Digital Communications Fundamentals

Chapter 4

Introduction

We will provide an overview of several key fundamental concepts employed in the transmission of digital data.

4.1 What Is Digital Transmission?

A digital transceiver is a system composed of a collection of both digital and analog processes that work in concert with each other in order to handle the treatment and manipulation of binary information.

The purpose of these processes is to achieve data transmission and reception across some sort of medium.

The book considers the bit to be the fundamental unit of information used by a digital communication system. (We are not gonna argue with them 🙄)

4.1 What Is Digital Transmission?

A digital transceiver is essentially responsible for the translation between a stream of digital data represented by bits and electromagnetic waveforms possessing physical characteristics that uniquely represent those bits.

Several physical characteristics of electromagnetic waveforms commonly used to represent digital data per time interval T include:

- Amplitude
- Phase
- Frequency

Some advanced mapping regimes, binary patterns can potentially be represented by two or more physical quantities (i.e. APSK).

4.1 What Is Digital Transmission?

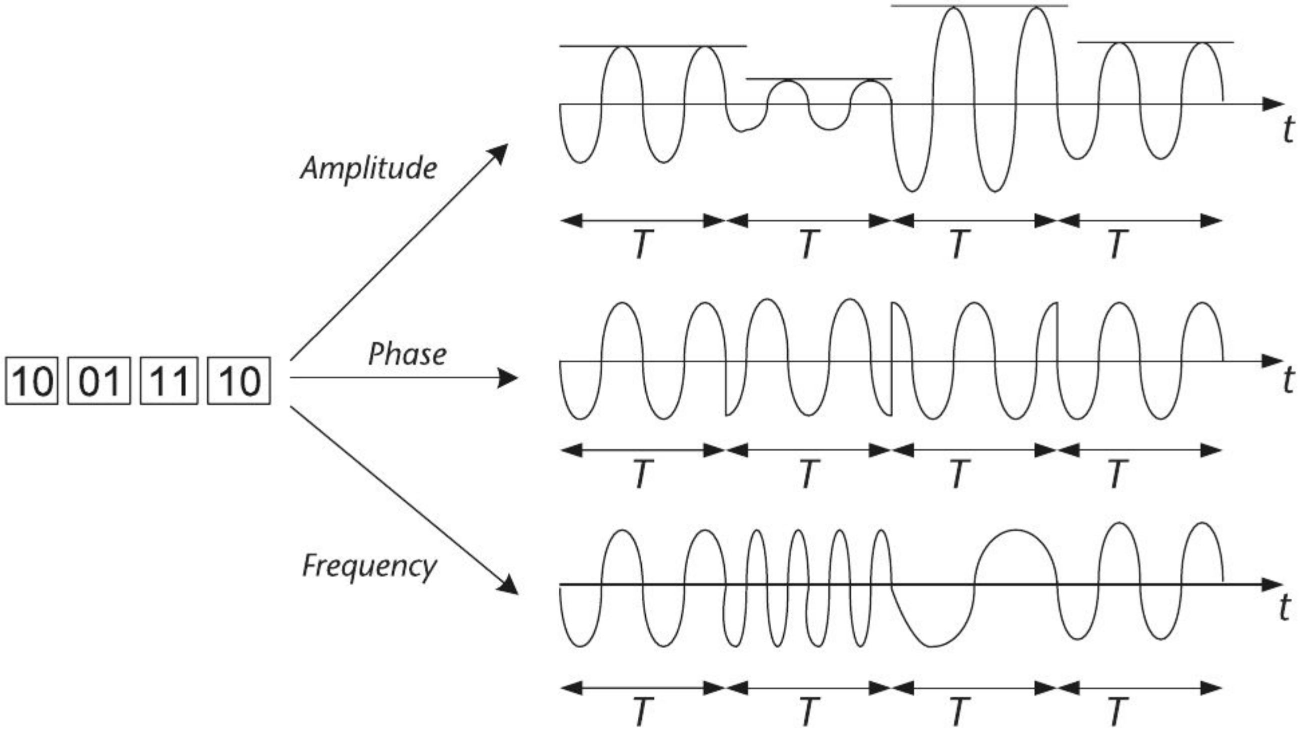

Figure 4.1 Possible mappings of binary information to EM wave properties.

4.1 What Is Digital Transmission?

Functional blocks that constitute a communication system.

