

The noise power from a source (not necessarily a thermal source) can be specified by a number called the *effective noise temperature*:

$$T_e = \frac{P}{kB} \tag{2.2}$$

Effective noise temperature can be interpreted as the temperature of a fictitious thermal noise source at the *input*, that would be required to produce the same noise power at the *output*. Note that if the noise source is not thermal noise, then T_e may have nothing to do with the physical temperature of the device.

The important thing to note from this section is that *noise is inevitable*.

2.2.2 The additive noise channel

The simplest model for the effect of noise in a communication system is the additive noise channel, shown in Fig. 2.1. Using this model the transmitted signal $s(t)$ is corrupted by the addition of a random noise signal $n(t)$. If this noise is introduced primarily by electronic components and amplifiers at the receiver, then we have seen that it can be characterised statistically as a Gaussian process. It turns out that the noise introduced in most physical channels is (at least approximately) Gaussian, and thus, this simple model is the predominant one used in communication system analysis and design.

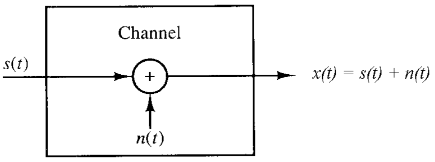


Figure 2.1: The additive noise channel. [Proakis & Salehi, Fig. 1.8]

[e] J.B. Johnson, "Thermal agitation of electricity in conductors", *Physical Review*, vol. 32, pp.97-109, July 1928.

[f] This equation is actually an approximation, although it is valid for frequencies up to about 100 GHz.

2.3 A Statistical Description of Noise

As we have already hinted at, noise is completely random in nature. The noise signal $n(t)$ is a time-varying waveform, however, and just like any other signal it must be affected by the system through which it passes. We therefore need a model for the noise that allows us to answer questions such as: How does one quantitatively determine the effect of systems on noise? What happens when noise is picked up at the receiver and passed through a demodulator? And what effect does this have on the original message signal? Here we seek a representation that will enable us to answer such questions in the following chapters.

2.3.1 Background on Probability

Before developing a mathematical model for the noise, we need to define a few terms.

Random Variable

Consider a *random experiment*, that is, an experiment whose outcome cannot be predicted precisely. The collection of all possible separately identifiable outcomes of a random experiment is called the *sample space*, $\mathcal{S}$. An *event* is a collection of possible outcomes of the random experiment. So, the sample space is the event that contains all possible outcomes of the random experiment.

A *random variable* is a rule or relationship (denoted by x) that assigns a real number x_i to the ith sample point in the sample space. In other words, the random variable x can take on values $x_i \in \mathcal{S}$. So, a random variable is the value we assign to the outcome of a random experiment. Random variables are described in terms of their distribution functions, which are defined in terms of the probability of an event to happen. The *probability* of the random variable x taking on the value x_i is denoted $P_{\mathrm{x}}(x_i)$.

The probability of an event to happen is a non-negative number, with the following properties:

- The probability of the event that includes all possible outcomes of the experiment is 1.
- The probability of two events that do not have any common outcome is the sum of the probabilities of the two events separately.

Distribution (Cumulative) and Probability Density Functions

The *distribution* (or *cumulative*) *function* of a random variable f is defined as the probability of the variable taking a value less than the argument of the distribution function:

$$\underbrace{F_f(z)}_{\substack{Distribution\\ function}} \equiv \underbrace{P_f}_{Probability} (\underbrace{f}_{\substack{random\\ variable}} \leq \underbrace{z}_{a\ number.}) \tag{2.3}$$

Obviously:

$$F_f(-\infty) = 0 \qquad F_f(\infty) = 1 \tag{2.4}$$

If $z_1 \leq z_2 \Rightarrow F_f(z_1 \leq F_f(z_2)$.

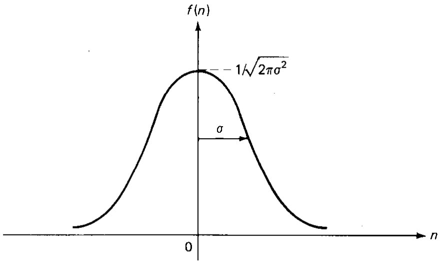


Figure 2.2: Gaussian probability density function. [Schwartz, Fig. 5-2]

The *probability density function* (pdf) of a random variable f is defined as the derivative of the distribution function:

$$p_f(f) \equiv \frac{d}{dz} F_f(z). \tag{2.5}$$

Note that we use upper case P to denote probability, and lower case p to denote a pdf.

We also have that

$$P(z_1 < f \leq z_2) = P_f(f \leq z_2) - P_f(f \leq z_1) = \int_{z_1}^{z_2} p_f(z)\, dz. \tag{2.6}$$

One specific pdf that we will be particularly interested in is the *Gaussian* pdf, defined as

$$p_{\mathbf{x}}(x) = \frac{1}{\sigma\sqrt{2\pi}}\, e^{-(x-m)^2/(2\sigma^2)}, \tag{2.7}$$

where m is the *mean* of the pdf and σ^2 is the *variance*. This pdf is shown in Fig. 2.2.

Statistical Averages

One is often interested in calculating averages of a random variable. Denote the *expected value* (or mean value) of a random variable f as $E\{f\}$. If the random variable f has a pdf $p_f(z)$, then the expected value is defined as

$$E\{f\} \equiv \mu_f \equiv \int_{-\infty}^{\infty} z p_f(z)\, dz, \tag{2.8}$$

where $E\{\cdot\}$ denotes the *expectation operator*.

It is also often necessary to find the mean value of a function of a random variable, for example, the mean square amplitude of a random signal. Suppose we want to find $E\{y\}$ where y is a random variable given by

$$y \equiv g(f),$$

where f is a random variable with a known pdf, and $g(\cdot)$ is an arbitrary function. Then,

$$E\{y\} = E\{g(f)\} = \int_{-\infty}^{\infty} g(z) p_f(z)\, dz. \tag{2.9}$$

The *variance* of a random variable f is defined as

$$\sigma_f^2 \equiv E\{(f-\mu_f)^2\} = \int_{-\infty}^{\infty} (z-\mu_f)^2 p_f(z)\, dz \tag{2.10}$$

The following relationship holds:

$$\begin{aligned} \sigma_f^2 &= E\left\{f^2 + \mu_f^2 - 2f\mu_f\right\} \\ &= E\left\{f^2\right\} + E\left\{\mu_f^2\right\} - 2E\{f\}\mu_f \\ &= E\left\{f^2\right\} + (\mu_f)^2 - 2\,(\mu_f)^2 \\ &= E\{f^2\} - \mu_f^2 \end{aligned} \tag{2.11}$$

Note that for a zero-mean random variable, the variance is equal to the mean square.

2.3.2 n random variables

The joint distribution function of n random variables is

$$F_{f_1 f_2 \ldots f_n}(z_1, z_2, \ldots, z_n) \equiv P(f_1 \le z_1, f_2 \le z_2, \ldots, f_n \le z_n) \tag{2.12}$$

Their joint probability density function is:

$$p_{f_1 f_2 \ldots f_n}(z_1, z_2, \ldots, z_n) \equiv \frac{\partial^n F_{f_1 f_2 \ldots f_n}(z_1, z_2, \ldots, z_n)}{\partial z_1 \partial z_2 \ldots \partial z_n} \tag{2.13}$$

For independent n random variables we have:

$$F_{f_1 f_2 \ldots f_n}(z_1, z_2, \ldots, z_n) = F_{f_1}(z_1) F_{f_2}(z_2) \ldots F_{f_n}(z_n) \tag{2.14}$$

For uncorrelated n random variables we have:

$$E\{f_i f_j\} = E\{f_i\} E\{f_j\}, \forall i, j, i \neq j \tag{2.15}$$

Two random variables are orthogonal when

$$E\{f_i f_j\} = 0 \tag{2.16}$$

Element c_{ij} of the covariance matrix of two random variables is defined as

$$c_{ij} \equiv E\left\{(f_i - \mu_{f_i})(f_j - \mu_{f_j})\right\} \tag{2.17}$$

2.3.3 Random Processes

A random process is a time-varying function that assigns the outcome of a random experiment to each time instant: $f(t; s_i)$.

For fixed s_i (outcome, or sample of the random experiment) it is a time varying function, eg a signal.

If s_i scans all possible outcomes of the underlying random experiment, we shall get a series of signals. These signals will constitute an ensemble.

For a given outcome, (fixed s_i), the random process gives us the value of the signal. For fixed t, the random process becomes a random variable, with an expectation value that depends on t:

$$\mu_f(t) = E\left\{f(t;s_i)\right\} = \int_{-\infty}^{\infty} z p_f(z;t) dz \tag{2.18}$$

We define the autocorrelation function as

$$R_{ff}(t_1,t_2) \equiv E\left\{f(t_1;s_i)f(t_2;s_i)\right\} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} z_1 z_2 p_f(z_1,z_2;t_1,t_2) dz_1 dz_2 \tag{2.19}$$

The autocovariance $C_{ff}(t_1,t_2)$ of a random process is:

$$\begin{aligned} C_{ff}(t_1,t_2) &= E\left\{\left[f(t_1;s_i)-\mu_f(t_1)\right]\left[f(t_2;s_i)-\mu_f(t_2)\right]\right\} \\ &= E\left\{f(t_1;s_i)f(t_2;s_i) - f(t_1;s_i)\mu_f(t_2) - \mu_f(t_1)f(t_2;s_i) + \mu_f(t_1)\mu_f(t_2)\right\} \\ &= E\left\{f(t_1;s_i)f(t_2;s_i)\right\} - E\left\{f(t_1;s_i)\right\}\mu_f(t_2) - \mu_f(t_1)E\left\{f(t_2;s_i)\right\} + \mu_f(t_1)\mu_f(t_2) \\ &= R_{ff}(t_1,t_2) - \mu_f(t_1)\mu_f(t_2) - \mu_f(t_1)\mu_f(t_2) + \mu_f(t_1)\mu_f(t_2) \\ &= R_{ff}(t_1,t_2) - \mu_f(t_1)\mu_f(t_2) \end{aligned} \tag{2.20}$$

2.3.4 Two random processes

Cross correlation of two random processes:

$$R_{fg}(t_1,t_2) \equiv E\left\{f(t_1;s_i)g(t_2;s_j)\right\} \tag{2.21}$$

Cross covariance of two random processes:

$$\begin{aligned} C_{fg}(t_1,t_2) &= E\left\{\left[f(t_1;s_i)-\mu_f(t_1)\right]\left[g(t_2;s_j)-\mu_g(t_2)\right]\right\} \\ &= E\left\{f(t_1;s_i)g(t_2;s_j) - f(t_1;s_i)\mu_g(t_2) - \mu_f(t_1)g(t_2;s_j) + \mu_f(t_1)\mu_g(t_2)\right\} \\ &= E\left\{f(t_1;s_i)g(t_2;s_j)\right\} - E\left\{f(t_1;s_i)\right\}\mu_g(t_2) - \mu_f(t_1)E\left\{g(t_2;s_j)\right\} + \mu_f(t_1)\mu_g(t_2) \\ &= R_{fg}(t_1,t_2) - \mu_f(t_1)\mu_g(t_2) - \mu_f(t_1)\mu_g(t_2) + \mu_f(t_1)\mu_g(t_2) \\ &= R_{fg}(t_1,t_2) - \mu_f(t_1)\mu_g(t_2) \end{aligned} \tag{2.22}$$

For two uncorrelated random processes we have:

$$C_{fg}(t_1,t_2) = 0 \tag{2.23}$$

2.3.5 Stationary random process

A random process is homogeneous if

1. its expectation value does not depend on t and

2. its autocorrelation function is translation invariant:

$$R_{ff}(t_1,t_2) = E\left\{f(t_1;s_i)f(t_2;s_i)\right\} = E\left\{f(t_1+t_0;s_i)f(t_2+t_0;s_i)\right\} \tag{2.24}$$

for any t_0.

The autocorrelation function $R_{ff}(t_1, t_2)$ of a stationary random process depends only on the difference time $t_1 - t_2$. This is easily seen from (2.24) by setting $t_0 = -t_2$.

A stationary random process is otherwise known as a **homogenous** process.

2.3.6 Some characteristics of the autocorrelation function

The autocorrelation function of a signal is a 2-dimensional matrix.

For uncorrelated zero mean data, this matrix will be diagonal with the non-zero elements along the diagonal equal to the variance at each sample position.

If each sample of a signal conveys information that cannot be predicted from the other samples, then the autocovariance matrix of the signal must be diagonal.

2.3.7 Temporal statistics of a random process

Temporal average:

$$\mu(s_i) \equiv \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} f(t; s_i) dt \tag{2.25}$$

The result $\mu(s_i)$ is a function of the outcome on which f depends, ie $\mu(s_i)$ is a random variable.

Temporal autocorrelation function of the random process:

$$R(t_0; s_i) \equiv \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} f(t; s_i) f(t + t_0; s_i) dt \tag{2.26}$$

This is another random variable.

2.3.8 Ergodicity

A random process is ergodic with respect to the mean if it is stationary and its temporal average is equal to the ensemble average.

A random process is ergodic with respect to the autocorrelation function, if it is stationary and its temporal autocorrelation function is equal to its ensemble autocorrelation function:

$$E\left\{f(t; s_i) f(t + t_0; s_i)\right\} = \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} f(t; s_i) f(t + t_0; s_i) dt = R(t_0; s_i) \tag{2.27}$$

A random process is ergodic when it is ergodic with respect to the mean and with respect to its autocorrelation function.

Important implication of ergodicity:

If an ensemble of signals is ergodic, we can calculate its mean and autocorrelation function by simply calculating temporal averages over any signal of the ensemble we happen to have.

All the random processes encountered in this course will be assumed to be ergodic. The importance of this is that time averages can be easily measured, whereas ensemble averages cannot.

We can now interpret some engineering measures of the noise $n(t)$ in terms of statistical quantities: Note that the mean value $E\{n(t)\}$ locates the centre of gravity of the area under the probability density function of its values.

$$\text{DC component:} \quad E\{n(t)\} = \langle n(t) \rangle \tag{2.28}$$

$$\text{Average power:} \quad E\{n^2(t)\} = \langle n^2(t) \rangle \tag{2.29}$$

Notice that for a zero-mean process, the variance is equivalent to the average power, i.e., $\sigma^2 = E\{n^2(t)\}$. This could be measured in the lab using a power metre.

Each waveform in this figure represents a different outcome of a random experiment.

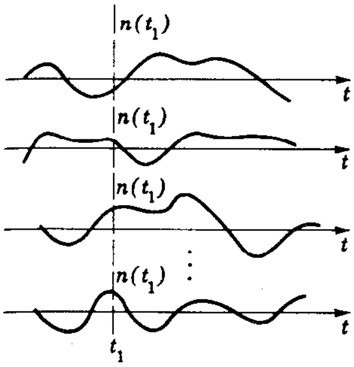


Figure 2.3: Ensemble averages. [Schwartz, Fig. 5-12]

2.3.9 Autocorrelation and Power Spectral Density

To understand bandwidth issues relating to random signals, we must now find a reasonable spectral representation of $n(t)$. In particular, we are interested in a frequency representation that reflects an ensemble average of all possible random processes.

Autocorrelation

The frequency content of a process depends on how rapidly the amplitude changes as a function of time. This can be measured by correlating the amplitudes at times t_1 and t_2. We saw that the *autocorrelation* of a real random process is defined as

$$R_{\mathrm{x}}(t_1, t_2) = E\{x(t_1)x(t_2)\} \tag{2.30}$$

For a stationary process, the autocorrelation depends only on the time difference, so

$$R_{\mathrm{x}}(\tau) = E\{x(t)x(t+\tau)\} \tag{2.31}$$

Recall that the average power of a waveform is the mean square. Hence,

$$\begin{aligned} P &= E\{x^2(t)\} \\ &= R_{\mathrm{x}}(0) \end{aligned} \tag{2.32}$$

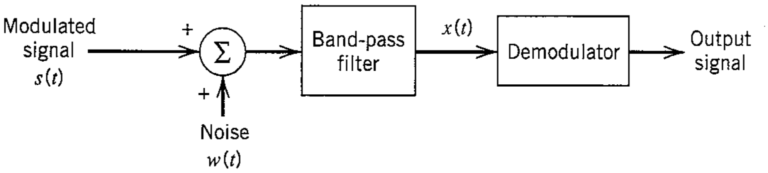


Figure 2.4: Receiver model. [Haykin, Fig. 2.33]

Power Spectral Density

Power spectral density (PSD) is a function that measures the distribution of power of a random signal with frequency. To illustrate the idea, consider a power meter tuned to a frequency f_0 that measures the power in a very narrow bandwidth around f_0; the output of this metre would give a good approximation to the PSD at the frequency f_0. PSD is only defined for stationary signals.[g]

Theorem 2.1 (Wiener-Khinchine Theorem)
The power spectral density of a random process is defined as the Fourier transform of the autocorrelation:

$$\mathbb{S}_{\mathrm{x}}(f) = \int_{-\infty}^{\infty} R_{\mathrm{x}}(\tau)\, e^{-j2\pi f\tau}\, d\tau$$

Since the autocorrelation is thus given by the inverse Fourier transform of the PSD, it follows from (2.32) that the average power of a random process can be found by integrating the PSD over all frequencies:

$$P = R_{\mathrm{x}}(0) = \int_{-\infty}^{\infty} \mathbb{S}_{\mathrm{x}}(f)\, df \tag{2.33}$$

One particular example of a PSD that plays an extremely important role in communications and signal processing is one in which the PSD is constant over all frequencies, i.e.,

$$\mathbb{S}(f) = \frac{N_o}{2} \tag{2.34}$$

Noise having such a PSD is referred to as *white noise*, and is used in the same sense as white light which contains equal amounts of all frequencies within the visible band of electromagnetic radiation. Note that the factor $1/2$ is included to indicate that half the power is associated with positive frequency and half with negative.

2.4 Representation of Bandlimited Noise

2.4.1 Development

Any communication system that uses carrier modulation will typically have a bandpass filter at the front-end of the receiver (see Fig. 2.4). This filter will have a bandwidth wide enough to pass the modulated signal of interest, and is designed to restrict out-of-band noise from entering the receiver. Any noise that does enter the receiver will therefore be bandpass in nature, i.e., its spectral magnitude is non-zero only for some band concentrated around the carrier frequency f_c.

[g] In fact, it is defined for *wide-sense* stationary processes. These are processes for which the mean and variance are independent of time. For a strictly stationary process, all ensemble averages are time-invariant.

For example, if white noise have a PSD of $N_o/2$ is passed through such a filter, then the PSD of the noise that enters the receiver is given by

$$\mathbb{S}(f) = \begin{cases} \frac{N_o}{2}, & f_c - W \leq |f| \leq f_c + W \\ 0, & \text{otherwise} \end{cases} \tag{2.35}$$

and is shown in Fig. 2.5. We are now in a position to develop a representation specifically for such bandpass noise. To achieve this, we will use a simple artifice, namely, to visualise the noise as being composed of the sum of many closely spaced randomly-phased sine waves.