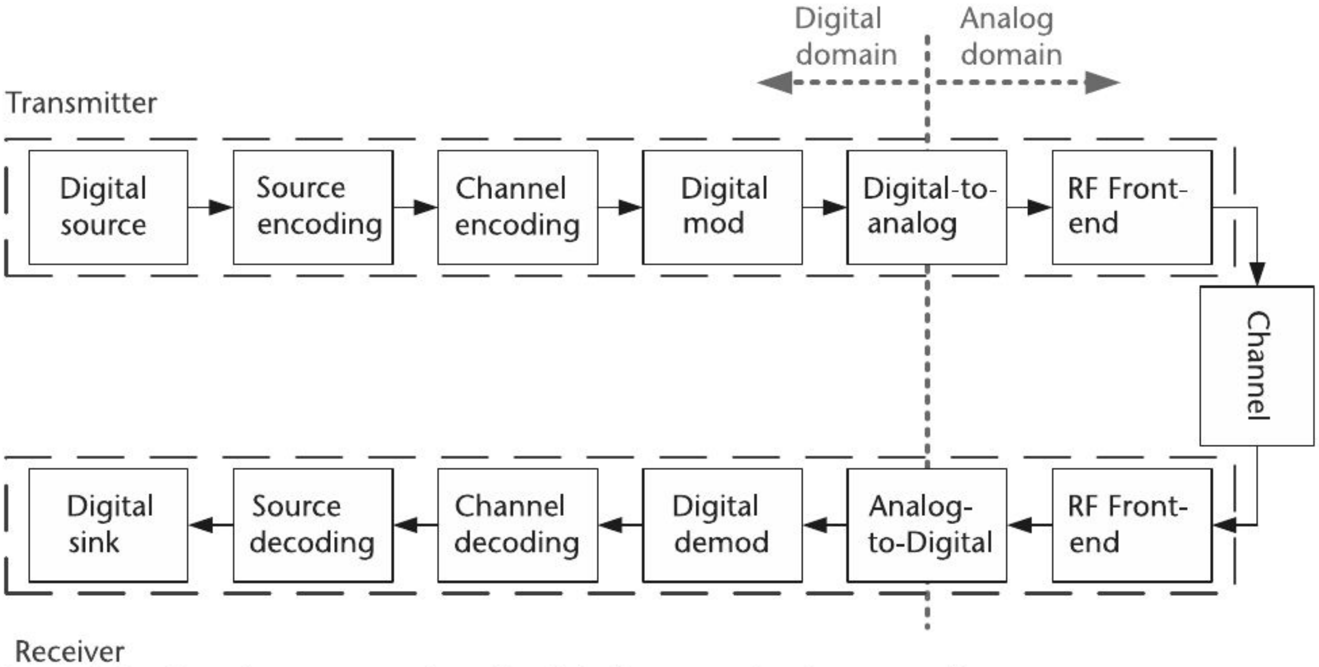

Figure 4.2 Generic representation of a digital communication transceiver.

4.1 What Is Digital Transmission?

Q: Why do we need all these blocks in our digital communication system?

A: Because the channel is not ideal! If so, the design would be trivial.

4.1 What Is Digital Transmission?

In reality a channel introduces a variety of random impairments to a digital transmission that can potentially affect the correct reception of waveforms intercepted at the receiver.

Many of these nonideal effects introduced by the channel are time-varying and thus difficult to deal with, especially if they vary rapidly in time.

4.1 What Is Digital Transmission?

The primary goal of any digital communication system is to transmit a binary message $m(t)$ and have the reconstructed version of this binary message $\hat{m}(t)$ at the output of the receiver to be equal to each other.

Our goal is to have the probability of error or BER, $Pe = P(\hat{m}(t) \neq m(t))$, as small as needed for a particular application.

4.1 What Is Digital Transmission?

Let’s look at some MATLAB code on *modulations.m* file.

4.1.1 Source Encoding

Source encoding is a mechanism designed to remove redundant information in order to facilitate more efficient communications.

Perform a mapping of source symbols $\mathbf{u}$ into uncorrelated source encoded symbols $\mathbf{v}$.

4.1.1 Source Encoding

Let’s look at some MATLAB code on *source_coding.m* file.

4.1.2 Channel Encoding

Channel encoding is designed to correct for channel transmission errors by introducing controlled redundancy into the data transmission.

Opposed to the redundancy that is removed during the source encoding process, which is random in nature, the redundancy introduced by a channel encoding is specifically designed to combat the effects of bit errors in the transmission.

4.1.2 Channel Encoding

Channel encoding operates as follows:

- Each vector of a source encoded output of length K is assigned a unique codeword of length N.
- The codeword comes from a codebook.
- The channel encoder introduces N−K = r controlled number of bits to the channel encoding process.
- The code rate of a communications system is equal to k/N

4.1.2 Channel Encoding

Hamming distance is often used to determine the effectiveness of a set of codewords contained within a codebook by evaluating the relative difference between any two codewords.

The Hamming distance $d_H(c_i, c_j)$ is equal to the number of components in which c_i and c_j are different.

When determining the effectiveness of a codebook design, we often are looking for the minimum Hamming distances between codewords

$$d_{H,\min} = \min_{c_i, c_j \in \mathbb{C},\ i \neq j} d_H(c_i, c_j).$$

4.1.2 Channel Encoding

Let's look at some MATLAB code on *channel_coding.m* file.

4.1.2 Channel Encoding

In the event that a codeword is corrupted during transmission, decoding spheres (also known as Hamming spheres) can be employed in order to make decisions on the received information

Codewords that are corrupted during transmission are mapped to the nearest eligible codeword

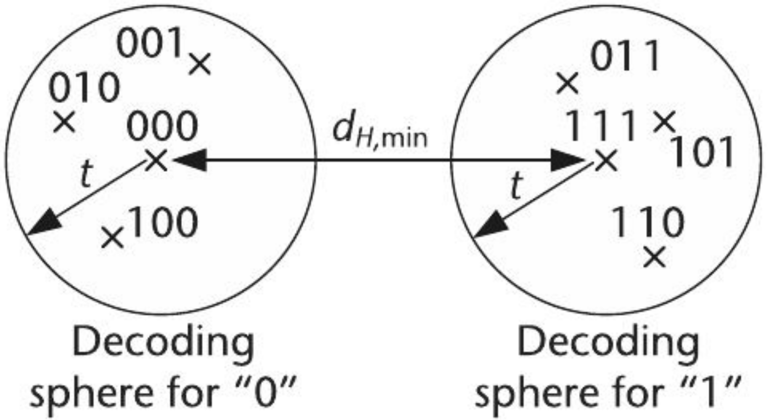

Figure 4.5 Example of decoding spheres.

4.1.2.1 Shannon’s Channel Coding Theorem

In 1949 Claude Shannon published his seminar paper that addressed the problem of establishing the upper limit of the data rate for a specific digital transceiver, entitled “Communication in the Presence of Noise”.

Shannon defined a quantitative expression that described the limit on the data rate, or capacity, of a digital transceiver in order to achieve error-free transmission.

4.1.2.1 Shannon's Channel Coding Theorem

Consider a channel with capacity C:

We transmit data at a fixed or constant code rate $R_c = K/N$

If we increase N, then we must increase K in order to keep R_c equal to a constant.

Shannon states that a code exists such that for $R_c = K/N < C$ and as $N \to \infty$, we have the probability of error $P_e \to 0$.

Therefore no such code exists that $R_c = K/N \geq C$, and C is the limit in rate for reliable communications.

4.1.2.1 Shannon's Channel Coding Theorem

Shannon derived the information capacity of the channel, which turned out to be equal to

$$C = B \log_2(1 + SNR) \qquad [\mathrm{b/s}].$$

Where B is the transmission bandwidth and SNR is the received signal-to-noise ratio.

This information capacity tells us the achievable data rate.

4.1.2.1 Shannon's Channel Coding Theorem

Shannon only provided us with the theoretical limit for the achievable capacity of a data transmission, but he does not tell us how to build a transceiver to achieve this limit. 😱

4.2 Digital Modulation

Most digital modulation techniques possess an intermediary step where collections of b bits forming a binary message m_b are mapped to a symbol, which is then used to define the physical characteristics of a continuous waveform in terms of amplitude and phase.