Consider the bandpass noise signal $n(t)$, whose PSD is given by (2.35) and is shown in Fig. 2.5. The average noise power in the frequency slices Δf at frequencies f_k and $-f_k$, is found from (2.33) to be

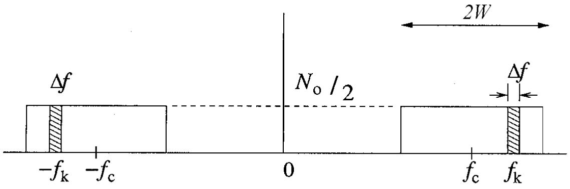


Figure 2.5: Power spectral density of the bandlimited white noise process $n(t)$.

$$P_k = 2\,\frac{N_o}{2}\Delta f = N_o \Delta f \tag{2.36}$$

where the factor of 2 is present because we are summing the slices at negative and positive frequencies. For Δf small, the component associated with this frequency interval can be written[h]

$$n_k(t) = a_k \cos(2\pi f_k t + \theta_k) \tag{2.37}$$

where θ_k is a random phase assumed independent and uniformly distributed in the range $[0, 2\pi)$, and a_k is a random amplitude. It can be shown that the average power of the randomly-phased sinusoid (2.37) is

$$P_k = \frac{E\{a_k^2\}}{2} \tag{2.38}$$

The complete bandpass noise waveform $n(t)$ can be constructed by summing up such sinusoids over the entire band, i.e.,

$$\begin{aligned} n(t) &= \sum_k n_k(t) \\ &= \sum_k a_k \cos(2\pi f_k t + \theta_k) \end{aligned} \tag{2.39}$$

where

$$f_k = f_c + k\Delta f. \tag{2.40}$$

Now, let $f_k = (f_k - f_c) + f_c$, and using the identity for the $\cos(\cdot)$ of a sum[i] we obtain the required result.

[h] Recall that the $\cos(2\pi f)$ wave contains frequency components at both f and $-f$.

[i] $\cos(A + B) = \cos A \cos B - \sin A \sin B$

2.4.2 Result

$$n(t) = n_c(t)\cos(2\pi f_c t) - n_Q(t)\sin(2\pi f_c t) \tag{2.41}$$

where

$$n_I(t) = \sum_k a_k \cos(2\pi(f_k - f_c)t + \theta_k) \tag{2.42}$$

and

$$n_Q(t) = \sum_k a_k \sin(2\pi(f_k - f_c)t + \theta_k) \tag{2.43}$$

From (2.40) we see that $f_k - f_c = k\Delta f$. Hence, $n_I(t)$ and $n_Q(t)$ are *baseband* signals. The representation for $n(t)$ given by (2.41) is the representation we seek, and is referred to as the *bandpass representation*. Although we have derived it for the specific case of a bandlimited white noise process, it is actually a very general representation that can be used for *any* bandpass signal.[j]

2.4.3 Average power and power spectral density

If this representation of bandpass noise is to be of use in our later analyses, we must find suitable statistical descriptions. Hence, we will now derive the average power in $n(t)$, together with the average power and PSD for both $n_Q(t)$ and $n_I(t)$.

The average power in $n(t)$ is $P_n = E\{n^2(t)\}$. Recall from Sec. 2.3.3 that for a zero-mean Gaussian process the average power is equal to the variance σ^2. Substituting (2.39) yields

$$\begin{aligned} P_n = E\{n^2(t)\} &= E\{\sum_k a_k \cos(2\pi f_k t + \theta_k) \sum_l a_l \cos(2\pi f_l t + \theta_l)\} \\ &= \sum_k \sum_l E\{a_k a_l \cos(2\pi f_k t + \theta_k)\cos(2\pi f_l t + \theta_l)\} \\ &= \sum_k \sum_l E\{a_k a_l\}\, E\{\cos(2\pi f_k t + \theta_k)\cos(2\pi f_l t + \theta_l)\}. \end{aligned} \tag{2.44}$$

Note that frequencies are not random variables, while phases are. So the expectation operator refers to phases and not to frequencies. Then since we have assumed that the phase terms are independent, it follows that[k]

$$E\{\cos(2\pi f_k t + \theta_k)\cos(2\pi f_l t + \theta_l)\} = 0, \text{for } k \neq l, \tag{2.45}$$

and

$$E\{\cos(2\pi f_k t + \theta_k)\cos(2\pi f_l t + \theta_l)\} = E\{\cos^2(2\pi f_k t + \theta_k)\} = \frac{1}{2}, \text{for } k = l. \tag{2.46}$$

Hence,

$$P_n = E\{n(t)^2\} = \sum_k \frac{E\{a_k^2\}}{2} = \sigma^2 \tag{2.47}$$

[j] A more general derivation would be based on the Wiener-Khinchine relation, and would involve integrals rather than summations. In this course, however, a bandlimited noise representation is all that is required. Details for general bandpass signals can be found, for example, in Chapter 4 of Couch.

[k] If the phase terms are independent, then $E\{\cos(\cdot)\cos(\cdot)\} = E\{\cos(\cdot)\}E\{\cos(\cdot)\}$.

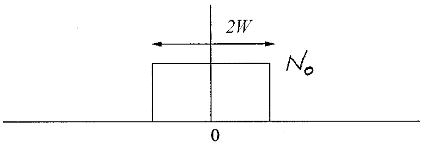


Figure 2.6: Power spectral density of each of the baseband noise processes $n_I(t)$ and $n_Q(t)$.

This is what you should intuitively expect to obtain, given (2.38). A similar derivation for each of $n_I(t)$ and $n_Q(t)$ reveals that

$$P_c = E\{n_I(t)^2\} = \sum_k \frac{E\{a_k^2\}}{2} = \sigma^2 \tag{2.48}$$

and

$$P_s = E\{n_Q(t)^2\} = \sum_k \frac{E\{a_k^2\}}{2} = \sigma^2 \tag{2.49}$$

Thus, the average power in *each* of the baseband waveforms $n_I(t)$ and $n_Q(t)$ is *identical* to the average power in the bandpass noise waveform $n(t)$.

Now, considering the PSD of $n_I(t)$ and $n_Q(t)$, we note from (2.42) and (2.43) that each of these waveforms consists of a sum of closely spaced baseband sinusoids. Thus, each baseband noise waveform will have the same PSD, which is shown in Fig. 2.6. Since the average power in each of the baseband waveforms is the same as the average power in the bandpass waveform, it follows that the area under the PSD in Fig. 2.6 must equal the area under the PSD in Fig. 2.5. The PSD of $n_I(t)$ and $n_Q(t)$ is therefore given by

$$\mathbb{S}_c(f) = \mathbb{S}_s(f) = \begin{cases} N_o, & |f| \le W \\ 0, & \text{otherwise} \end{cases} \tag{2.50}$$

2.4.4 A phasor interpretation

Finally, we will interpret the bandpass representation in another way. Notice that (2.41) can be written

$$n(t) = \mathfrak{Re}\left\{g(t)e^{j2\pi f_c t}\right\} \tag{2.51}$$

where

$$g(t) = n_I(t) + jn_Q(t) \tag{2.52}$$

and $\mathfrak{Re}\{\cdot\}$ denotes the real part. We can also write $g(t)$ in terms of magnitude and phase as

$$g(t) = r(t)e^{j\phi(t)} \tag{2.53}$$

where $r(t) = \sqrt{n_I(t)^2 + n_Q(t)^2}$ is the envelope and $\phi(t) = \tan^{-1}[n_Q(t)/n_I(t)]$ is the phase of the noise. The phasor diagram representation is shown in Fig. 2.7.

Because of this representation, $n_I(t)$ is often referred to as the *in-phase* component (the real part), and $n_Q(t)$ as the *quadrature-phase* component (imaginary part). Substituting the magnitude-phase representation for $g(t)$ into (2.51) gives

$$n(t) = r(t)\cos[2\pi f_c t + \phi(t)] \tag{2.54}$$

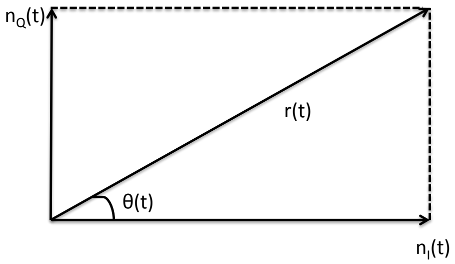


Figure 2.7: Phasor representation of bandlimited noise.

This is an intuitively satisfying result. Since the bandpass noise $n(t)$ is narrow band in the vicinity of f_c, one would expect it to be oscillating on the average at f_c. It can be shown that if $n_I(t)$ and $n_Q(t)$ are Gaussian-distributed, then the magnitude $r(t)$ has a Rayleigh distribution, and the phase $\phi(t)$ is uniformly distributed.

References

- Lathi, sections 10.1, 10.2, 10.3, 11.1, 11.2, 11.4, 11.5
- Couch, sections 4-1, 6-1, 6-2
- Haykin, sections 1.2, 1.7, 1.9

Chapter 3

Noise in Analog Communication Systems

3.1 Background

You have previously studied ideal analog communication systems. Our aim here is to compare the performance of different analog modulation schemes in the presence of noise. The performance will be measured in terms of the signal-to-noise ratio (SNR) at the output of the receiver, defined as

$$\mathrm{SNR}_o = \frac{\text{average power of message signal at the receiver output}}{\text{average power of noise at the receiver output}} \tag{3.1}$$

Note that this measure is unambiguous if the message and noise are additive at the receiver output; we will see that in some cases this is not so, and we need to resort to approximation methods to obtain a result.

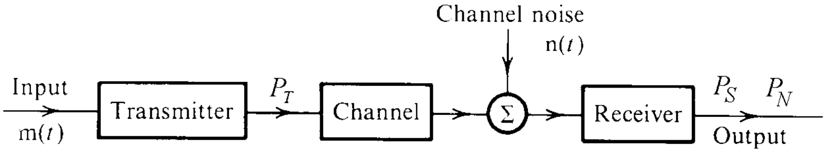


Figure 3.1: Model of an analog communication system. [Lathi, Fig. 12.1]

A model of a typical communication system is shown in Fig. 3.1, where we assume that a modulated signal with power P_T is transmitted over a channel with additive noise. At the output of the receiver the signal and noise powers are P_S and P_N respectively, and hence, the output SNR is $\mathrm{SNR}_o = P_S/P_N$. This ratio can be increased as much as desired simply by increasing the transmitted power. However, in practice the maximum value of P_T is limited by considerations such as transmitter cost, channel capability, interference with other channels, etc. In order to make a fair comparison between different modulation schemes, we will compare systems having the same transmitted power.

Also,we need a common measurement criterion against which to compare the difference modulation schemes. For this, we will use the *baseband* SNR. Recall that all modulation schemes are bandpass (i.e., the modulated signal is centred around a carrier frequency). A baseband communication system is one that does not use modulation. Such a scheme is suitable for transmission over wires, say, but is not terribly practical. As we will see, however, it does allow a direct performance comparison of different schemes.

3.2 Baseband Communication System

A baseband communication system is shown in Fig. 3.2(a), where $m(t)$ is the band-limited message signal, and W is its bandwidth.

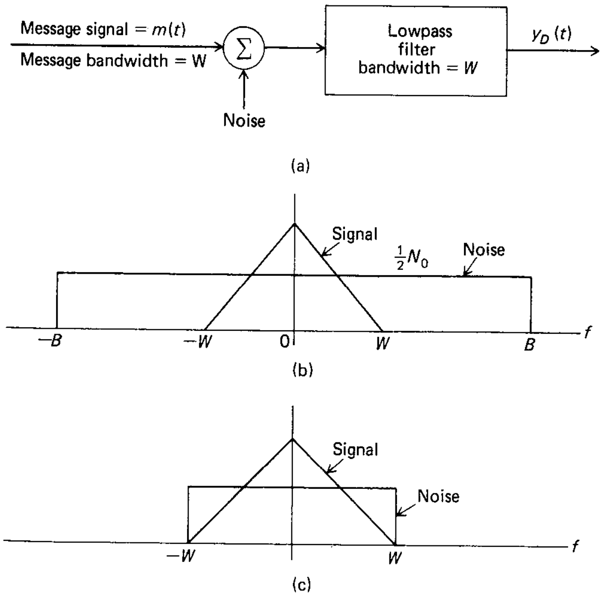


Figure 3.2: Baseband communication system: (a) model, (b) signal spectra at filter input, and (c) signal spectra at filter output. [Ziemer & Tranter, Fig. 6.1]

An example signal PSD is shown in Fig. 3.2(b). The average signal power is given by the area under the triangular curve marked "Signal", and we will denote it by P. We assume that the additive noise has a double-sided white PSD of $N_o/2$ over some bandwidth $B > W$, as shown in Fig. 3.2(b). For a basic baseband system, the transmitted power is identical to the message power, i.e., $P_T = P$.

The receiver consists of a low-pass filter with a bandwidth W, whose purpose is to enhance the SNR by cutting out as much of the noise as possible. The PSD of the noise at the output of the LPF is shown in Fig. 3.2(c), and the average noise power is given by

$$\int_{-W}^{W} \frac{N_o}{2}\, df = N_o W \tag{3.2}$$

Thus, the SNR at the receiver output is

$$\text{SNR}_{\text{baseband}} = \frac{P_T}{N_o W} \tag{3.3}$$

Notice that for a baseband system we can improve the SNR by: (a) increasing the transmitted power ($P_T \uparrow$), (b) restricting the message bandwidth ($W \downarrow$), or (c) making the receiver less noisy ($N_o \downarrow$).

3.3 Amplitude Modulation

3.3.1 Review

In amplitude modulation, the *amplitude* of a sinusoidal carrier wave is varied linearly with the message signal. The general form of an AM signal is

$$s(t)_{\text{AM}} = [A + m(t)]\cos(2\pi f_c t) \tag{3.4}$$

where A is the amplitude of the carrier, f_c is the carrier frequency, and $m(t)$ is the message signal. The *modulation index*, μ, is defined as

$$\mu = \frac{m_p}{A}, \tag{3.5}$$

where m_p is the peak amplitude of $m(t)$, i.e., $m_p = \max|m(t)|$. Recall that if $\mu \leq 1$, (i.e., $A \geq m_p$), then the envelope of $s(t)$ will have the same shape as the message $m(t)$, and thus, a simple envelope detector can be used to demodulate the AM wave. The availability of a particularly simple receiver is the major advantage of AM, since as we will see, its noise performance is not great.

If an envelope detector cannot be used, another form of detection known as *synchronous detection* can be used.[a] The block diagram of a synchronous detector is shown in Fig. 3.3. The process involves multiplying the waveform at the receiver by a local carrier of the same frequency (and phase) as the carrier used at the transmitter. This basically replaces the $\cos(\cdot)$ term in (3.4) by a $\cos^2(\cdot)$ term. From the identity

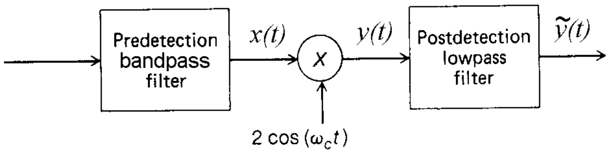


Figure 3.3: Synchronous demodulator. [Ziemer & Tranter, Fig. 6.2]

$$2\cos^2(x) = 1 + \cos(2x) \tag{3.6}$$

the result is a frequency translation of the message signal, down to baseband (i.e., $f = 0$) and up to twice the carrier frequency. The low-pass filter is then used to recover the baseband message signal. As one might expect, the main disadvantage with this scheme is that it requires generation of a local carrier signal that is perfectly synchronised with the transmitted carrier.

Notice in (3.4) that the AM signal consists of two components, the carrier $A\cos(2\pi f_c t)$ and the sidebands $m(t)\cos(2\pi f_c t)$. Since transmitting the carrier term is wasteful, another variation

[a] This is also known as coherent detection, or a product demodulator.

of AM that is of interest is one in which the carrier is suppressed. This is referred to as *double-sideband suppressed carrier* (DSB-SC), and is given by

$$s(t)_{\text{DSB-SC}} = Am(t)\cos(2\pi f_c t) \tag{3.7}$$

In this case the envelope of the signal looks nothing like the original message signal, and a synchronous detector must be used for demodulation (see next).

3.3.2 Noise in DSB-SC

The *predetection* signal (i.e., just before the multiplier in Fig. 3.3) is

$$x(t) = s(t) + n(t) \tag{3.8}$$

The purpose of the predetection filter is to pass only the frequencies around the carrier frequency, and thus reduce the effect of out-of-band noise. The noise signal $n(t)$ after the predetection filter is bandpass with a double-sided white PSD of $N_o/2$ over a bandwidth of $2W$ (centred on the carrier frequency), as shown in Fig. 2.5. Hence, using the bandpass representation given by equation (2.41) into equation (3.7), the predetection signal is

$$x(t) = [Am(t) + n_I(t)]\cos(2\pi f_c t) - n_Q(t)\sin(2\pi f_c t) \tag{3.9}$$

After multiplying by $2\cos(2\pi f_c t)$, this becomes

$$\begin{aligned} y(t) &= 2\cos(2\pi f_c t)x(t) \\ &= Am(t)2\cos(2\pi f_c t)^2 + n_I(t)2\cos(2\pi f_c t)^2 - n_Q(t)\sin(4\pi f_c t) \\ &= Am(t)[1+\cos(4\pi f_c t)] + n_I(t)[1+\cos(4\pi f_c t)] \\ &\quad -n_Q(t)\sin(4\pi f_c t) \end{aligned} \tag{3.10}$$

where we have used (3.6) and

$$2\cos x \sin x = \sin(2x) \tag{3.11}$$

Low-pass filtering will remove all of the $2f_c$ frequency terms, leaving

$$\tilde{y}(t) = Am(t) + n_I(t) \tag{3.12}$$

The signal power at the receiver output is[b]

$$P_S = E\{A^2m^2(t)\} = A^2E\{m^2(t)\} = A^2P \tag{3.13}$$

where, recall, P is the power in the message signal $m(t)$. The power in the noise signal $n_I(t)$ is

$$P_N = \int_{-W}^{W} N_o df = 2N_oW \tag{3.14}$$

since from (2.50) the PSD of $n_I(t)$ is N_o and the bandwidth of the LPF is W. Thus, for the DSB-SC synchronous demodulator, the SNR at the receiver output is

$$\text{SNR}_o = \frac{A^2P}{2N_oW} \tag{3.15}$$

[b] Note that the power in $A\cos(2\pi f_c t)$ is $A^2/2$, but the power in A is A^2.

To make a fair comparison with a baseband system, we need to calculate the transmitted power

$$P_T = E\{A^2 m^2(t) \cos^2(2\pi f_c t)\} = \frac{A^2 P}{2} \tag{3.16}$$

and substitution gives

$$\text{SNR}_o = \frac{P_T}{N_o W} \tag{3.17}$$

Comparison with (3.3) gives

$$\text{SNR}_{\text{DSB-SC}} = \text{SNR}_{\text{baseband}} \tag{3.18}$$

We conclude that a DSB-SC system provides no SNR performance gain over a baseband system.

It turns out that an SSB (single sideband) system also has the same SNR performance as a baseband system (see next).

3.3.3 Noise in AM, Synchronous Detection

For an AM waveform, using equations (2.41) and (3.4) the predetection signal is

$$x(t) = [A + m(t) + n_I(t)] \cos(2\pi f_c t) - n_Q(t) \sin(2\pi f_c t) \tag{3.19}$$

After multiplication by $2\cos(2\pi f_c t)$, this becomes

$$\begin{aligned} y(t) &= A[1 + \cos(4\pi f_c t)] + m(t)[1 + \cos(4\pi f_c t)] \\ &\quad + n_I(t)[1 + \cos(4\pi f_c t)] - n_Q(t) \sin(4\pi f_c t) \end{aligned} \tag{3.20}$$

After low-pass filtering this becomes

$$\tilde{y}(t) = A + m(t) + n_I(t) \tag{3.21}$$

Note that the DC term A can be easily removed with a DC block (i.e., a capacitor), and most AM demodulators are not DC-coupled.