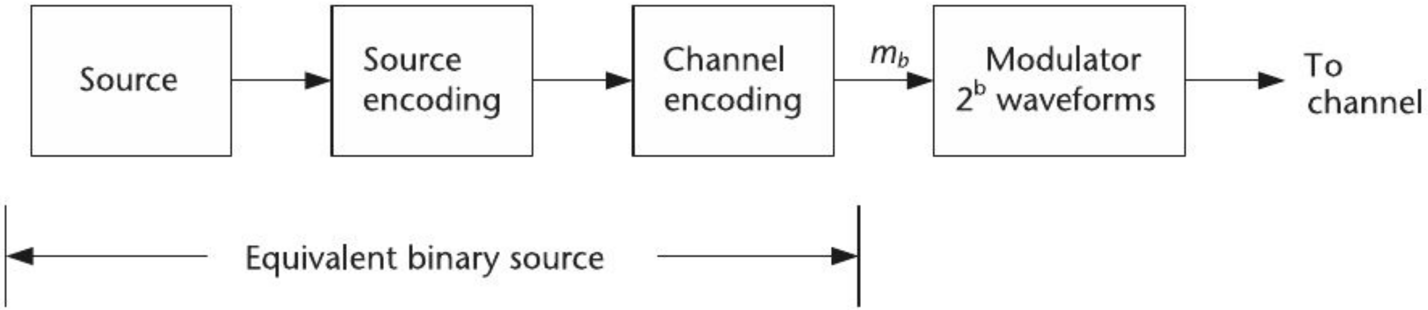

Figure 4.7 Modulation process of equivalent binary data.

4.2 Digital Modulation

For each of the possible 2^b values of m_b , we need a unique signal $s_i(t), 1 \leq i \leq 2^b$ that can then be used to modulate the continuous waveform.

There are several different families of approaches for mapping binary data into symbols that can then be used to modulate continuous waveforms.

There exist various trade-offs between these different families, including how efficiently a bit is mapped to a symbol in terms of the transmit **power expended**.

4.2.1 Power Efficiency

Energy of a Symbol $s(t)$

$$E_s = \int_0^T s^2(t)dt,$$

Where T is the period of the symbol.

Average Symbol Energy

$$\bar{E}_s = P(s_1(t)) \cdot \int_0^T s_1^2(t)dt + \cdots + P(s_M(t)) \cdot \int_0^T s_M^2(t)dt$$

where $P(s_i(t))$ is the probability that the symbol $s_i(t)$ occurs.

4.2.1 Power Efficiency

Average Energy per Bit

$$\bar{E}_b = \frac{\bar{E}_s}{b} = \frac{\bar{E}_s}{\log_2(M)}$$

Euclidean Distance Between Symbols

$$d_{ij}^2 = \int_0^T (s_i(t) - s_j(t))^2 dt = E_{\Delta s_{ij}}$$

4.2.1 Power Efficiency

Since we are often interested in the worst-case scenario we usually compute the minimum Euclidean distance

$$d_{min}^2 = \min_{s_i(t), s_j(t), i \neq j} \int_0^T (s_i(t) - s_j(t))^2 dt$$

Power Efficiency of a Signal

$$\varepsilon_p = \frac{d_{\min}^2}{\bar{E}_b}$$

4.2.2 Pulse Amplitude Modulation (PAM)

PAM is a digital modulation scheme where the message information is encoded in the amplitude of a series of signal pulses.

Demodulation of a PAM transmission is performed by detecting the amplitude level of the carrier at every symbol period.

4.2.2 Binary Pulse Amplitude Modulation (B-PAM)

The most basic form of PAM is binary PAM (B-PAM).

Modulation rule:

$$1 \rightarrow s_1(t)$$
$$0 \rightarrow s_2(t)$$

where $s_1(t)$ is the waveform $s(t)$ possessing one unique amplitude level while $s_2(t)$ is also based on the waveform $s(t)$ but possesses another unique amplitude level.

The duration of the symbols is equivalent to the duration of the bits, the bit rate for a B-PAM transmission is defined as $R_b = 1/T$ bits per second.

4.2.2 Binary Pulse Amplitude Modulation (B-PAM)

Suppose we have s(t) given by a rectangular waveform

$$s(t) = A \cdot [u(t) - u(t - T)].$$

Our modulation rule

$$1 \rightarrow s(t)$$
$$0 \rightarrow -s(t)$$

4.2.2 Binary Pulse Amplitude Modulation (B-PAM)

Energy of a Symbol $s(t)$

$$E_s = E_{-s} = A^2 T = \frac{A^2}{R_b}$$

Where $R_b = 1/T$ is the bit rate.