The signal power at the receiver output is

$$P_S = E\{m^2(t)\} = P \tag{3.22}$$

and the noise power is

$$P_N = 2N_o W \tag{3.23}$$

The SNR at the receiver output is therefore

$$\text{SNR}_o = \frac{P}{2N_o W} \tag{3.24}$$

The transmitted power for an AM waveform is

$$P_T = \frac{A^2}{2} + \frac{P}{2} \tag{3.25}$$

and substituting this into the baseband SNR (3.3) we find that for a baseband system with the same transmitted power

$$\text{SNR}_{\text{baseband}} = \frac{A^2 + P}{2N_o W} \tag{3.26}$$

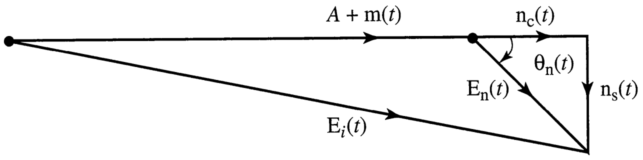


Figure 3.4: Phasor diagram of the signals present at an AM receiver. [Lathi, Fig. 12.5]

Thus, for an AM waveform using a synchronous demodulator we have

$$\text{SNR}_{\text{AM}} = \frac{P}{A^2 + P}\,\text{SNR}_{\text{baseband}} \tag{3.27}$$

In other words, the performance of AM is always worse than that of a baseband system. This is because of the wasted power which results from transmitting the carrier explicitly in the AM waveform.

3.3.4 Noise in AM, Envelope Detection

Recall that an envelope detector can only be used if $\mu \leq 1$. An envelope detector works by detecting the *envelope* of the received signal (3.19). To get an appreciation of the effect of this, we will represent the received signal by phasors, as shown in Fig. 3.4. The receiver output, denoted by "$E_i(t)$" in the figure, will be given by

$$\begin{aligned} y(t) &= \text{envelope of } x(t) \\ &= \sqrt{[A + m(t) + n_I(t)]^2 + n_Q(t)^2} \end{aligned} \tag{3.28}$$

This expression is somewhat more complicated than the others we have looked at, and it is not immediately obvious how we will find the SNR at the receiver output. What we would like is an approximation to $y(t)$ in which the message and the noise are additive.

(a) Small Noise Case

The receiver output can be simplified if we assume that for almost all t the noise power is small, i.e., $n(t) \ll [A + m(t)]$. Hence

$$|A + m(t) + n_I(t)| \gg |n_Q(t)| \tag{3.29}$$

Then, most of the time,

$$y(t) \approx A + m(t) + n_I(t) \tag{3.30}$$

which is identical to the post-detection signal in the case of synchronous detection. Thus, (ignoring the DC term A again) the output SNR is

$$\text{SNR}_o = \frac{P}{2N_oW} \tag{3.31}$$

which can be written in terms of baseband SNR as

$$\text{SNR}_{\text{env}} = \frac{P}{A^2 + P}\,\text{SNR}_{\text{baseband}} \tag{3.32}$$

Note that whereas SNR_{AM} in (3.27) is valid always, the expression for SNR_{env} is *only* valid for small noise power.

(b) Large Noise Case

Now consider the case where noise power is large, so that for almost all t we have $n(t) \gg [A + m(t)]$. Rewrite (3.28) as

$$\begin{aligned} y^2(t) &= [A + m(t) + n_I(t)]^2 + n_Q(t)^2 \\ &= A^2 + m^2(t) + n_I^2(t) + 2Am(t) + 2An_I(t) + 2m(t)n_I(t) + n_Q^2(t) \end{aligned} \tag{3.33}$$

For $n_I(t) \gg [A + m(t)]$, this reduces to

$$\begin{aligned} y^2(t) &\approx n_I^2(t) + n_Q^2(t) + 2[A + m(t)]n_I(t) \\ &= E_n^2(t)\left(1 + \frac{2[A + m(t)]n_I(t)}{E_n^2(t)}\right) \end{aligned} \tag{3.34}$$

where $E_n(t) = \sqrt{n_I^2(t) + n_Q^2(t)}$ is the envelope of the noise (as described in Section 2.4.4). But from the phasor diagram in Fig. 3.4, we have $n_I(t) = E_n(t)\cos\theta_n(t)$, giving

$$y(t) \approx E_n(t)\sqrt{1 + \frac{2[A + m(t)]\cos\theta_n(t)}{E_n(t)}} \tag{3.35}$$

Further, $\sqrt{1 + x} \approx 1 + x/2$ for $x \ll 1$, so this reduces to

$$\begin{aligned} y(t) &\approx E_n(t)\left(1 + \frac{[A + m(t)]\cos\theta_n(t)}{E_n(t)}\right) \\ &= E_n(t) + [A + m(t)]\cos\theta_n(t) \end{aligned} \tag{3.36}$$

The main thing to note about (3.36) is that the output of the envelope detector contains no term that is proportional to the message $m(t)$. The term $m(t)\cos\theta_n(t)$ is the message multiplied by a noise term $\cos\theta_n(t)$, and is no use in recovering $m(t)$. This multiplicative effect corrupts the message to a far greater extent than the additive noise in our previous analysis; the result is that there is a complete loss of information at the receiver. This produces a *threshold effect*, in that below some carrier power level, the performance of the detector deteriorates very rapidly.

Despite this threshold effect, we find that in practice it does not matter terribly. This is because the quality of a signal with an output SNR less than about 25 dB is so poor, that no-one would really want to listen to it anyway. And for such a high output SNR, we are well past the threshold level and we find that (3.27) holds. From a practical point of view, the threshold effect is seldom of importance for envelope detectors.

3.4 Frequency Modulation

Having studied the effect of additive noise on amplitude modulation systems, we will now look at the SNR performance on frequency modulation systems. There is a fundamental difference between these two. In AM, the message information is contained within the amplitude of the signal, and since the noise is additive it adds directly to the modulated signal. For FM, however, it is the frequency of the modulated signal that contains the message information. Since the frequency of a signal can be described by its zero crossings, the effect of noise on an FM signal is determined by the extent to which it changes the zero crossing of the modulated signal. This suggests that the effect of noise on an FM signal will be less than that for an AM system, and we will see in this section that this is in fact the case.

3.4.1 Review

Consider the following general representation of a carrier waveform

$$s(t) = A\cos[\theta_i(t)] \tag{3.37}$$

where $\theta_i(t)$ is the *instantaneous phase angle*. Comparing this with the generic waveform $A\cos(2\pi ft)$, where f is the frequency, we can define the *instantaneous frequency* of (3.37) as

$$f_i(t) = \frac{1}{2\pi}\frac{d\theta_i(t)}{dt} \tag{3.38}$$

For an FM system, the instantaneous frequency of the carrier is varied linearly with the message, i.e.,

$$f_i(t) = f_c + k_f\, m(t) \tag{3.39}$$

where k_f is the *frequency sensitivity* of the modulator. Hence, the instantaneous phase is

$$\begin{aligned}\theta_i(t) &= 2\pi\int_{-\infty}^{t} f_i(\tau)\, d\tau \\ &= 2\pi f_c t + 2\pi k_f \int_{-\infty}^{t} m(\tau)\, d\tau \end{aligned} \tag{3.40}$$

and by using equations (3.4) and (3.37) the modulated signal is

$$s(t) = A\cos\left[2\pi f_c t + 2\pi k_f \int_{-\infty}^{t} m(\tau)\, d\tau\right] \tag{3.41}$$

There are two things to note about the FM signal: (a) the envelope is constant, and (b) the signal $s(t)$ is a non-linear function of the message signal $m(t)$.

Bandwidth of FM

Let the peak message amplitude be $m_p = \max|m(t)|$, so that the instantaneous frequency will vary between $f_c - k_f m_p$ and $f_c + k_f m_p$. Denote the deviation of the instantaneous frequency from the carrier frequency as the *frequency deviation*

$$\Delta f = k_f m_p \tag{3.42}$$

Define the *deviation ratio* (also called the FM modulation index in the special case of tone-modulated FM) as

$$\beta = \frac{\Delta f}{W} \tag{3.43}$$

where W is the message bandwidth.

Unlike AM, the bandwidth of FM is not dependent simply on the message bandwidth. For small β, the FM bandwidth is approximately twice the message bandwidth (referred to as narrow-band FM). But for large β (referred to as wide-band FM) the bandwidth can be much larger than this. A useful rule-of-thumb for determining the transmission bandwidth of an FM signal is *Carson's rule*:

$$B_T = 2W(\beta + 1) = 2(\Delta f + W) \tag{3.44}$$

Observe that for $\beta \ll 1$, $B_T \approx 2W$ (as is the case in AM). At the other extreme, for $\beta \gg 1$, $B_T \approx 2\Delta f$, which is independent of W.

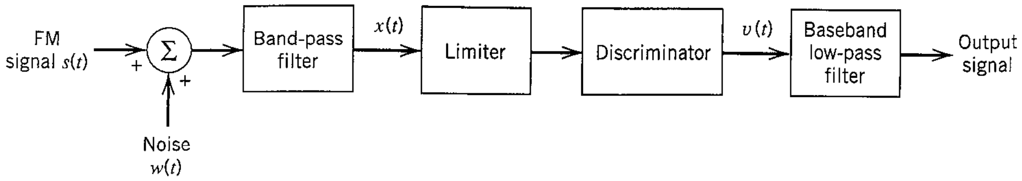


Figure 3.5: Model of an FM receiver. [Haykin Fig. 2.40]

3.4.2 Noise in FM

The model of an FM receiver is shown in Fig. 3.5, where $s(t)$ is the FM signal (3.41), and $w(t)$ is white Gaussian noise with power spectral density $N_o/2$. The bandpass filter is used to remove any signals outside the bandwidth of $f_c \pm B_T/2$, and thus, the predetection noise at the receiver is bandpass with a bandwidth of B_T. Since an FM signal has a constant envelope, the limiter is used to remove any amplitude variations. The discriminator is a device whose output is proportional to the deviation in the instantaneous frequency (i.e., it recovers the message signal), and the final baseband low-pass filter has a bandwidth of W and thus passes the message signal and removes out-of-band noise.

Using equations (2.41) and (3.41) the predetection signal is

$$x(t) = A\cos\left[2\pi f_c t + 2\pi k_f \int_{-\infty}^{t} m(\tau)\,d\tau\right] + n_I(t)\cos(2\pi f_c t) - n_Q(t)\sin(2\pi f_c t) \qquad (3.45)$$

First, let us consider the signal power at the receiver output. When the predetection SNR is high, it can be shown that the noise does not affect the power of the signal at the output.[c] Thus, ignoring the noise, the instantaneous frequency of the input signal is

$$f_i = f_c + k_f m(t) \qquad (3.46)$$

and the output of the discriminator (which is designed to simply return the deviation of the instantaneous frequency away from the carrier frequency) is $k_f m(t)$. The output signal power is therefore

$$P_S = k_f^2 P \qquad (3.47)$$

where P is the average power of the message signal.

Now, to calculate the noise power at the receiver output, it turns out that for high predetection SNR the noise output is approximately independent of the message signal.[d] In this case, we only have the carrier and noise signals present. Thus,

$$\tilde{x}(t) = A\cos(2\pi f_c t) + n_I(t)\cos(2\pi f_c t) - n_Q(t)\sin(2\pi f_c t) \qquad (3.48)$$

The phasor diagram of this is shown in Fig. 3.6. From this diagram, we see that the instantaneous phase is

$$\theta_i(t) = \tan^{-1}\frac{n_Q(t)}{A + n_I(t)} \qquad (3.49)$$

[c] The derivation of this can be found in D. Sakrison, *Communication Theory*, John Wiley & Sons, New York, 1968.

[d] Again, the proof of this is not very exciting, so we shall take it as given.

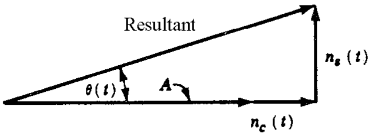


Figure 3.6: Phasor diagram of the FM carrier and noise signals. [Taub & Schilling, Fig. 9.2-1]

For large carrier power, then most of the time

$$\begin{aligned} \theta_i(t) &= \tan^{-1}\frac{n_Q(t)}{A} \\ &\approx \frac{n_Q(t)}{A} \end{aligned} \tag{3.50}$$

where the last line follows from $\tan\epsilon \approx \epsilon$ for small ϵ. But, the discriminator output is the instantaneous frequency, given by

$$\begin{aligned} f_i(t) &= \frac{1}{2\pi}\frac{d\theta_i(t)}{dt} \\ &= \frac{1}{2\pi A}\frac{dn_Q(t)}{dt} \end{aligned} \tag{3.51}$$

We know the PSD of $n_Q(t)$ shown in Fig. 3.7(a), but what is the PSD of $dn_Q(t)/dt$?

Fourier theory tells us that:

$$\begin{aligned} \text{if} \quad x(t) &\leftrightarrow X(f) \\ \text{then} \quad \frac{dx(t)}{dt} &\leftrightarrow j2\pi f X(f) \end{aligned}$$

In other words, differentiation with respect to time is the same as passing the signal through a system having a transfer function of $H(f) = j2\pi f$.

It can be shown[e] that if a signal with PSD $\mathbb{S}_i(f)$ is passed through a linear system with transfer function $H(f)$, then the PSD at the output of the system is $\mathbb{S}_o(f) = |H(f)|^2\mathbb{S}_i(f)$.

Proof

Assume that a system has impulse response function $h(t)$ with transfer function $H(t)$. An input signal $x(t)$ passing through this system comes out as $y(t)$:

$$y(t) = h(t) \star x(t) \tag{3.52}$$

The corresponding relationship of their Fourier transforms is

$$Y(f) = H(f)X(f) \tag{3.53}$$

where $Y(f)$ and $X(f)$ are the Fourier transforms of $y(t)$ and $x(t)$ respectively. We would like to know the power spectral density (PSD) of $y(t)$.

[e] Lathi p.510, or Couch p. 420.

We know that the PSD is the Fourier transform of the autocorrelation of the signal. Let us compute therefore, the autocorrelation function of the output signal:

$$\begin{aligned} R_y(\tau) &\equiv E\{y(t)y(t+\tau)\} \\ &= E\left\{\int_{-\infty}^{\infty} h(t_1)x(t-t_1)dt_1 \times \int_{-\infty}^{\infty} h(t_2)x(t+\tau-t_2)dt_2\right\} \\ &= E\left\{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(t_1)x(t-t_1)h(t_2)x(t+\tau-t_2)dt_1dt_2\right\} \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(t_1)h(t_2)E\left\{x(t-t_1)x(t+\tau-t_2)\right\}dt_1dt_2 \end{aligned} \tag{3.54}$$

We remember that by definition

$$E\left\{x(t-t_1)x(t+\tau-t_2)\right\} = R_x(t+\tau-t_2-(t-t_1)) = R_x(\tau-t_2+t_1) \tag{3.55}$$

Upon substitution in (3.54) we obtain:

$$\begin{aligned} R_y(\tau) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(t_1)h(t_2)R_x(\tau-t_2+t_1)dt_1dt_2 \\ &= \int_{-\infty}^{\infty} h(t_1)\left[\int_{-\infty}^{\infty} h(t_2)R_x(\tau+t_1-t_2)dt_2\right]dt_1 \end{aligned} \tag{3.56}$$

We recognise inside the square bracket the convolution of $h(w)$ with $R_x(w)$, with w being $\tau + t_1$. The result will be a function say $s(w) \equiv h(w) \star R_x(w) \Rightarrow s(\tau+t_1) = h(\tau+t_1) \star R_x(\tau+t_1)$. Then we shall have

$$R_y(\tau) = \int_{-\infty}^{\infty} h(t_1)s(\tau+t_1)dt_1 \tag{3.57}$$

To interprete this integral as a convolution integral, we must have $h(t_1)$ multiplying a function with argument $(? - t_1)$, where ? stands for something. So, the question is, "how can we re-write $s(\tau+t_1)$ so that its argument has the desired form?". Let us define a new function $u(x)$ such that $u(x) = s(-x)$. Then, I can write $s(\tau+t_1) = u(-\tau-t_1)$ and substitute in (3.57):

$$R_y(\tau) = \int_{-\infty}^{\infty} h(t_1)u(-\tau-t_1)dt_1 \tag{3.58}$$

This integral now is the convolution of a function $h(v)$ with function $u(v)$ where v stands for $-\tau$ here. So we have:

$$R_y(\tau) = h(-\tau) \star u(-\tau) = h(-\tau) \star s(\tau) \tag{3.59}$$

If I re-write (3.59) in the Fourier domain, we obtain:

$$PSD_y = FT(h(-\tau)) \times FT(s(\tau)) \tag{3.60}$$

Here we remembered that the FT of the autocorrelation function is the PSD of the signal. We also remember that if $D(f)$ is the FT of a signal $d(t)$, the FT of $d(-t)$ is $D^*(f)$ where * means complex conjugate. Then we may write:

$$PSD_y = H^*(f) \times FT(s(\tau)) \tag{3.61}$$

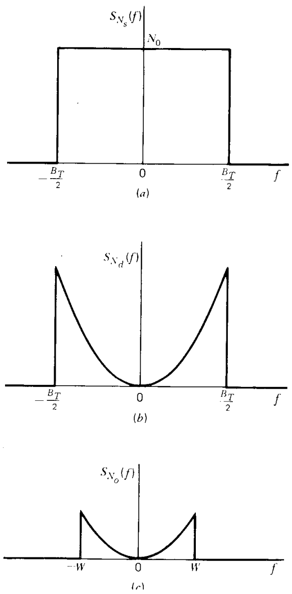


Figure 3.7: Power spectral densities for FM noise analysis: (a) $n_Q(t)$, (b) $dn_Q(t)/dt$, and (c) noise at the receiver output. [Haykin, Fig. 2.42]

We remember that $s(\tau)$ was an auxiliary function we introduced for convenience and it actually is $s(\tau) = h(\tau) \star R_x(\tau)$. So, its FT is the product of the FTs of $h(\tau)$, ie $H(f)$, and $R_x(\tau)$, ie the power spectrum of the input signal $x(t)$:

$$PSD_y = H^*(f) \times H(f) \times PSD_x = |H(f)|^2 \times PSD_x \tag{3.62}$$

This concludes the proof of the statement and concludes this diversion from the main thread of argument.

If the PSD of $n_Q(t)$ has a value of N_o within the band $\pm B_T/2$ as shown in Fig. 3.7(a), then $dn_Q(t)/dt$ has a PSD of $|j2\pi f|^2 N_o$. The PSD of $dn_Q(t)/dt$ before and after the baseband LPF is shown in Fig. 3.7(b) and (c) respectively.

Returning to (3.51), now that the PSD of $dn_Q(t)/dt$ is known, we can calculate the average noise power at the receiver output. It is given by

$$P_N = \int_{-W}^{W} \mathbb{S}_D(f)\, df \tag{3.63}$$

where $\mathbb{S}_D(f)$ is the PSD of the noise component at the discriminator output (i.e., the PSD of $f_i(t)$ in (3.51)); the limits of integration are taken between $-W$ and W to reflect the fact that the output signal is low-pass filtered. Thus,

$$\begin{aligned} P_N &= \int_{-W}^{W} \left(\frac{1}{2\pi A}\right)^2 |j2\pi f|^2 N_o \, df \\ &= \int_{-W}^{W} \frac{N_o}{A^2} f^2 \, df \\ &= \frac{2N_o W^3}{3A^2} \end{aligned} \tag{3.64}$$

This expression is quite important, since it shows that the average noise power at the output of a FM receiver is inversely proportional to the carrier power $A^2/2$. Hence, increasing the carrier power has a *noise quieting* effect. This is one of the major advantages of FM systems.