Average Symbol Energy

$$\bar{E}_s = E_s\{P(1) + P(0)\} = E_s = \int_0^T s^2(t)dt = A^2 T$$

4.2.2 Binary Pulse Amplitude Modulation (B-PAM)

Average Energy per Bit

$$\bar{E}_b = \frac{\bar{E}_s}{b} = A^2 T$$

Euclidean Distance Between Symbols

$$d_{min}^2 = \int_0^T (s(t) - (-s_(t)))^2 dt = \int_0^T (2s(t))^2 dt = 4A^2 T$$

4.2.2 Binary Pulse Amplitude Modulation (B-PAM)

Power Efficiency of a Signal

$$\varepsilon_p = \frac{d_{\min}^2}{\bar{E}_b} = \frac{4A^2T}{A^2T} = 4$$

A power efficiency result of 4 is the best possible result that you can obtain for any digital modulation scheme when all possible binary sequences are each mapped to a unique symbol.

4.2.2 M-ary Pulse Amplitude Modulation (M-PAM)

We try mapping binary sequences to one of M possible unique signal amplitude levels.

The M-PAM waveform are given as

$$s_i(t) = A_i \cdot p(t), \text{ for } i = 1, 2, \ldots, M/2$$

where $A_i = A(2i - 1)$ and $p(t) = u(t) - u(t - T)$.

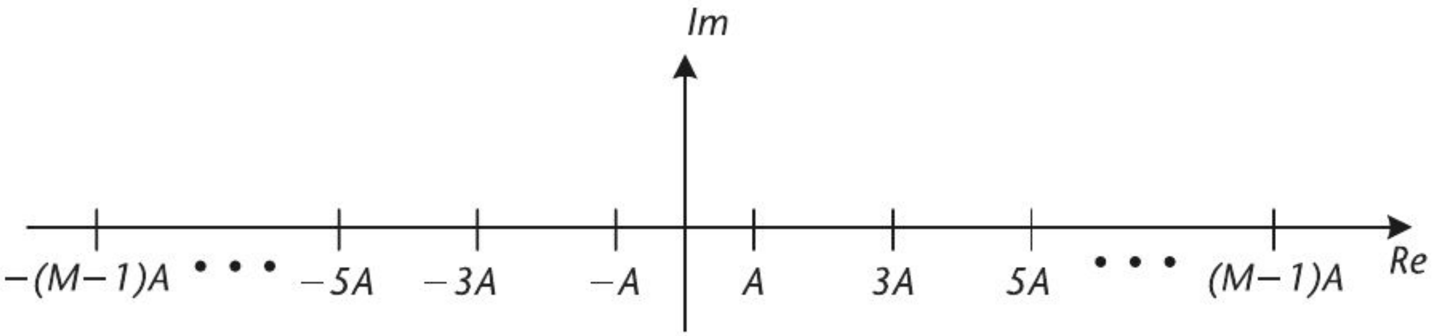

Figure 4.8 M-PAM signal constellation.

4.2.2 M-ary Pulse Amplitude Modulation (M-PAM)

The Minimum Euclidean Distance

$$d_{\min}^2 = 4A^2T$$

Average Symbol Energy

$$\begin{aligned}\bar{E}_s &= \frac{2}{M}A^2T\sum_{i=1}^{M/2}(2i-1)^2 \\ &= A^2T\frac{(M^2-1)}{3}\end{aligned}$$

4.2.2 M-ary Pulse Amplitude Modulation (M-PAM)

Average Energy per Bit

$$\bar{E}_b = \frac{\bar{E}_s}{\log_2(M)} = \frac{A^2T(2^{2b} - 1)}{3b}$$

Power Efficiency

$$\varepsilon_{p,\text{M-PAM}} = \frac{12b}{2^{2b} - 1}$$

4.2.3 Quadrature Amplitude Modulation (QAM)

QAM modulation is a two-dimensional signal modulation scheme that uses two orthogonal signals (in-phase and quadrature).

Rectangular QAM can be thought of as two orthogonal PAM signals being transmitted simultaneously.