Finally, we have that at the output the SNR is

$$\text{SNR}_o = \frac{3A^2 k_f^2 P}{2N_o W^3} \tag{3.65}$$

Since the transmitted power of an FM waveform is

$$P_T = A^2/2 \tag{3.66}$$

substitution into (3.3) gives

$$\text{SNR}_{\text{FM}} = \frac{3k_f^2 P}{W^2} \, \text{SNR}_{\text{baseband}} = 3\beta^2 \frac{P}{m_p^2} \, \text{SNR}_{\text{baseband}} \tag{3.67}$$

The SNR expression (3.67) is based on the assumption that the carrier power is large compared to the noise power. It is found that, like an AM envelope detector, the FM detector exhibits a *threshold effect*. As the carrier power decreases, the FM receiver breaks, as Haykin describes: “At first, individual clicks are heard in the receiver output, and as the carrier-to-noise ratio decreases still further, the clicks rapidly merge into a crackling or sputtering sound”.[f] Experimental studies indicate that this noise mutilation is negligible in most cases if the predetection SNR (i.e., just after the receiver bandpass filter) is above 10. In other words, the threshold point occurs around

$$\frac{A^2}{2N_o B_T} = 10 \tag{3.68}$$

where, recall, $B_T = 2W(\beta + 1)$. For predetection SNRs above this value, the output SNR is given by (3.67).

One should note that whereas (3.67) suggests that output SNR for an FM system can be increased arbitrarily by increasing β while keeping the signal power fixed, inspection of (3.68) shows this not to be strictly true. The reason is that if β increases too far, the condition (3.68) that we are above threshold may no longer be true, meaning that (3.67) no longer provides an expression for the true SNR.

[f] A qualitative analysis of the FM threshold effect can be found in Haykin pp. 149–152, and Taub & Schilling devote an entire chapter to the subject.

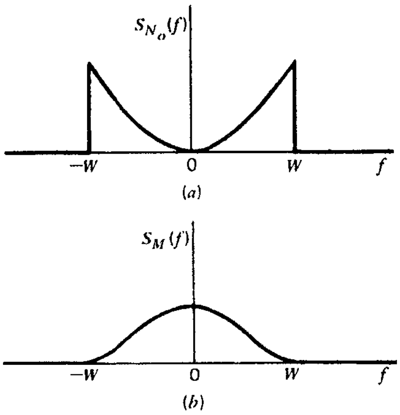


Figure 3.8: Power spectral densities of: (a) noise at the output of FM receiver, and (b) a typical message signal. [Haykin, Fig. 2.48]

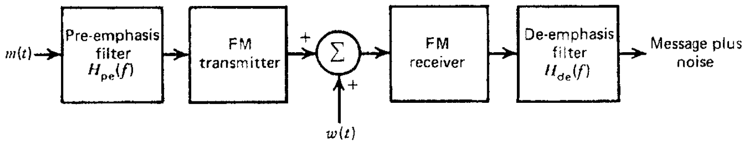

m(t)
Pre-emphasis filter $H_{pe}(f)$
FM transmitter
+
Σ
+
w(t)
FM receiver
De-emphasis filter $H_{de}(f)$
Message plus noise

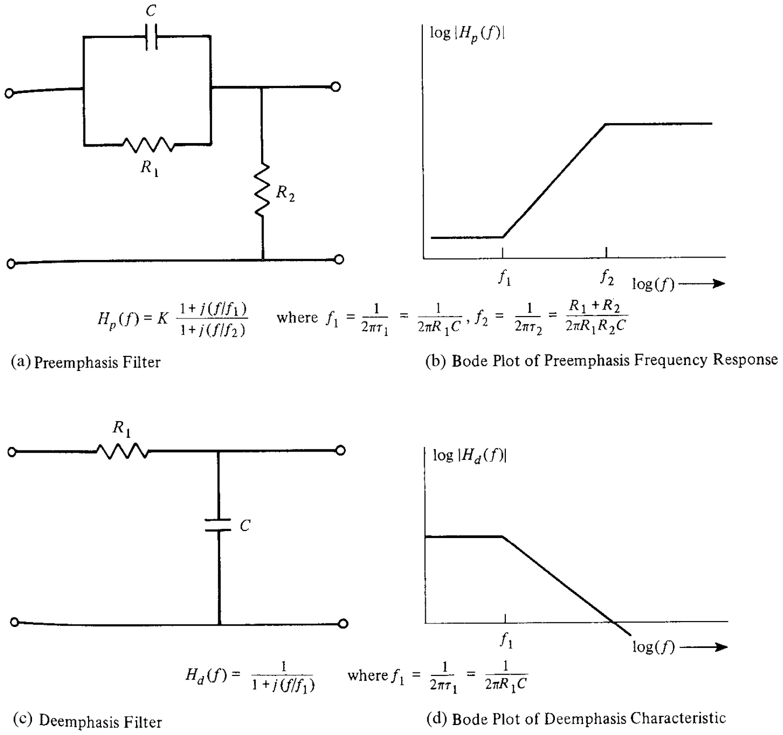
Figure 3.9: Pre-emphasis and de-emphasis in an FM system. [Haykin, Fig. 2.49]

3.4.3 Pre-emphasis and De-emphasis

There is another way in which the SNR of an FM system may be increased. We saw in the previous subsection that the PSD of the noise at the detector output has a square-law dependence on frequency. On the other hand, the PSD of a typical message source is not uniform, and typically rolls off at around 6 dB per decade (see Fig. 3.8). We note that at high frequencies the relative message power is quite low, whereas the noise power is quite high (and is rapidly increasing). It is possible that this situation could be improved by reducing the bandwidth of the transmitted message (and the corresponding cutoff frequency of the baseband LPF in the receiver), thus rejecting a large amount of the out-of-band noise. In practice, however, the distortion introduced by low-pass filtering the message signal is unsatisfactory.

A better solution is obtained by using the *pre-emphasis* and *de-emphasis* stages shown in Fig. 3.9. The intention of this scheme is that $H_{pe}(f)$ is used to artificially emphasise the high frequency components of the message prior to modulation, and hence, before noise is introduced. This serves to effectively equalise the low- and high-frequency portions of the message PSD such that the message more fully utilises the bandwidth available to it. At the receiver, $H_{de}(f)$ performs the inverse operation by de-emphasising the high frequency components, thereby restoring the original PSD of the message signal.

Simple circuits that perform pre- and de-emphasis are shown in Fig. 3.10, along with their respective frequency responses. Haykin shows (see Example 2.6, p.156) that these circuits can improve the output SNR by around 13 dB. In closing this section, we also note that Dolby noise


C
R_1
R_2
log $|H_p(f)|$
f_1
f_2
log(f) →
$H_p(f) = K \frac{1+j(f/f_1)}{1+j(f/f_2)}$ where $f_1 = \frac{1}{2\pi\tau_1} = \frac{1}{2\pi R_1 C}$, $f_2 = \frac{1}{2\pi\tau_2} = \frac{R_1+R_2}{2\pi R_1 R_2 C}$
R_1
C
log $|H_d(f)|$
f_1
log(f) →
$H_d(f) = \frac{1}{1+j(f/f_1)}$ where $f_1 = \frac{1}{2\pi\tau_1} = \frac{1}{2\pi R_1 C}$

(a) Preemphasis Filter

(b) Bode Plot of Preemphasis Frequency Response

(c) Deemphasis Filter

(d) Bode Plot of Deemphasis Characteristic

Figure 3.10: Simple linear pre-emphasis and de-emphasis circuits. [Couch, Fig. 5-16]

reduction uses an analogous pre-emphasis technique to reduce the effects of noise.

3.5 Comparison of Analogue Communication Systems

To conclude this analysis of the effect of noise on various analog modulation schemes, we will now compare the relative performance of the schemes. In making the comparison we assume the following:

(i) single-tone modulation, ie:

$$m(t) = A_m \cos(2\pi f_m t) \Rightarrow P = \frac{A_m^2}{2} \Rightarrow A_m = \sqrt{2P} \tag{3.69}$$

where we made use of equation (1.5);

(ii) the message bandwidth $W = f_m$;

(iii) for the AM system, $\mu = 1$, which with the help of equation (3.5) $\mu = m_p/A$ implies

$$1 = \frac{m_p}{A} \Rightarrow A = m_p = peak\ amplitude\ of\ real\ signal \Rightarrow A = \sqrt{2P} \tag{3.70}$$

by making use of (3.69), or

$$m_p = \sqrt{2P} \tag{3.71}$$

(iv) for the FM system, $\beta = 5$ (which is what is used in commercial FM transmission, with $\Delta f = 75$ kHz, and $W = 15$ kHz).

With these assumptions we find that the SNR expressions for the various modulation schemes become:

$$\text{SNR}_{\text{DSB-SC}} = \text{SNR}_{\text{baseband}} \tag{3.72}$$

This is the same as equation (3.18) which is always valid. From equation (3.27)

$$\text{SNR}_{\text{AM}} = \frac{P}{A^2 + P}\,\text{SNR}_{\text{baseband}} \tag{3.73}$$

and by substitution from (3.70) we obtain

$$\text{SNR}_{\text{AM}} = \frac{P}{2P + P}\text{SNR}_{\text{baseband}} = \frac{1}{3}\,\text{SNR}_{\text{baseband}} \tag{3.74}$$

From equation (3.67) we have

$$\text{SNR}_{\text{FM}} = 3\beta^2 \frac{P}{m_p^2}\,\text{SNR}_{\text{baseband}} = 3\beta^2 \frac{P}{2P}\,\text{SNR}_{\text{baseband}} \tag{3.75}$$

where we used (3.71). Then

$$\text{SNR}_{\text{FM}} = \frac{3}{2}\,\beta^2 \text{SNR}_{\text{baseband}} = \frac{75}{2}\text{SNR}_{\text{baseband}} \tag{3.76}$$

where we used $\beta = 5$.

The plots of these equations are shown in Fig. 3.11. We make the following comments (which are based on the above assumptions):

AM: The SNR performance is 4.8 dB worse than a baseband system, and the transmission bandwidth is $B_T = 2W$.

DSB: The SNR performance is identical to a baseband system, and the transmission bandwidth is $B_T = 2W$ (for SSB, the SNR performance is again identical, but the transmission bandwidth is only $B_T = W$).

FM: The SNR performance is 15.7 dB better than a baseband system, and the transmission bandwidth is $B_T = 2(\beta + 1)W = 12W$ (with pre- and de-emphasis the SNR performance is increased by about 13 dB with the same transmission bandwidth).

References

- Lathi, sections 4.1,4.2,4.3 (AM review); 5.1,5.2,5.5 (FM review); 12.1, 12.2, 12.3
- Couch, sections 5-6, 7-8, 7-9
- Haykin, sections 2.2, 2.3, 2.6, 2.7, 2.10, 2.11, 2.12, 2.13

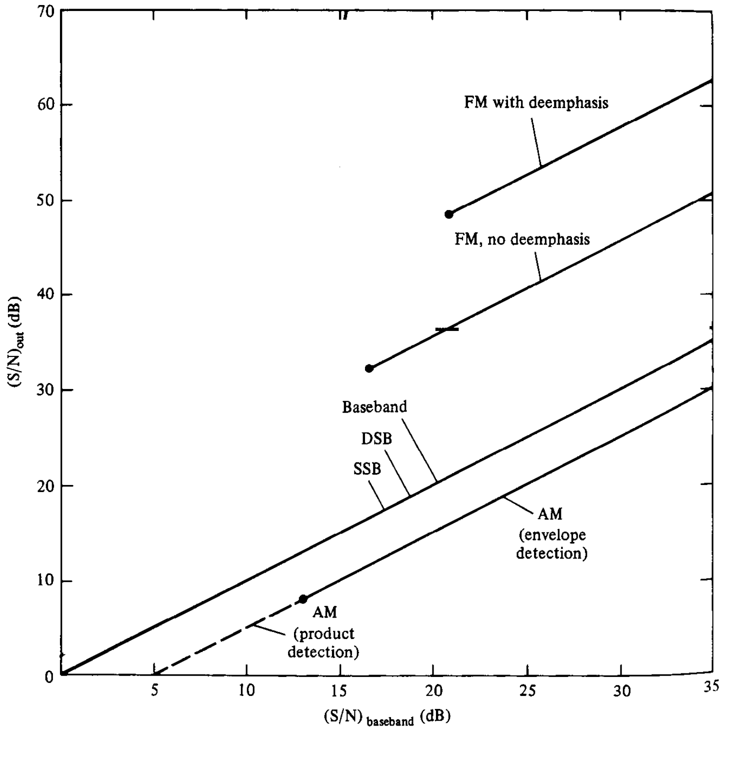


Figure 3.11: Noise performance of analog communication systems. [Couch, Fig. 7-27]

Chapter 4

Digital Communication Systems

4.1 Background

In analog communication systems the primary goal is to reproduce the waveforms accurately. If the received waveform matches the transmitted waveform closely, then the message signal will be correctly interpreted at the receiver. Hence, the performance criterion we used in Chapter 3 was the signal-to-noise ratio at the output of the receiver. In digital communication systems though, at any given time the transmitter sends one of a finite number of symbols. The goal of the receiver is not to reproduce the transmitted waveform accurately, since the possible waveforms are already known exactly. Rather, the receiver aims to correctly identify which one of the finite number of symbols was sent. The performance measure for a digital system will therefore be the probability of the receiver making a decision error.

4.2 Sampling

Analog signals can be converted through sampling to discrete-time samples. Providing these samples are taken at a *sufficient* rate, the analog signal can be reconstructed exactly from its discrete-time samples. Thus, in order to transmit the information in an analog signal, it is only necessary to transmit its samples.

4.2.1 Sampling Theorem

Nyquist's sampling theorem tells us what this sufficient rate is [Lathi, p.251]:

> A signal, the spectrum of which is band-limited to W Hz, can be reconstructed exactly from its samples taken uniformly at a rate of $R > 2W$ Hz. In other words, the minimum sampling frequency is $f_s = 2W$ Hz.

4.2.2 Maximum Transmission Rate

Suppose we have a channel of bandwidth B Hz, what is the maximum rate of message transfer over that channel, in the absence of noise? Neglecting noise, a channel of bandwidth B Hz, will pass a signal of bandwidth B Hz without distortion. This signal can be exactly represented by its Nyquist samples, which occur at a rate of $2B$ Hz (according to Nyquist's sampling theorem). So the channel of bandwidth B Hz can transmit $2B$ independent pieces of information. This is one of the basic relationships in communications.

4.3 Pulse-Code Modulation

4.3.1 Background

Pulse-code modulation (PCM) is a baseband scheme that can be used to represent *any* analog signal (e.g., voice, video, etc.) in digital form. There are three major advantages of using a digital representation of analog signals: (i) digital signals are more immune to channel noise, (ii) repeaters along the transmission path can detect a digital signal and retransmit a new noise-free signal, and (iii) PCM signals derived from all types of analog sources can be represented using a uniform format. PCM was first introduced into the American telephone network in 1962, and is now used in every telephone network in the world (although as often happens, different standards are used in America than elsewhere).

PCM is essentially a particular type of analog-to-digital conversion in which an analog signal is sampled in time, and the amplitude of each sample is rounded off to the nearest one of a finite set of allowable values. Thus, the analog signal is represented by samples that are discrete in both time and amplitude.

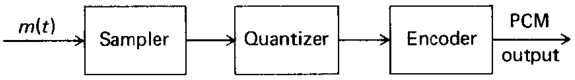


Figure 4.1: PCM modulator [Ziemer & Tranter, Fig. 3.67]

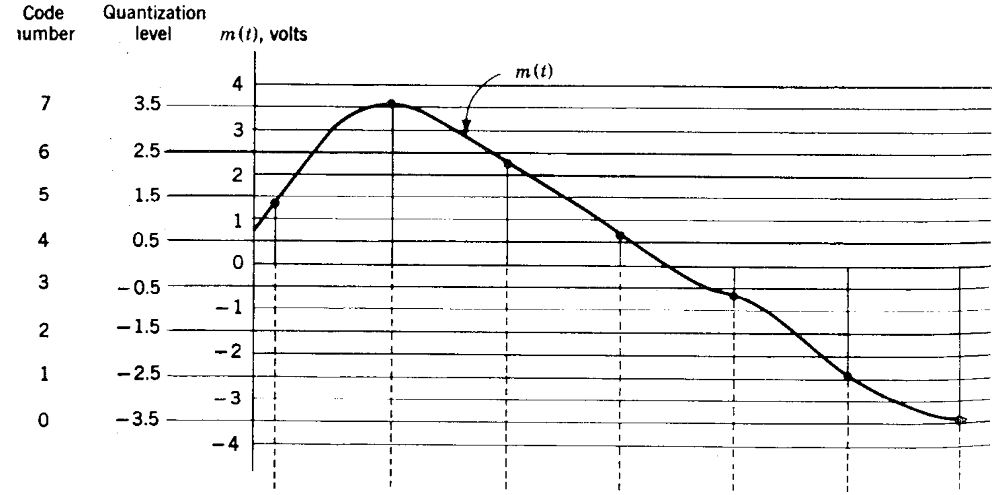
Consider a message signal $m(t)$ having a bandwidth W. The generation of a PCM signal is a three-step process as shown in Fig. 4.1. First the message signal is sampled, and providing the sampling frequency is above $2W$, then Nyquist's sampling theorem tells us that perfect reconstruction of the message is possible. Next the sampled signal is *quantised*, i.e., the amplitude of each sample is rounded to the nearest discrete level. Finally, the discrete amplitudes are encoded into a binary codeword. This process is illustrated in Fig. 4.2.

4.3.2 Quantisation Noise

Notice that quantisation is a destructive process—once a signal has been quantised it cannot be perfectly reconstructed. The noise so introduced is called *quantisation noise*, and we will now analyse its effect.

Consider a quantiser with uniform separation of Δ volts between quantising levels. The quantisation error is a random variable bounded by $-\Delta/2 \leq q \leq \Delta/2$. Assume that it is uniformly distributed over the available range, with a probability density function

$$p(q) = \begin{cases} \frac{1}{\Delta}, & -\Delta/2 \leq q \leq \Delta/2 \\ 0, & \text{otherwise} \end{cases} \tag{4.1}$$



Sample value	1.3	3.6	2.3	0.7	−0.7	−2.4	−3.4
Nearest quantization level	1.5	3.5	2.5	0.5	−0.5	−2.5	−3.5
Code number	5	7	6	4	3	1	0
Binary representation	101	111	110	100	011	001	000

Figure 4.2: The PCM process. [Taub & Schilling, Fig. 5.9-1]

The mean square error is

$$P_N = E\{e^2\} = \int_{-\infty}^{\infty} q^2\, p(q)\, dq$$
$$= \frac{1}{\Delta}\int_{-\Delta/2}^{\Delta/2} q^2\, dq = \frac{\Delta^2}{12} \tag{4.2}$$

Assume the encoded symbol has n bits, so that the maximum number of quantising levels is $L = 2^n$. The maximum peak-to-peak *dynamic range* of the quantiser is therefore $2^n\Delta$. Let the power of the message signal be P, and denote the maximum absolute value of the message signal by $m_p = \max|m(t)|$. Assume that the message signal fully loads loads the quantiser, such that

$$m_p = \frac{1}{2}2^n\Delta = 2^{n-1}\Delta \tag{4.3}$$

This gives a SNR at the quantiser output of

$$\text{SNR}_o = \frac{P_S}{P_N} = \frac{P}{\Delta^2/12} \tag{4.4}$$

But, from (4.3) we have

$$\Delta^2 = \frac{m_p^2}{(2^{n-1})^2} = \frac{4m_p^2}{2^{2n}} \tag{4.5}$$

Substitution gives the SNR as

$$\text{SNR}_o = \frac{3P}{m_p^2}\, 2^{2n} \tag{4.6}$$

or, expressing this ratio in decibels

$$\begin{aligned} \text{SNR}_o(\text{dB}) &= 10\log_{10}\left(2^{2n}\right) + 10\log_{10}\left(\frac{3P}{m_p^2}\right) \\ &= 6.02n + 10\log_{10}\left(\frac{3P}{m_p^2}\right) \text{ dB} \qquad (4.7) \end{aligned}$$

Hence, each extra bit in the encoder adds 6 dB to the output SNR of the quantiser.

Example 4.1 – *Sinusoidal message signal*

Consider a full-load sinusoidal signal of amplitude A_m, i.e., $m(t) = A_m \cos(2\pi f_m t)$. The average signal power is

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

and the maximum signal value is $m_p = A_m$. Substitution into (4.6) gives

$$\text{SNR}_o = \frac{3A_m^2}{2A_m^2}\, 2^{2n}$$

Expressing this in decibels gives

$$\text{SNR}_o(\text{dB}) = 6.02n + 1.76 \ \text{ dB}.$$

□

In practice, the quantisation noise can generally be made so small that it is negligible to the end user of the message. For example, audio CDs use 16-bit PCM to achieve a quantisation SNR of greater than 90 dB.

4.3.3 Bandwidth Requirements

For an n-bit quantiser with $L = 2^n$ quantisation levels, each sample of an input signal is represented using n bits. If the signal is bandlimited to W Hz, then its PCM representation contains $2nW$ bits per second. Recall from Section 4.2.2 that, in the absence of channel noise, we can transmit 2 bits per second per hertz. Therefore, PCM requires a minimum transmission bandwidth of

$$B_T = nW \qquad (4.8)$$

Recall from (4.6) that the output SNR due to quantisation is

$$\text{SNR}_o = \frac{3P}{m_p^2}\, 2^{2n}$$

But from (4.8), $n = B_T/W$. Substitution gives

$$\text{SNR}_o = \frac{3P}{m_p^2}\, 2^{2B_T/W} \qquad (4.9)$$

Hence, in a PCM system, SNR increases *exponentially* with the transmission bandwidth B_T, i.e., a small increase in bandwidth yields a large increase in SNR.

4.3.4 Companding

There is a final component that is generally part of a PCM system, especially one used for voice signals. It is found that typically small signal amplitudes occur more often than large signal amplitudes. This means that often the signal does not use the entire range of quantisation levels available. Notice that the amount of quantisation noise added to a given sample (4.2) is independent of the signal amplitude. Small signal amplitudes will therefore suffer more from quantisation effects than large amplitude signals.

To counteract this effect it is better to have more closely-spaced quantisation levels at low signal amplitudes and more widely-spaced levels at high signal amplitude. In practice, a uniform quantiser is easier to implement, so to achieve non-uniform quantisation, the input signal is first compressed, then quantised and transmitted, and then expanded at the receiver. This process of *comp*ressing and then exp*anding* the signal is referred to as *companding*. The exact SNR gain obtained with companding naturally depends on the exact form of the compression used. However, each extra bit in the encoder still adds an extra 6 dB to the quantisation SNR.

The concept of predistorting a message signal in order to achieve better performance in the presence of noise, and then removing the distortion at the receiver, should be familiar—that is precisely what was done in Chapter 3 with pre-emphasis and de-emphasis circuits in FM systems.

4.4 Baseband Data Transmission

The effect of additive noise on digital transmission is that at the receiver, a symbol 1 is sometimes mistaken for a symbol 0, and vice versa. This leads to *bit errors*. Here we wish to derive an expression for the probability of error, i.e., the probability that the symbol at the receiver output differs from the transmitted symbol.

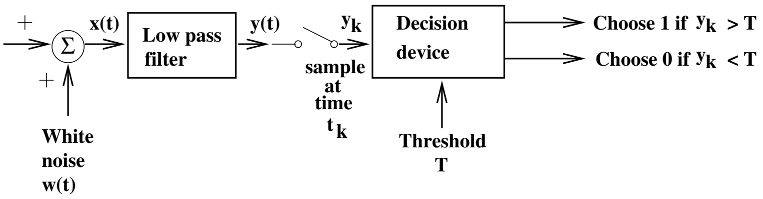


Figure 4.3: Model of a binary baseband data communication system.

Consider the binary data communication system shown in Fig. 4.3. In this system, the symbol "0" is represented by 0 volts, and the symbol "1" is represented by A volts. This is a *unipolar* system.[a] We assume that the channel noise is additive white Gaussian noise, with a PSD of $N_0/2$. The received signal is first low-pass filtered to the signal bandwidth W (to remove out-of-band noise) where W is chosen large enough to pass the digital waveform essentially unchanged. It is then sampled, and the sampled value is compared with some predetermined threshold.

After the LPF, the predetection signal is

$$y(t) = s(t) + n(t) \tag{4.10}$$

[a] Note that the development in Haykin section 4.3 uses *bipolar* symbols, i.e., "1" is A and "0" is $-A$.

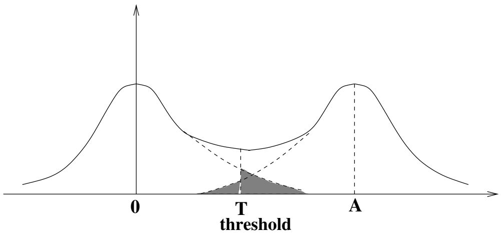


Figure 4.4: Probability density function for binary data transmission in noise. The choice of a threshold, used to discriminate between the two symbols, interpretes wrongly the symbols that arrive with noise-corrupted values falling in the shaded tails of the two Gaussians: anything with value lower than T will be interpreted as 0 even if it comes from the Gaussian on the right; anything with value greater than T will be interpreted as 1 even if it comes from the Gaussian on the left.

where $s(t)$ is the binary-valued function (either 0 or A volts), and $n(t)$ is the low-pass filtered additive white Gaussian noise with zero mean and variance

$$\sigma^2 = \int_{-W}^{W} N_0/2 \, df = N_0 W. \tag{4.11}$$

Recall that a sample value N of $n(t)$ is a Gaussian random variable the probability density function of which is

$$p_N(n) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{n^2}{2\sigma^2}\right). \tag{4.12}$$

This PDF is also known as a *normal* distribution, and often denoted as $\mathcal{N}(0, \sigma^2)$ (i.e., a mean of zero, and variance of σ^2).

Let Y denote a sample value of $y(t)$. If a symbol 0 was transmitted, then $y(t) = n(t)$ and Y will have a PDF of $\mathcal{N}(0, \sigma^2)$. If, however, a symbol 1 was transmitted, then $y(t) = A + n(t)$ and Y will have a PDF of $\mathcal{N}(A, \sigma^2)$. Let us consider a threshold T we use to separate the two sets of values. We then make receiver decisions as:

$$\begin{array}{lll} \text{if} & Y < T, & \text{choose symbol } 0 \\ \text{if} & Y > T, & \text{choose symbol } 1 \end{array}$$

We must choose T so the total error of symbol interpretation is minimised. Such a detector is referred to as a *maximum likelihood detector*.

There are two cases of decision error:

(i) a symbol 0 was transmitted, but a symbol 1 was chosen

(ii) a symbol 1 was transmitted, but a symbol 0 was chosen

These errors correspond to the shaded regions in Fig. 4.4. Let P_{e0} be the conditional probability of error, given that the symbol 0 was transmitted. This is defined as

$$P_{e0} = \frac{1}{\sigma\sqrt{2\pi}} \int_{T}^{\infty} \exp\left(-\frac{n^2}{2\sigma^2}\right) dn \tag{4.13}$$

The probability of error event (i) occurring is equal to the probability of an error, given a symbol 0 has been transmitted, multiplied by the probability of a symbol 0 being transmitted, i.e.,

$$p_{(\text{i})} = p_0 P_{e0}$$

where p_0 is the *a priori* probability of transmitting a symbol 0.

Similarly, the conditional probability of error, given that the symbol 1 was transmitted is

$$P_{e1} = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{T} \exp\left(-\frac{(n-A)^2}{2\sigma^2}\right) dn \tag{4.14}$$

and the probability of error event (ii) occurring is

$$p_{(ii)} = p_1 P_{e1}$$

where p_1 is the *a priori* probability of transmitting a symbol 1.

The probability of error event (i) occurring is

$$p_{(i)} = p_0 P_{e0} = (1-p_1) P_{e0} \tag{4.15}$$

The total error probability then will be a function of the chosen threshold T and given by

$$\begin{aligned} P_e(T) &= p_{(i)} + p_{(ii)} \\ &= p_1 P_{e1} + (1-p_1) P_{e0} \qquad (4.16) \\ &= p_1 \int_{-\infty}^{T} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(n-A)^2}{2\sigma^2}\right) dn + (1-p_1) \int_{T}^{\infty} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{n^2}{2\sigma^2}\right) dn \end{aligned}$$

We wish to choose T so that $P_e(T)$ is minimum:

$$\frac{dP_e(T)}{dT} = 0 \tag{4.17}$$

We have to substitute the expression of $P_e(T)$ from (4.16) into (4.17) in order to solve for T. This means that we have to differentiate the right-hand side of (4.16) with respect to T and set the result to 0. The right-hand side of (4.16) depends on T via the limits of the integrals. To perform such a differentiation we have to invoke the Leibnitz rule of differentiating an integral with respect to a parameter: If

$$I(\lambda) = \int_{a(\lambda)}^{b(\lambda)} f(x;\lambda) dx \tag{4.18}$$

then

$$\frac{dI(\lambda)}{d\lambda} = \frac{db(\lambda)}{d\lambda} f(b(\lambda);\lambda) - \frac{da(\lambda)}{d\lambda} f(a(\lambda);\lambda) + \int_{a(\lambda)}^{b(\lambda)} \frac{\partial f(x;\lambda)}{\partial \lambda} dx \tag{4.19}$$

In order to apply this formula to the first integral on the right-hand side of (4.16) we have to use $\lambda = T$, $a(\lambda) = -\infty$ and $b(\lambda) = T$. In order to apply it for the second integral on the right-hand side of (4.16) we have to use $\lambda = T$, $a(\lambda) = T$ and $b(\lambda) = \infty$. In both cases there will be only one term left of formula (4.19) because only one of the limits depends on the parameter with respect to which we differentiate, and the integrand does not depend on it either:

$$\frac{dP_e(T)}{dT} = p_1 \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(T-A)^2}{2\sigma^2}\right) - (1-p_1)\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{T^2}{2\sigma^2}\right) = 0 \tag{4.20}$$

After performing some straightforward manipulation of this expression, we get:

$$\begin{aligned} \frac{p_1}{1-p_1} &= \exp\left(-\frac{T^2-(T-A)^2}{2\sigma^2}\right) \\ &= \exp\left(-\frac{T^2-T^2-A^2+2TA}{2\sigma^2}\right) \\ &= \exp\left(-\frac{A(2T-A)}{2\sigma^2}\right) \end{aligned} \tag{4.21}$$

If we have some prior knowledge on the relative frequency with which the two symbols are transmitted, ie the ration $p_1/(1-p_1)$, we may solve this equation to derive the optimal threshold:

$$\begin{aligned} \ln\frac{p_1}{1-p_1} &= -\frac{A(2T-A)}{2\sigma^2} \Rightarrow \\ 2\sigma^2 \ln\frac{p_1}{1-p_1} &= -A(2T-A) \Rightarrow \\ -\frac{2\sigma^2}{A}\ln\frac{p_1}{1-p_1} &= 2T-A \Rightarrow \\ T = -\frac{\sigma^2}{A}\ln\frac{p_1}{1-p_1} + \frac{A}{2} & \end{aligned} \tag{4.22}$$

If the two symbols are transmitted with equal frequency ($p_1 = p_0 = 1-p_1$), then the first term of the right-hand side of (4.22) is zero ($\ln 1 = 0$), and the optimal threshold is $T = A/2$.

Once we know the threshold we should use, we can use (4.16) to work out the total error. The integrals on the right-hand side of (4.16) cannot be calculated analytically, but they can be defined in terms of some well known and well studied and tabulated function known as “error function” $\mathrm{erf}(z)$:

$$\mathrm{erf}(z) \equiv \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt \tag{4.23}$$

Note that

$$\mathrm{erf}(-z) = -\mathrm{erf}(z) \tag{4.24}$$

The complementary error function $\mathrm{erfc}(z)$ is defined as:

$$\mathrm{erfc}(z) \equiv 1 - \mathrm{erf}(z) \tag{4.25}$$

Let us examine first P_{e0} given by equation (4.13). We may write it as

$$\begin{aligned} P_{e0} &= \frac{1}{\sigma\sqrt{2\pi}} \int_T^\infty \exp\left(-\frac{n^2}{2\sigma^2}\right) dn \\ &= \frac{1}{\sigma\sqrt{2\pi}} \int_0^\infty \exp\left(-\frac{n^2}{2\sigma^2}\right) dn - \frac{1}{\sigma\sqrt{2\pi}} \int_0^T \exp\left(-\frac{n^2}{2\sigma^2}\right) dn \end{aligned} \tag{4.26}$$

Now, the integral from 0 to infinity is half the integral over the full range of values (from $-\infty$ to ∞) and the integrand is symmetric (even function) about 0. So, the integral from 0 to infinity must be equal to half the value of the integral from $-\infty$ to ∞. The integral from $-\infty$ to ∞ is 1 as it is the integral of a probability density function over all possible outcomes. So, the first term on the right-hand side of (4.26) is equal to $1/2$. For the second term, we introduce a new variable of integration $\tilde{n}$, such that $\tilde{n} \equiv n/(\sqrt{2}\sigma)$. Then $dn = \sqrt{2}\sigma d\tilde{n}$ and the limits of integration become from 0 to $T/(\sqrt{2}\sigma)$. Then:

$$\begin{aligned} P_{e0} &= \frac{1}{2} - \frac{1}{\sigma\sqrt{2\pi}} \int_0^{\frac{T}{\sqrt{2}\sigma}} e^{-\tilde{n}^2} \sqrt{2}\sigma d\tilde{n} \\ &= \frac{1}{2} - \frac{1}{\sqrt{\pi}} \int_0^{\frac{T}{\sqrt{2}\sigma}} e^{-\tilde{n}^2} d\tilde{n} \\ &= \frac{1}{2} - \frac{1}{2}\mathrm{erf}\left(\frac{T}{\sqrt{2}\sigma}\right) \\ &= \frac{1}{2}\left[1 - \mathrm{erf}\left(\frac{T}{\sqrt{2}\sigma}\right)\right] \\ &= \frac{1}{2}\mathrm{erfc}\left(\frac{T}{\sqrt{2}\sigma}\right) \end{aligned} \tag{4.27}$$

For the case of equiprobable symbols, $T = A/2$ and

$$P_{e0} = \frac{1}{2}\mathrm{erfc}\left(\frac{A}{2\sqrt{2}\sigma}\right) \tag{4.28}$$

Let us examine next P_{e1} given by equation (4.29):

$$P_{e1} = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{T} \exp\left(-\frac{(n-A)^2}{2\sigma^2}\right) dn \tag{4.29}$$

Let us define a new variable of integration $\tilde{n} \equiv (n - A)/(\sqrt{2}\sigma)$. Then $dn = \sqrt{2}\sigma d\tilde{n}$ and the limits of integration become from $-\infty$ to $(T - A)/(\sqrt{2}\sigma)$:

$$\begin{aligned} P_{e1} &= \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{\frac{T-A}{\sqrt{2}\sigma}} e^{-\tilde{n}^2} \sqrt{2}\sigma d\tilde{n} \\ &= \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\frac{T-A}{\sqrt{2}\sigma}} e^{-\tilde{n}^2} d\tilde{n} \\ &= \frac{1}{\sqrt{\pi}} \int_{-\infty}^{0} e^{-\tilde{n}^2} d\tilde{n} + \frac{1}{\sqrt{\pi}} \int_{0}^{\frac{T-A}{\sqrt{2}\sigma}} e^{-\tilde{n}^2} d\tilde{n} \\ &= \frac{1}{2} + \frac{1}{2}\mathrm{erf}\left(\frac{T-A}{\sqrt{2}\sigma}\right) \\ &= \frac{1}{2} - \frac{1}{2}\mathrm{erf}\left(\frac{A-T}{\sqrt{2}\sigma}\right) \\ &= \frac{1}{2}\left[1 - \mathrm{erf}\left(\frac{A-T}{\sqrt{2}\sigma}\right)\right] \\ &= \frac{1}{2}\mathrm{erfc}\left(\frac{A-T}{\sqrt{2}\sigma}\right) \end{aligned} \tag{4.30}$$

For the case of equiprobable symbols, $T = A/2$ and

$$P_{e1} = \frac{1}{2}\text{erfc}\left(\frac{A}{2\sqrt{2}\sigma}\right) \tag{4.31}$$

That is, $P_{e1} = P_{e0}$. The total error rate will be the weighed sum of these two error rates, with the weights being the prior symbol probabilities, ie $p_1 = p_0 = 1/2$, so the total error rate will also be given by (4.30) since $P_e = 0.5P_{e0} + 0.5P_{e1} = P_{e0} = P_{e1}$. In summary, we have

$$P_e = \frac{1}{2}\text{erfc}\left(\frac{A}{2\sqrt{2}\sigma}\right). \tag{4.32}$$

♠

Remark: Another standard mathematical function, in terms of which P_{e0} and P_{e1} might be expressed is the Q-function (used in the slides), defined as the area under the tail of a normalised Gaussian random variable, $\mathcal{N}(0,1)$. It is defined by

$$Q(u) = \frac{1}{\sqrt{2\pi}} \int_u^\infty \exp(-n^2/2)\, dn.$$

In terms of the Q-function, it can be shown that the total error rate for the case of equiprobable errors is:

$$P_e = Q\left(\frac{A}{2\sigma}\right)$$

The relation between the Q-function and erfc is

$$Q(u) = \frac{1}{2}\text{erfc}\left(\frac{u}{\sqrt{2}}\right).$$

With this transformation, formulas in erfc may be converted into those in $Q(u)$, and vice versa.

Comments

Note that the probability of error (4.32) depends solely on A/σ, the ratio of the signal amplitude to the noise standard deviation (i.e., RMS noise). Thus, the ratio A/σ is the *peak signal-to-noise ratio*.[b] The probability of error versus the peak signal-to-noise ratio is shown in Fig. 4.5. Note that for $A/\sigma = 7.4$ (i.e., 17.4 dB), $P_e = 10^{-4}$. So, if the transmission rate is 10^5 bits/sec, then on average there will be an error every 0.1 seconds. However, if $A/\sigma = 11.2$ (i.e., 21 dB), an increase of just 3.5 dB, P_e decreases to 10^{-8}. And for a transmission rate of 10^5 bits/sec, this means that on average there will be an error only about every 15 minutes. (In practise designers often use $P_e = 10^{-5}$ as a design goal for binary communication systems.)

It should also be noted that the P_e curve exhibits a *threshold effect*. In other words, above some threshold (approximately 18–20 dB) the probability of error decreases very rapidly with small changes in signal strength. Below this threshold, errors occur quite frequently.

[b] If we express this ratio in decibels, we must take $20\log_{10}(A/\sigma)$, since it is a voltage ratio (not a power ratio).

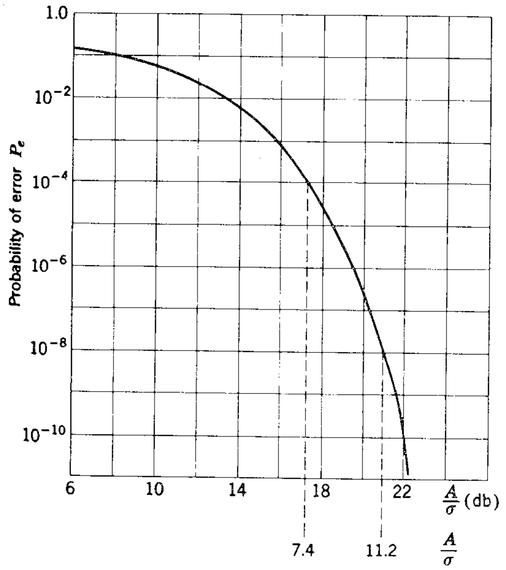


Figure 4.5: Probability of bit error for binary detection in Gaussian noise. [Schwartz, Fig. 5-5]

4.5 Bandpass Data Transmission

4.5.1 Background

Analogously to analog modulation schemes, digital modulation schemes use a step change in the amplitude, phase, or frequency of a sinusoidal carrier to distinguish a symbol 0 from a symbol 1. More complicated forms of digital modulation use a combination of these, e.g., 16-QAM is a hybrid amplitude/phase modulation scheme which can represent symbols from a 16-symbol alphabet. The basic binary transmission modulation schemes are shown in Fig. 4.6

For digital modulation schemes, there are essentially two common methods of demodulation, *synchronous detection* or *envelope detection*. We will only consider synchronous detection here. In digital systems, synchronous detection consists of multiplying the incoming waveform by a locally generated carrier frequency, and then low-pass filtering the resultant multiplied signal.[c]

4.5.2 ASK

In amplitude-shift keying (ASK), the signals used to represent the binary symbols are:

$$s_0(t) = 0 \tag{4.33}$$

$$s_1(t) = A\cos(2\pi f_c t) \tag{4.34}$$

Since there is no signal when the symbol 0 is transmitted, the scheme is also referred to as on-off keying (OOK). More generally, we can write the transmitted signal as

$$s(t) = A(t)\cos(2\pi f_c t), \quad A(t) \in \{0, A\} \tag{4.35}$$

Consider the synchronous detector shown in Fig. 4.7. The predetection filter is used to restrict

[c] Notice that this is identical to an analog synchronous detector.

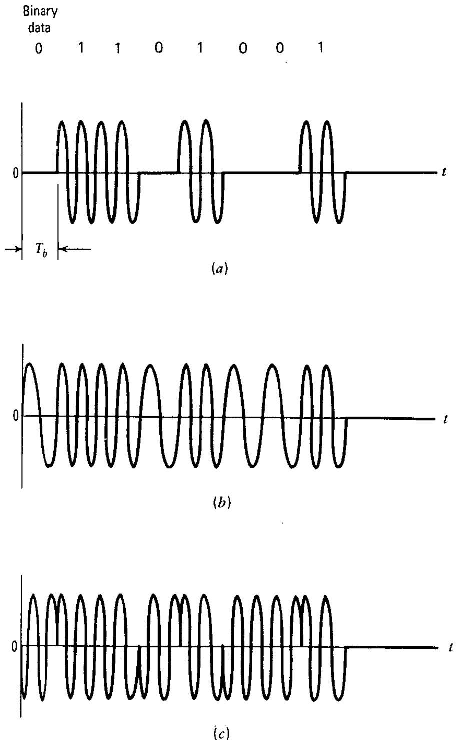


Figure 4.6: Transmitted waveforms for common digital modulation schemes: (a) amplitude-shift keying, (b) frequency-shift keying, and (c) phase-shift keying [Haykin, Fig. 6.1]

out-of-band noise, and the predetection signal (i.e., before the multiplier) is

$$\begin{aligned} x(t) &= s(t) + n(t) && (4.36) \\ &= [A(t) + n_I(t)]\cos(2\pi f_c t) - n_Q(t)\sin(2\pi f_c t) && (4.37) \end{aligned}$$

where the noise signal $n(t)$ has a double-sided white PSD of $N_o/2$ over the bandwidth $2W$. Since it is bandpass noise, we have used the bandpass representation (2.41) developed in Chapter 2.

After multiplication by the local oscillator, $2\cos(2\pi f_c t)$, the received signal is:

$$\begin{aligned} y(t) &= [A(t) + n_I(t)]\,2\cos^2(2\pi f_c t) - n_Q(t)2\sin(2\pi f_c t)\cos(2\pi f_c t) && (4.38) \\ &= [A(t) + n_I(t)]\,(1 + \cos(4\pi f_c t)) - n_Q(t)\sin(4\pi f_c t) && (4.39) \end{aligned}$$

where the last line follows from the trigonometric identities $2\cos^2 x = 1 + \cos 2x$, and $2\sin x\cos x = \sin 2x$.

After low-pass filtering, this becomes

$$\tilde{y}(t) = A(t) + n_I(t) \qquad (4.40)$$

Since the in-phase noise component $n_I(t)$ has the same variance as the original bandpass noise $n(t)$, it follows that the received signal (4.40) is identical to the received signal (4.10) for baseband digital transmission considered in the previous section. Thus, the sample values of $\tilde{y}(t)$ will have PDFs that are identical to those shown in Fig. 4.4. As in the baseband transmission case, the

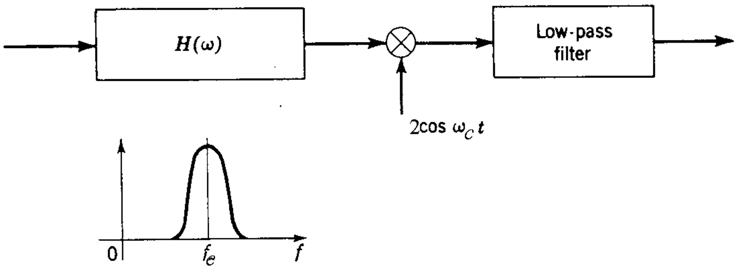


Figure 4.7: Model of a synchronous detector. [Schwartz, Fig. 5-37]

decision threshold will be set at $A/2$. Thus, we conclude that at the receiver, the statistics of the receiver signal are identical to those of a baseband system, and thus, the probability of error for ASK is the same as (4.32), i.e.,

$$P_{e,\text{ASK}} = \frac{1}{2}\,\text{erfc}\left(\frac{A}{\sigma 2\sqrt{2}}\right) \tag{4.41}$$

4.5.3 PSK

For a PSK system, we can write the transmitted signal as

$$s(t) = A(t)\cos(2\pi f_c t), \quad A(t) \in \{-A, A\} \tag{4.42}$$

The detector is again shown in Fig. 4.7, and after multiplication by the local carrier and low-pass filtering, the signal at the receiver output is

$$\tilde{y}(t) = A(t) + n_I(t) \tag{4.43}$$

where again $n_I(t)$ has the same variance as the input bandpass noise $n(t)$. In this case, however, the PDFs for sample values of $\tilde{y}(t)$ are as shown in Fig. 4.8. The threshold level would be set at 0 volts in this case, and the conditional error probabilities are

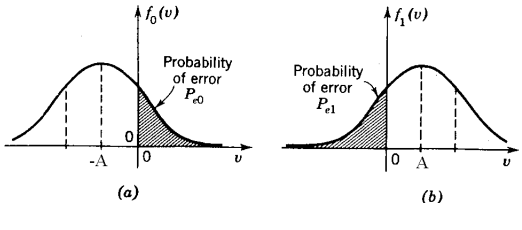


Figure 4.8: Probability density functions for PSK in noise: (a) symbol 0 transmitted, and (b) symbol 1 transmitted. [Schwartz, Fig. 5-6]

$$P_{e0} = \int_0^{\infty} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(n+A)^2}{2\sigma^2}\right) dn \tag{4.44}$$

$$P_{e1} = \int_{-\infty}^{0} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(n-A)^2}{2\sigma^2}\right) dn \tag{4.45}$$

Because of symmetry, we have $P_{e0} = P_{e1}$. We also note that each of these is equivalent to

$$P_{e,\text{PSK}} = \int_A^\infty \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{n^2}{2\sigma^2}\right) dn \tag{4.46}$$

With the change of variable $z = n/(\sigma\sqrt{2})$, we find that

$$P_{e,\text{PSK}} = \frac{1}{2} \,\text{erfc}\left(\frac{A}{\sigma\sqrt{2}}\right) \tag{4.47}$$

4.5.4 FSK

In a FSK system, the binary symbols are represented by

$$s_0(t) = A\cos(2\pi f_0 t), \quad \text{if symbol 0 is transmitted} \tag{4.48}$$
$$s_1(t) = A\cos(2\pi f_1 t), \quad \text{if symbol 1 is transmitted} \tag{4.49}$$

This requires two sets of synchronous detectors as shown in Fig. 4.9, one operating at a frequency f_0 and the other at f_1. Each branch of this detector is basically an ASK detector, and the output of the LPF on each branch is A plus noise if the symbol is present, and noise only if it is not. Denote the noise output of the top branch as $n_0(t)$, and that of the bottom branch as $n_1(t)$, where each of these noise terms has identical statistics to $n(t)$. If a symbol 1 was transmitted, the output of the summation is

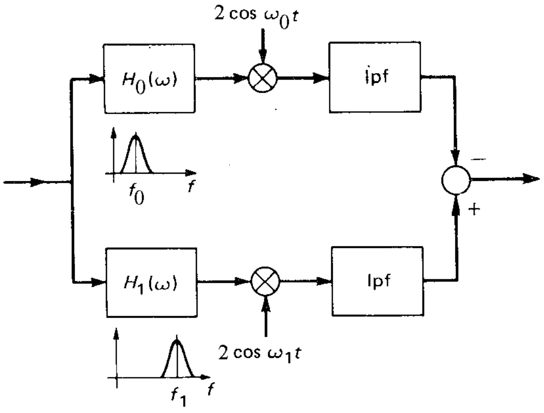


Figure 4.9: Synchronous detector for FSK. [Schwartz, Fig. 5-38]

$$y_1(t) = A + [n_1(t) - n_0(t)] \tag{4.50}$$

whereas, if a symbol 0 was transmitted, the output of the summation is

$$y_0(t) = -A + [n_1(t) - n_0(t)] \tag{4.51}$$

As in PSK, the threshold level would be set at 0 volts. The difference from PSK, however, is that the noise term is now $n_1(t) - n_0(t)$. If the noises in the two channels are independent (true if the system input noise is white and the bandpass filters $H_0(\omega)$ and $H_1(\omega)$ do not overlap), then the variances add (as shown in the example below). Hence, the noise has effectively doubled. The probability of error for FSK can be easily found by replacing σ^2 in (4.47) by $2\sigma^2$, giving

$$P_{e,\text{FSK}} = \frac{1}{2}\,\text{erfc}\left(\frac{A}{2\sigma}\right) \tag{4.52}$$

Example 4.2 – *Noise variance for FSK*

Let x_1 and x_2 be zero-mean independent random variables with variances σ_1^2 and σ_2^2, respectively. Consider $y = x_1 - x_2$. By definition, the variance of y is

$$\begin{aligned} \sigma_y^2 &= E\{y^2\} - E^2\{y\} \\ &= E\{(x_1 - x_2)^2\} \\ &= E\{x_1^2 - 2x_1x_2 + x_2^2\} \end{aligned}$$

For independent variables, $E\{x_1x_2\} = E\{x_1\}E\{x_2\} = 0$ for zero-mean random variables. So

$$\begin{aligned} \sigma_y^2 &= E\{x_1^2\} + E\{x_2^2\} \\ &= \sigma_1^2 + \sigma_2^2. \end{aligned}$$

□

4.5.5 Discussion

A comparison of the error probabilities for the various digital systems is shown in Fig. 4.10. This reveals that for the same error probability, the signal amplitude in PSK can be reduced by 6 dB (i.e., a factor of 2 reduction in amplitude) compared with a baseband or ASK system, and the signal amplitude in FSK can be reduced by 3 dB (i.e., a factor of $\sqrt{2}$ reduction in amplitude).

References

- Lathi, sections 6.1, 6.2, 7.6, 7.8, 13.2, 13.3
- Couch, sections 3-3, 7-3
- Haykin, sections 3.6, 4.3

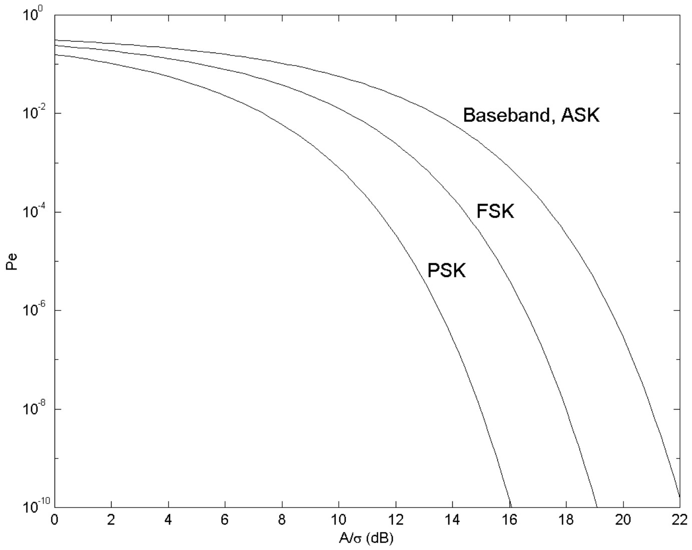


Figure 4.10: Noise performance of digital communication systems in Gaussian noise.

Chapter 5

Information Theory and Coding

5.1 Background

In 1948, Claude E. Shannon laid the foundation for a whole new discipline with the publication of his seminal paper “A Mathematical Theory of Communication”.[a] This paper, and its companion paper the following year, “Communication in the Presence of Noise”,[b] created the field of *Information Theory* , which made it possible to determine the theoretical limit of any channel’s information carrying capacity. Information Theory also made possible the development of digital systems, which handle information - voice, data video - in streams of coded pulses. Without Information Theory, the Web would not exist. In this course, only the basics of information theory are given. Because of its paramount importance in modern information science and technology, information theory is a major research area throughout the world. Students interested in this exciting area should take the course Information Theory in the fourth year.

In the previous chapters we have seen that the performance of communication systems is limited by the available signal power, the inevitable background noise, and the limited bandwidth available for transmission. We found that some systems perform better than others. This naturally leads to a fundamental question: Is there a system that performs the best? This question is perhaps stated best by Taub & Schilling, who ask “what would be the characteristics of an *ideal system*, [one that] is not limited by our engineering ingenuity and inventiveness but limited rather only by the fundamental nature of the physical universe”. This is the kind of question that drove Shannon to establish the results of information theory. In this chapter we will look at some of these results, and then compare the performance of real communication systems with that of the yet-to-be-defined ideal system.

You will find that most of the definitions in this chapter are in terms of digital sources. The definitions for continuous sources, although covered in Shannon’s original paper, are not presented here because the mathematics becomes complicated and the physical meanings are more difficult to interpret. One does not need to be too concerned about the continuous case, however, since through PCM any analog source can be approximated by a digital source with as much accuracy as required. If all this sounds like a side-step of the issue, in the final section of this chapter we will look at what information theory has to say about analog modulation and compare the results with those obtained in Chapter 3.

[a] *Bell Syst. Tech. J.*, vol. 27, pp. 379–423, 623–656, July and Oct. 1948

[b] *Proc. IRE*, vol. 37, pp. 10–21, Jan. 1949

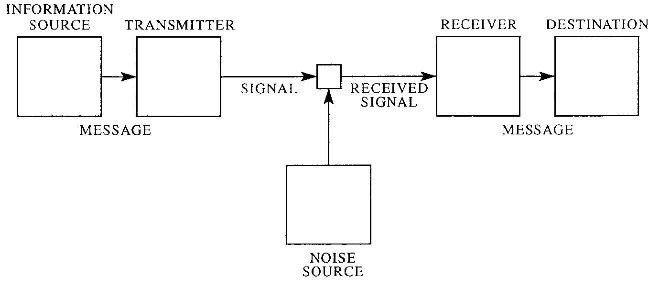


Figure 5.1: Shannon's model of a communication system. [Shannon(1948) Fig. 1]

5.2 Concept of Information

The function of any communication system is to convey information. There is an information source that produces the information, and the communication system is designed to transmit the output of the source from transmitter to receiver. In radio broadcasting, for example, the source might be a music or speech signal, in TV it is a video signal, etc. In order to perform an analysis of communication systems, we need a quantitative measure of the information that is output by an information source.

An intuitive notion of information refers to any new knowledge about something. Messages containing knowledge of a high probability of occurrence (i.e., those conveying little uncertainty in outcome) convey relatively little information. In contrast, messages containing knowledge with low probability of occurrence convey relatively large amounts of information. Thus, a reasonable measure of the information that is output by an information source should be a decreasing function of the probability of that particular output. Also, a small change in the probability of a certain output should not change the information delivered by that output by a large amount. Such considerations lead Hartley[c] to define the *amount of information* in a particular symbol s as

$$I(s) = \log \frac{1}{p} \tag{5.1}$$

where p is the probability of occurrence of the symbol s.

This definition has a number of important properties. When $p = 1$, $I(s) = 0$, i.e., a symbol that is certain to occur contains no information. For $0 \leq p \leq 1$, $0 \leq I(s) \leq \infty$, i.e., the information measure is monotonic and real-valued. Finally, if $p = p_1 \times p_2$, $I(s) = I(p_1) + I(p_2)$, i.e., information is additive for statistically independent events.

In its original form, Hartley's definition of information used a base-10 logarithm. However, in communications it has become standard to use $\log_2$, and to give information the units of *bits* (even though it is strictly dimensionless). Thus, if we have 2 symbols with equal probability $p_1 = p_2 = 1/2$, each symbol represents $I(s) = \log_2(1/0.5) = 1$ bit of information.

We will therefore use

$$I(s) = \log_2 \frac{1}{p} \tag{5.2}$$

as the definition of information content of a particular symbol s having probability of occurrence p.

[c] R.V.L. Hartley, "Transmission of information", *Bell Syst. Tech. J.*, vol. 7, pp. 535-563, 1928.

Note that $\log_2$ can be easily calculated as

$$\log_2 x = \frac{\log_{10} x}{\log_{10} 2} = 3.32 \log_{10} x \tag{5.3}$$

5.3 Source Entropy

In general, the average information associated with the output of an information source is of interest, rather than the information associated with a particular single output. This is especially important since the output of the information source occurs at random. Here we define a suitable measure for this average information.

Suppose we have an information source emitting a sequence of symbols from a finite alphabet

$$\mathcal{S} = \{s_1, s_2, \ldots, s_K\} \tag{5.4}$$

If we assume that successive symbols are statistically independent, then this is referred to as a *zero-memory source* or a *discrete memoryless source*. Further assume that each symbol has a probability of occurrence

$$p_k, k = 1, \ldots, K, \quad \text{such that} \sum_{k=1}^{K} p_k = 1. \tag{5.5}$$

If we are told that the particular symbol s_k has occurred, then, by definition, we have received

$$I(s_k) = \log_2 \frac{1}{p_k} = -\log_2 p_k \tag{5.6}$$

bits of information. But s_k occurs at random, so the expected (or mean) value of $I(s_k)$ over the source alphabet is

$$E\{I(s_k)\} = \sum_{k=1}^{K} p_k I(s_k) = -\sum_{k=1}^{K} p_k \log_2 p_k.$$

Let us define the *source entropy* as the average amount of information per source symbol:

$$H(\mathcal{S}) = -\sum_{k=1}^{K} p_k \log_2 p_k \tag{5.7}$$

and give it the units of bits / symbol. The significance of entropy is that, although one cannot say which symbol the source will produce next, on the average we expect to get $H(\mathcal{S})$ bits of information per symbol. Note that entropy in thermodynamics is a measure of disorder and randomness, and this agrees somewhat with the information theory concept of entropy as a measure of uncertainty.

Example 5.1 – *Entropy of a Binary Source*

Consider a binary source for which symbol s_1 occurs with probability p_1, and symbol s_0 occurs with probability $p_0 = 1 - p_1$. From (5.7) the entropy is

$$\begin{aligned} H(\mathcal{S}) &= -p_0 \log_2 p_0 - p_1 \log_2 p_1 \\ &= -(1-p_1)\log_2(1-p_1) - p_1 \log_2 p_1. \end{aligned}$$

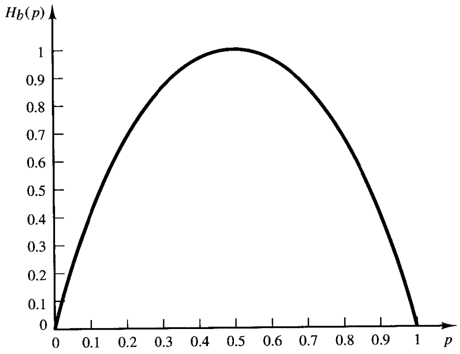


Figure 5.2: Entropy function of a binary source.[Proakis & Salehi, Fig 4.3]

This is shown in Fig. 5.2 as a function of p_1. Notice that the entropy is zero if either $p_1 = 0$ (and hence, $p_0 = 1$) or $p_1 = 1$. This is intuitively correct, since if either symbol is certain to occur then its transmission conveys zero information. Also notice that the maximum entropy occurs when $p_1 = p_0 = 1/2$. This is when either symbol is equally likely to occur, and thus, there is the maximum uncertainty.

□

Example 5.2 – *A three-symbol alphabet*

Consider a source that produces one of three possible symbols, A, B, or C, with respective probabilities 0.7, 0.2, and 0.1. The entropy (5.7) is

$$\begin{aligned} H(\mathcal{S}) &= -0.7\log_2(0.7) - 0.2\log_2(0.2) - 0.1\log_2(0.1) \\ &= 0.7 \times 0.515 + 0.2 \times 2.322 + 0.1 \times 3.322 \\ &= 1.157 \text{ bits/symbol} \end{aligned}$$

For a binary system, the most straightforward way to encode these symbols is

$$\begin{aligned} A &= 00 \\ B &= 01 \\ C &= 10 \end{aligned}$$

or permutations thereof. This would require 2 bits/symbol. However, the entropy calculation predicts that the average amount of information is only 1.157 bits per symbol.

□

In the second example above, a naive coding scheme resulted in a requirement to transmit 2 bits for every symbol, whereas the average information content of these symbols was just over 1 bit per symbol. The number of bits that one needs to transmit for each symbol clearly limits the total number of symbols (and thus, the amount of information) that can be transmitted in a given time. Thus, one would like to reduce the number of bits to be transmitted as much as possible. This raises some important questions. First, what is the minimum number of bits that are required to transmit a particular symbol? And second, how can we encode symbols such that we achieve (or at least come arbitrarily close to) this limit? These questions are addressed in the following section.

5.4 Source Coding

5.4.1 Background

A model of a communication system is shown in Fig. 5.3. We have already established that the function of this system is to transmit information reliably from the source to the destination (denoted "user" in this figure). The information source generates a sequence of symbols which are to convey a specific message to the user. These source symbols are then taken by the source encoder which assigns codewords to the source symbols in order to reduce the number of symbols that are actually transmitted to the user. The channel encoder then performs further encoding by using mechanisms that allow the receiver to correct errors caused by channel noise. The outputs of the channel encoder are then modulated and transmitted to the receiver, where the reverse of these operations are performed to recover the original information message.

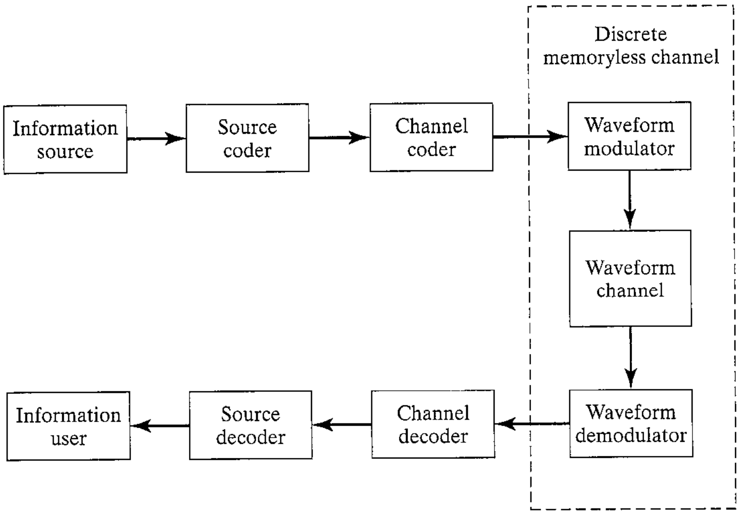


Figure 5.3: A coding model of a communication system. [Ziemer & Peterson, Fig. 6-1]

It is important to be clear about the difference between source encoding and channel encoding. Source encoding is concerned with minimizing the actual number of source bits that are transmitted to the user. Channel encoding is concerned with introducing redundant bits to enable the receiver to detect and possibly correct errors that are introduced by the channel. A simple example of channel coding is a parity check bit, which is added to a group of data bits so that the number of 1's within the group is either even or odd—this permits the detection of a single error within the group at the receiver. This is one example of a block code, in which k symbols are mapped to $n > k$ symbols by the channel encoder, with the purpose of providing the receiver with the ability to detect and possibly correct errors. Other block codes that you may come across are Reed-Solomon, Hamming, and Golay codes, just to name a few. Another class of channel codes are convolutional codes, which are generated by shift registers and exclusive-OR logic gates. These codes are detected by a device known as a Viterbi decoder. As one can imagine, the topic of channel coding is enormous. The aim here has been simply to identify that it is an important component of a communication system, and to alert you to some of the concepts that you may come across in the future.

5.4.2 Average Codeword Length

Returning to the problem of source coding, recall that this topic arose from a desire to reduce to the minimum possible the average number of data bits per unit time that must be transmitted over the channel. A basic premise of source coding is that if symbols are not equally probably, coding efficiency can be increased by using variable-length code words. Morse code (dating back to the 1800's and used for telegraphy), is an example of this idea. Letters that occur frequently are assigned short code words, and letters that occur infrequently are assigned long code words (e.g., 'e' is assigned dot, and 'z' is assigned dash-dash-dot-dot).

Let ℓ_k be the number of bits used to code the kth symbol in a source alphabet with K symbols. Further, let the probability of occurrence of this symbol be p_k. Define the *average codeword length* as

$$\bar{L} = \sum_{k=1}^{K} p_k \ell_k \tag{5.8}$$

This represents the average number of bits per symbol in the source alphabet.

The first question to be addressed is: *What is the minimum codeword length for a particular alphabet of source symbols?*

First, let us consider a system with two symbols that are equally likely. One cannot do better than to encode them with one bit, i.e., 0 or 1. For four equally-likely symbols, one needs two bits; for 8 symbols one needs 3 bits, etc. In general, if there are n equally-likely symbols, each with probability of occurrence of $p = 1/n$, then one needs $L = \log_2(1/p) = \log_2 n$ bits to represent the symbols.

Now consider an alphabet $\mathcal{S} = \{s_1, \ldots, s_K\}$ with respective probabilities p_k, $k = 1, \ldots, K$. During a long period of transmission in which N symbols have been generated (where N is very large), there will be Np_1 occurrences of s_1, Np_2 occurrences of symbol s_2, etc. If these symbols are produced by a discrete memoryless source (so that all symbols are independent), the probability of occurrence of a typical sequence $\mathcal{S}_N$, will be

$$p(\mathcal{S}_N) = {p_1}^{Np_1} \times {p_2}^{Np_2} \times \ldots \times {p_K}^{Np_K}$$

Since any particular sequence of N symbols is equally likely, the number of bits required to represent a typical sequence $\mathcal{S}_N$ is

$$\begin{aligned} L_N &= \log_2 \frac{1}{p(\mathcal{S}_N)} = -\log_2({p_1}^{Np_1} \times \ldots \times {p_K}^{Np_K}) \\ &= -Np_1 \log_2 p_1 - Np_2 \log_2 p_2 - \ldots - Np_K \log_2 p_K \\ &= -N \sum_{k=1}^{K} p_k \log_2 p_k = NH(\mathcal{S}). \end{aligned}$$

This is the number of symbols required to encode a sequence of N symbols, so the average length for one symbol is

$$\bar{L} = \frac{L_N}{N} = H(\mathcal{S}) \quad \text{bits/symbol.}$$

This result leads us to the first important theorem of information theory.

Theorem 5.1 (Source Coding Theorem)
Given a discrete memoryless source of entropy $H(\mathcal{S})$, the average codeword length $\bar{L}$ for any source coding scheme is bounded as

$$\bar{L} \geq H(\mathcal{S}).$$

This theorem has answered our first question by providing a lower bound on the number of bits required to represent a particular source alphabet. A second question immediately arises as to how one can design an efficient source coding algorithm.

5.4.3 Huffman Coding

Here we will look at one particular method of source coding, known as Huffman coding.[d] This technique is important, since the Huffman coding procedure yields the shortest average codeword length.

The idea in Huffman coding is to choose codeword lengths such that more-probable sequences have shorter codewords. A flowchart of the algorithm is given in Fig. 5.4, and is best illustrated with an example.[e]

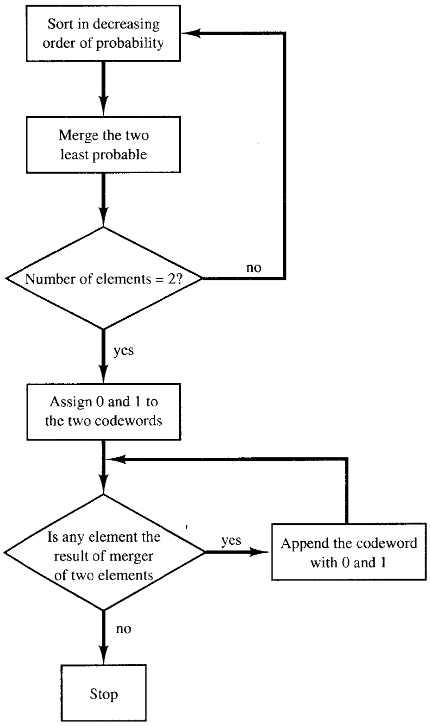

Sort in decreasing order of probability
Merge the two least probable
Number of elements = 2?
no
yes
Assign 0 and 1 to the two codewords
Is any element the result of merger of two elements
yes
Append the codeword with 0 and 1
no
Stop

Figure 5.4: Huffman coding procedure. [Proakis & Salehi, Fig. 4.5]

[d] D.A. Huffman, "A method for the construction of minimum redundancy codes", *Proc. IRE*, vol. 40, pp. 1098-1101, Sept. 1962

[e] See Haykin pp.578–579 for a different example.

Example 5.3 – *Huffman coding for a three-symbol alphabet*

Consider a source that produces one of three possible symbols, A, B, or C, with respective probabilities 0.7, 0.2, and 0.1. This is the same alphabet considered in Example 5.2. The Huffman procedure is illustrated in Fig. 5.5, and results in the following codewords:

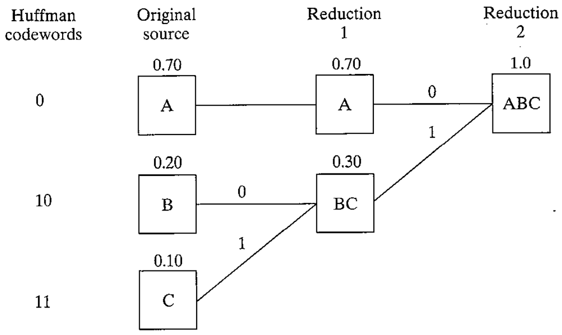


Figure 5.5: Example of the Huffman source coding procedure. [Ziemer & Peterson, Fig. 6-2]

$$
\begin{aligned}
A &= 0 \\
B &= 10 \\
C &= 11
\end{aligned}
$$

The average codeword length is calculated from (5.8) to be

$$\bar{L} = (1 \times 0.7) + (2 \times 0.2) + (2 \times 0.1) = 1.3$$

This is considerably better than the naive coding of Example 5.2 which resulted in 2 bits per symbol. However, it is still short of the entropy $H(\mathcal{S}) = 1.157$ bits per symbol.

□

To reduce the average codeword length further, symbols can be Huffman coded in pairs (or triples, quadruples, etc.) rather than one at a time. Grouping the source symbols in pairs, and treating each group as a new source symbol is referred to as the second extension of the source. Using groups of three is the third extension, and so on. For the example source alphabet, coding the second extension results in an average codeword length of $\bar{L} = 1.165$ bits per symbol, which is now appreciably closer to the lower bound $H(\mathcal{S})$. Continuing with the third extension would bring us still closer to the lower bound, and it is in this sense that Huffman coding is said to be optimal.

There is, however, one major problem with Huffman codes: the procedure relies strongly on the source statistics which must be known in advance. Before closing this section on source coding, we will have a brief look at one *universal source coding algorithm*. Such algorithms are a class of coding procedures that do not require *a priori* knowledge of the source symbol statistics.

5.4.4 Lempel-Ziv Algorithm

This algorithm is named after its inventors[f] and is widely used in computer file compression such as the Unix `gzip` and `compress` functions. It is a variable- to fixed-length coding scheme, in that any sequence of source symbols is uniquely parsed into phrases of varying length, and each phrase is then coded using equal length codewords. It basically works by identifying phrases of the smallest length that have not appeared so far, and maintaining these phrases in a dictionary. When a new phrase is identified it is encoded as the concatenation of the previous phrase and the new source output. It is not necessary to understand this procedure in detail,[g] rather the intention is to provide a flavour of the type of algorithms that are used in practice for source coding.

5.5 Channel Capacity

Consider a source that generates symbols at a rate of r symbols per second. We have already shown that the average number of bits of information per symbol is the entropy (5.7). Defining the *information rate*, R, as the average number of bits of information per second, we find

$$R = rH \tag{5.9}$$

According to this definition, one should be able to transmit information at an arbitrarily high rate, simply by increasing the source symbol rate r. If the symbols are transmitted over a noisy channel, however, one will obtain bit errors at the receiver (as we saw in Chapter 4).

Define the *channel capacity*, C, as the maximum rate of *reliable* information transmission over a *noisy* channel. In other words, it is the maximum rate of information transfer with an arbitrarily small probability of error. Shannon proved the following fundamental theory of communications regarding channel capacity.

Theorem 5.2 (Channel Capacity Theorem)
If $R \leq C$, then there exists a coding scheme such that the output of the source can be transmitted over a noisy channel with an arbitrarily small probability of error. Conversely, it is not possible to transmit messages without error if $R > C$.

You should appreciate that this is a surprising result. In Chapter 4 we saw that the probability density of Gaussian noise extends to infinity. Thus, we would expect that there will be some times, however infrequent they may be, when the noise must override the signal and produce an error. However, Theorem 5.2 says that this need not cause errors, and that it is possible to receive messages without error even over noisy channels. What this theorem says is that the basic limitation due to noise in a communication channel is not on the *reliability* of communication, but rather, on the *speed* of communication.

The following complementary theorem tells us what this maximum rate of reliable information transfer is for a Gaussian noise channel.

[f] J. Ziv and A.Lempel, "A universal algorithm for sequential data compression", *IEEE Trans. Infor. Theory*, vol. 23, pp.337-343, May 1977

[g] For the interested reader, an example is given in Proakis and Salehi pp.236–237, or Haykin pp.580–581.

Theorem 5.3 (Shannon Theorem)
For an additive white Gaussian noise channel, the channel capacity is

$$C = B \log_2 \left(1 + \frac{S}{N}\right)$$

where B is the bandwidth of the channel, S is the average signal power at the receiver, and N is the average noise power at the receiver.

Although it is strictly applicable only to additive white Gaussian noise channels, Theorem 5.3 is of fundamental importance.[h] One finds that in general, most physical channels are at least approximately Gaussian. Also, it turns out that the results obtained for a Gaussian channel often provide a lower bound on the performance of a system operating over a non-Gaussian channel.

According to this theorem, theoretically we can communicate error free up to C bits per second. Although telling us the upper theoretical limit of error-free communication, Theorems 5.2 and 5.3 tell us nothing about how to achieve this rate. This was one of the problems which had persisted to mock information theorists for almost half a century since Shannon's original paper in 1948. Despite an enormous amount of effort spent since that time in quest of this Holy Grail of information theory, a *deterministic* method of generating the codes promised by Shannon was not found until 1990's. Recently, certain error-correction channel coding schemes [i] have been developed that allow a system to come extremely close (within a fraction of a dB) to the Shannon limit.

5.6 Implications to the Performance of Communication Systems

The goal of analog communications systems is to reproduce signals reasonably faithfully, with a minimum of noise and distortion. This is not quite the same thing as reliable information transfer in information theory. Information theory does, however, have something to say about analog transmission. Specifically, it (a) tells us the best SNR that can be obtained with given channel parameters, (b) tells us the minimum power required to achieved a specific SNR as a function of bandwidth, and (c) indicates the optimum possible exchange of bandwidth for power.

In analog systems one might define the optimum communication system as that which achieves the largest signal-to-noise ratio at the receiver output, subject to certain design constraints (such as channel bandwidth and transmitted power). Is it possible to design a system with infinite signal-to-noise ratio at the output when transmission is over a noisy channel? As one might expect, the answer is no. However, Theorem 5.3 does allow us to derive the noise performance of the theoretical optimum analog system.

The general model of a communication system is shown in Fig. 5.6. A baseband message $m(t)$ of bandwidth W, is modulated to form a bandpass signal $s(t)$ having bandwidth B, and transmitted over a channel with additive white Gaussian noise. The received signal $x(t) = s(t) + n(t)$ is filtered to a bandwidth of B, the transmission bandwidth. According to Theorem 5.3, the maximum rate at which information may arrive at the receiver is

$$C_{in} = B \log_2(1 + \text{SNR}_{in}) \tag{5.10}$$

[h] For the interested reader, an overview of the derivation of this theorem (involving the volumes of n-dimensional hyperspheres) can be found in Proakis & Salehi, pp.733-736.

[i] C. Berrou *et al*, "Near Shannon limit error-correction coding and decoding: turbo codes", *Proc. ICC*, Geneva, May 1993.

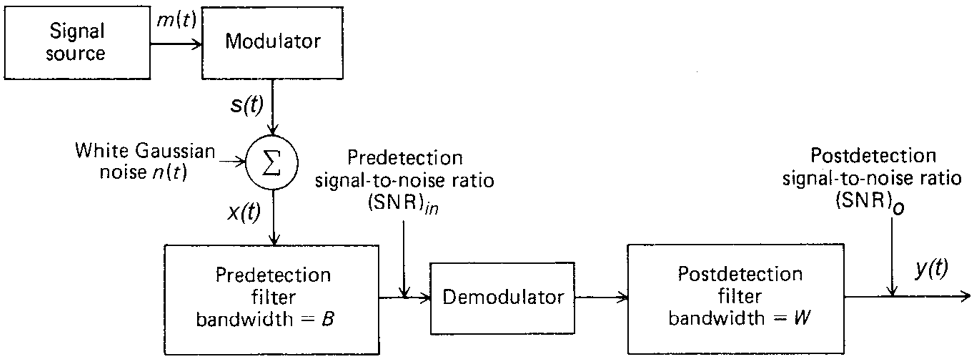

Signal source
$m(t)$
Modulator
$s(t)$
White Gaussian noise $n(t)$
Σ
$x(t)$
Predetection signal-to-noise ratio $(\mathrm{SNR})_{in}$
Postdetection signal-to-noise ratio $(\mathrm{SNR})_o$
Predetection filter bandwidth = B
Demodulator
Postdetection filter bandwidth = W
$y(t)$

Figure 5.6: General model of a modulated communication system. [Ziemer & Tranter, Fig. 10.29]

where SNR_{in} is the predetection signal-to-noise ratio at the input to the demodulator.

After demodulation, the signal is low-pass filtered to W the message bandwidth. The maximum rate at which information can leave the receiver is

$$C_o = W \log_2(1 + \mathrm{SNR}_o) \tag{5.11}$$

where SNR_o is the SNR at the output of the postdetection filter.

Notice that we have not specified a particular modulation or demodulation scheme. We assume that the scheme is optimum in some respect. Specifically, an ideal modulation scheme will be defined as one that does not lose information in the detection process, so that

$$C_o = C_{in} \tag{5.12}$$

Substitution yields

$$\mathrm{SNR}_o = [1 + \mathrm{SNR}_{in}]^{B/W} - 1 \tag{5.13}$$

which shows that the optimum exchange of SNR for bandwidth is exponential.

We can gain further insight by looking more closely at the predetection SNR at the demodulator input. Specifically, if the channel noise has a double-sided white PSD of $N_o/2$, then the average noise power at the demodulator will be N_oB. If the transmitted power is P, then we have

$$\mathrm{SNR}_{in} = \frac{P}{N_oB} = \frac{W}{B}\frac{P}{N_oW} \tag{5.14}$$

Recognize that $P/(N_oW)$ is just the baseband SNR (3.3) from Chapter 3. Hence, the output SNR of an ideal communication system is

$$\mathrm{SNR}_o = \left(1 + \frac{W}{B}\,\mathrm{SNR}_{\text{baseband}}\right)^{B/W} - 1 \tag{5.15}$$

and is shown in Fig. 5.7 as a function of baseband SNR for various B/W ratios.

In Chapter 3 we derived the noise performance of various analog modulation schemes in terms of baseband SNR. Since we now also have the performance of the ideal receiver in terms of baseband SNR, we can relate it to the performance curves from Chapter 3.

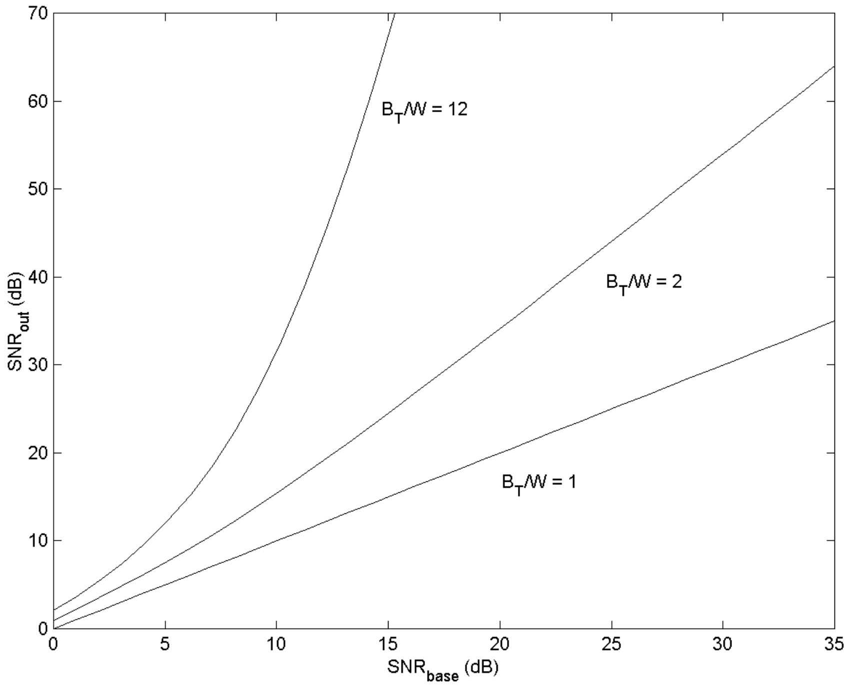


Figure 5.7: Noise performance of an ideal communication system.

Specifically, a bandwidth spreading ratio of $B/W = 1$ corresponds to both SSB and baseband. A bandwidth spreading ratio of $B/W = 2$ corresponds to DSB and AM, and $B/W = 12$ corresponds to commercial FM broadcasting. Comparing these curves with Fig. 3.11 we find that SSB and baseband systems provide noise performance identical to the ideal, whereas DSB has a worse performance because of its additional bandwidth requirements. Finally, FM systems, while providing far greater noise immunity than amplitude modulation systems, only come close to the ideal system near threshold.

5.7 Channel Coding

As we have seen, channel capacity is the ultimate limit of reliable information transmission. To approach the capacity, we need strong channel codes. From a more practical point of view, noise in the channel can corrupt the information that we wish to transmit. So it is a bad thing that should be avoided if possible. One should always bear in mind what data/information is being transmitted. A mistake in one 'bit' of a very large binary signal might be completely irrelevant if the signal is a large uncompressed image, but it might be highly important if the bit was part of a highly compressed image or if it was part of an electronic identifier from a security system. Different systems will generally require different levels of protection against errors due to noise and, consequently, there are a number of different techniques that have been developed to detect and correct different types and different numbers of errors.

We can measure the effect of noise in different ways. The most common is to specify an error probability, p. As with sources, there can be correlations (memory) between errors over several symbols or errors that vary with time or errors that cause data to be lost rather than corrupted (erasure), although these will not be considered in this course. All examples considered for this course will be for error probabilities that are symmetric, stationary and statistically independent. A typical example is the case of a symmetric binary channel with noise, shown in Fig. 5.8.

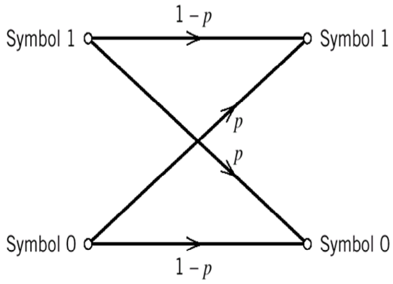


Figure 5.8: The binary symmetric channel.

If the error probability is small and the information is fairly fault tolerant, it is possible to use fairly simple methods to detect that an error has occurred:

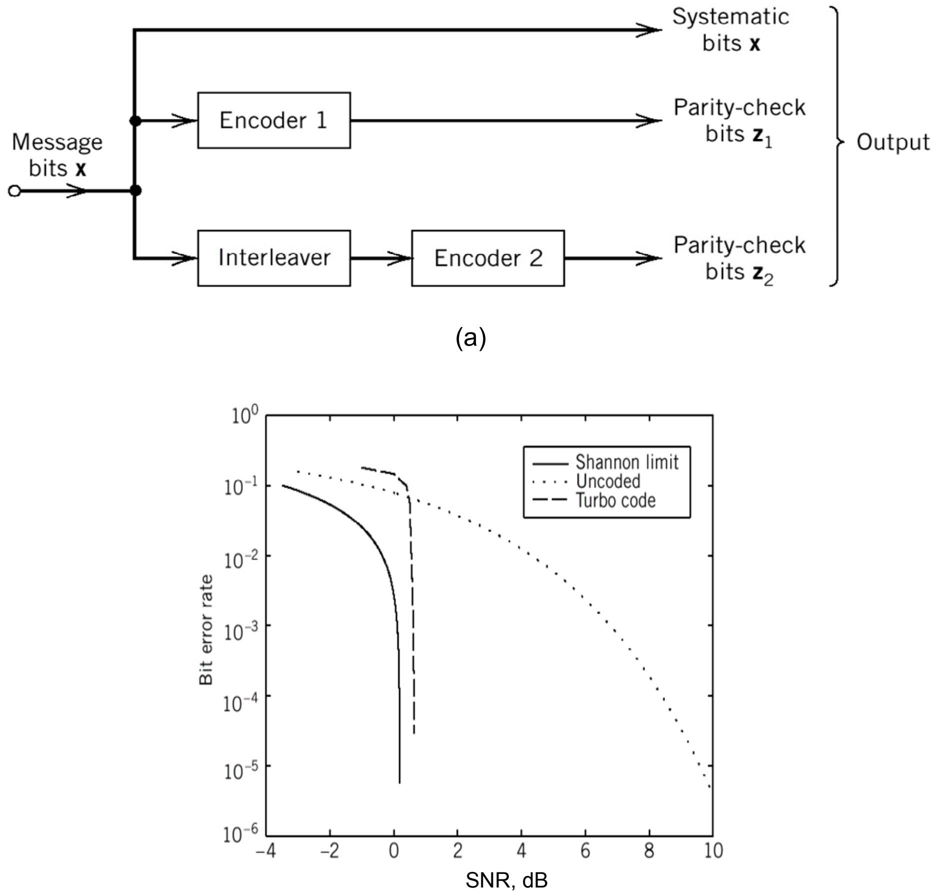
Repetition: Repeating each bit in the message is a simple method for checking whether an error has occurred. If the two symbols in an adjacent pair are different, it is likely that an error has occurred. However, this is not very efficient (efficiency is halved), and there is no way of telling whether it is the first bit of the second bit in a pair that has been corrupted. I.e. repetition provides a means for error checking, but not for error correction.

Parity bit: Another means for checking whether an error has occurred it to add a ‘parity bit’ to the end of the message. A parity bit is a single bit that corresponds to the sum of the other message bits (modulo 2). This allows any odd number of errors to be detected, but not even numbers. However, as with repetition, this technique only allows error checking, not error correction. It is more efficient than simple repetition.

5.7.1 Linear Block Codes

Although it is important to detect whether an error has occurred, it is also useful to be able to repair that error without needing for the message to be transmitted again, either as a whole or in part. An important class of codes that allow some errors to be corrected as well as detected are the linear block codes. The first error-correcting block codes were devised by Hamming around the same time as Shannon was working on the foundations of information theory (late 1940’s, early 1950’s). Hamming codes are a particular class of linear block code. The idea behind block codes is to take a series of symbols from a source, a ‘block’, and to encode these symbols into a longer string in such a way that any errors will show up because the received coded block will not be one of the recognised coded blocks. The additional step (decoding) is to arrange for the corrupted block to the associated with a valid coded block by its proximity (as measured by the ‘Hamming distance’). This additional step is important because it allows error correction as well as error detection.

Channel coding, also known as error correction coding or error control coding, is arguably the most important part of a communication system. After decades of research by mathematicians and coding theorists, capacity-approaching codes have been successfully constructed. The most famous ones of such powerful codes are turbo codes, low-density parity check codes, and polar codes. Fig. 5.9 shows the encoder diagram and the amazing error performance achieved by the turbo code.



(a)

(b)

Figure 5.9: (a) Diagram of the turbo encoder; (b) bit error rate performance.

Error correction codes may broadly be divided into two classes: block codes and convolutional codes. The theory of block codes heavily depends on mathematics, and particularly algebra. This is why block codes are also called algebraic codes. The most famous algebraic codes are Reed-Solomon codes and BCH codes. On the other hand, less mathematical convolutional codes actually find more applications in practice due to a better tradeoff between implementation complexity and performance. Students interested in this extremely important area should take the more advanced course Coding Theory in the fourth year.

In this course, we only consider linear block codes. Before looking in more detail at block codes, we need to discuss some of the mathematics that will be needed. Fortunately, we have restricted things to binary sources, so the mathematics is relatively simple. The binary alphabet $A = 0, 1$ is properly referred to as a Galois field with two elements, denoted GF(2). Since there are only two elements, and any mathematical operation should take us from one or more elements back to another element (closure), we must define mathematical operations accordingly:

Addition:

$$0 + 0 = 0, \quad 0 + 1 = 1 + 0 = 1, \quad 1 + 1 = 0;$$

Multiplication:

$$0 \cdot 1 = 0 \cdot 0 = 1 \cdot 0 = 0, \quad 1 \cdot 1 = 1.$$

This is also referred to as Boolean arithmetic or modulo 2 arithmetic. A message is built up from a number of binary fields, and forms a binary vector, rather than a larger binary number. Hence,

$$101 \neq 5, \quad 101 = \{1\}\{0\}\{1\}.$$

The Hamming weight of a binary vector, a (written as $w_H(a)$), is the number of non-zero elements that it contains. Hence,

- 001110011 has a Hamming weight of 5.
- 000000000 has a Hamming weight of 0.

The Hamming Distance between two binary vectors, a and b, is written $d_H(a, b)$, and is equal to the Hamming weight of their (Boolean) sum.

$$d_H(a, b) = w_H(a + b).$$

Hence, 01110011 and 10001011 have a Hamming distance of

$$d_H = w_H(01110011 + 10001011) = w_H(11111000) = 5.$$

A binary linear block code that takes block of k bits of source data and encodes them using n bits, is referred to as a (n, k) binary linear block code. The ratio between the number of source bits and the number of bits used in the code, $R = k/n$, is referred to as the code rate. From Shannon's channel coding theorem, the code rate must be less than the channel capacity for a code to exist that will have an arbitrarily small chance of error (even if we do not know what that code might be). The most important feature of a linear block code is that the (Boolean) sum of any code word must be another code word. This means that the set of code words forms a vector space, within which mathematical operations can be defined and performed.

To construct a linear block code we define a matrix, the generator matrix G, that converts blocks of source symbols into longer blocks corresponding to code words. G is a $k \times n$ matrix (k rows, n columns), that takes a source word or block, u (a binary vector of length k), to a code word, x (a binary vector of length n),

$$x = u \cdot G.$$

For the code words to be unique, the rows of the generator matrix must be linearly independent. That is, it should not be possible to write any of the rows as linear combinations of any of the other rows. A useful result from linear algebra is that a $k \times n$ matrix of linearly independent rows can always be written in the form,

$$G = [I_{k \times k} \vdots P_{k \times (n-k)}]$$

where $I_{k\times k}$ is the $k \times k$ identity matrix, and $P_{k\times(n-k)}$ is a $k \times (n-k)$ matrix of parity bits, called the parity check generator.

When G takes this form, the first k bits will be the original source block. Such codes are called systematic codes. Every linear code is equivalent to a systematic code in the sense that the two codes have the same set of codewords. To obtain the systematic code, one may apply elementary row operations on G, i.e., exchange two rows, or add one row to another. Any matrix G can be transformed into the systematic form so that the row space is not changed.

To determine how many errors a particular code can detect and how many errors a code can correct, we look at the minimum Hamming distance between any two code words. We have already said that the (Boolean) sum of any code word must be another code word, and we have seen that the sum of a Boolean vector with itself is a zero vector, so we can infer that the zero vector must be a code word. If we define the minimum distance between any two code words to be,

$$d_{min} = \min\{d_H(a,b), a, b \in C\} = \min\{d_H(0, a+b), a, b \in C\} = \min\{w_H(c), c \in C, c \neq 0\}$$

where C is the set of code words. The number of errors that can be detected is then

$$t \leq (d_{min} - 1),$$

since d_{min} errors can take an input code word and turn it into a different but valid code word. Less than d_{min} errors will take an input code word and turn it into a vector that is not a valid code word. The number of errors that can be corrected is simply the number of errors that can be detected divided by two and rounded down to the nearest integer,

$$t \leq \left\lfloor \frac{d_{min} - 1}{2} \right\rfloor$$

since any output vector with less than this number of errors will 'nearer' to the input code word.

Coding a block of source data is relatively simple. It is just a case of multiplying the source block vector by the generator matrix. To decode a block coded vector it is more complicated. If the generator matrix is of the form

$$G = [I_{k\times k} \vdots P_{k\times(n-k)}]$$

To check for errors, we define a new matrix, the parity check matrix, H. The parity check matrix is a $(n-k) \times n$ matrix that is defined so that it will produce a zero vector when no errors are present,

$$x \cdot H^T = 0$$

where H^T is the transpose of H. To satisfy this condition, it is sufficient to write the parity check matrix in the form

$$H = [(P_{k\times(n-k)})^T \vdots I_{(n-k)\times(n-k)}]$$

The minimum Hamming distance is equal to the smallest number of columns of H that are linearly dependent[j].

[j] See Haykin, Chapter 10 for a proof.

If the received coded block y does contain errors, then the product of the received block with the transpose of the parity check matrix will not be zero,

$$y \cdot H^T \neq 0.$$

Writing $y = x + e$ which is the sum of the original coded block x and the error e, we find that

$$y \cdot H^T = e \cdot H^T.$$

This value is referred to as the syndrome,

$$s = y \cdot H^T.$$

The syndrome is a function of the error only, and contains the information required to isolate the position (or positions) of the error (or errors). s is an $(n-k)$ row vector, taking $2^{(n-k)}-1$ possible values (the zero vector corresponding to no error). This means that it is necessary to calculate and store $2^{(n-k)} - 1$ syndromes as a look-up table to be able to pin-point the positions of the errors exactly. The problem is that this is impractical for large values of $(n - k)$.

5.7.2 Hamming Codes

Hamming codes are a class of linear block codes that can correct a single error. They are 'perfect' codes, in that they require a minimum number of additional parity bits. That is, they meet the lower limit set by the Hamming bound,

$$r = n - k = \log_2(n+1).$$

From this expression it is easy to see that the smallest Hamming codes correspond to

$$(n, k) = (7, 4), (15, 11), (31, 26), ...$$

The fact that they are 'perfect' codes does not mean that they are necessarily the best codes to use, but they do have the additional advantage that they are easy to construct and are simple to use. For all Hamming codes, $d_{min} = 3$. All (correctable) error vectors have unit Hamming weight, and the syndrome associated with an error in the ith column of the vector is the ith row of H^T. Columns of H are binary representations of $1, ..., n$.

For the $(7, 4)$ Hamming code,

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

The corresponding syndrome table is:

s	e
000	0000000
001	0000001
010	0000010
100	0000100
111	0001000
110	0010000
101	0100000
011	1000000

When a coded vector is received, the syndrome is calculated and any single error identified, and corrected by exchanging the relevant bit with the other binary value. However, problems can occur if there is more than one error.

The implicit assumption is that there is a one-to-one correspondence between the errors and the syndromes. However, this is only actually the case when the number of possible errors is less than or equal to the number of correctable errors. When this is not the case, we must use a mechanism to decide which valid code word was transmitted. This requires an algorithm to determine the output, usually based on the most likely error (minimum Hamming distance). A Hamming code has a minimum Hamming distance of 3, so we would expect it to be able to detect 2 errors and correct one. We have already seen that it can correct one error, but what about detecting two errors? The most likely error will be a single bit error in the wrong part of the code word. The syndrome has indeed detected an error, but it has associated it wrongly to a single error, and is likely to 'correct' it by 'fixing' the wrongly identified bit.

References

- Lathi, sections 15.2, 15.2, 15.6
- Couch, section 7-9
- Haykin, sections 9-10

Textbooks

- S. Haykin and M. Moher, *Communication Systems*, fifth ed., International Student Version, Wiley, 2009

- U. Madhow, *Introduction to Communication Systems*, Cambridge University Press, 2015.

- S. Haykin, *Communication Systems*, fourth ed., Wiley, New York, 2001

- B.P. Lathi, *Modern Digital and Analog Communication Systems*, third ed., Oxford University Press, 1998

- L.W. Couch II, *Digital and Analog Communication Systems*, sixth ed., Prentice-Hall, New Jersey, 2001

- J.G. Proakis and M. Salehi, *Communication Systems Engineering*, Prentice-Hall, New Jersey, 1994