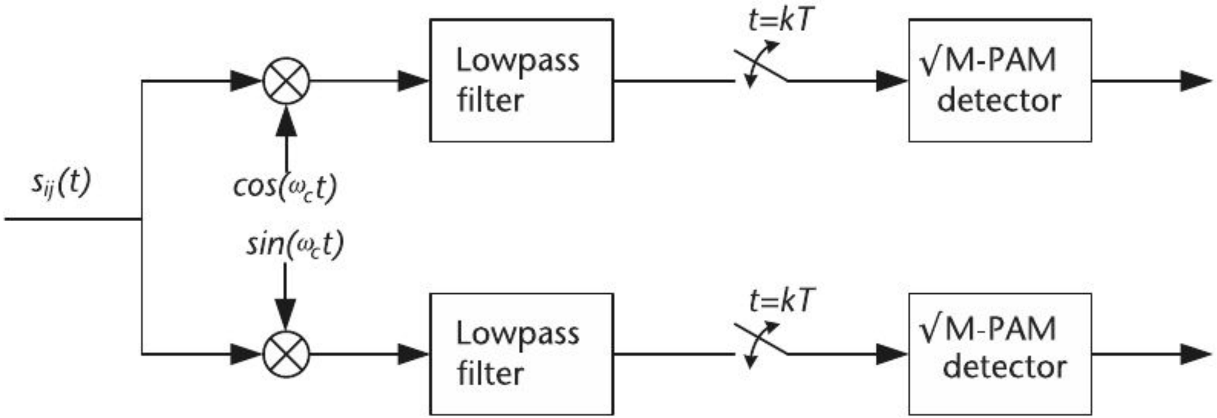

Figure 4.9 M-QAM receiver structure.

4.2.3 Quadrature Amplitude Modulation (QAM)

The mathematical representation of a signal waveform belonging to this form of modulation is

$$s_{ij}(t) = A_i \cdot \cos(\omega_c t) + B_j \cdot \sin(\omega_c t)$$

The Minimum Euclidean Distance

$$d_{\min}^2 = \int_0^T \Delta s^2(t)dt = 2A^2T$$

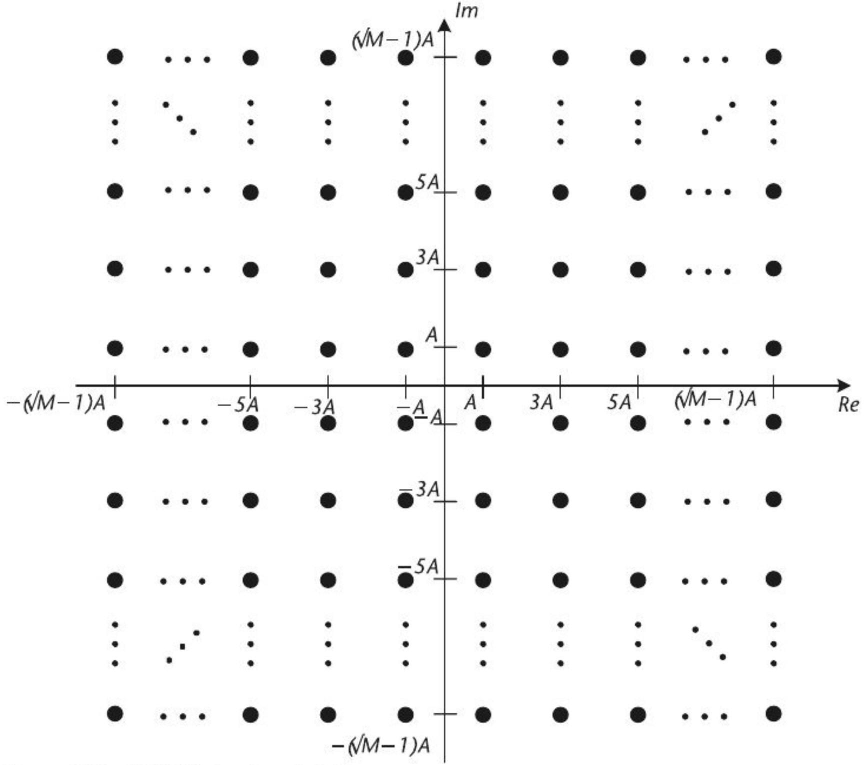

Figure 4.10 M-QAM signal constellation.

4.2.3 Quadrature Amplitude Modulation (QAM)

Average Symbol Energy

We use the expression from M-ary PAM by replacing M with $\sqrt{M}$ such that

$$\bar{E}_s = A^2 T \frac{M-1}{3}$$

Average Energy per Bit

$$\bar{E}_b = \frac{\bar{E}_s}{\log_2(M)} = A^2 T \frac{2^b - 1}{3b}$$

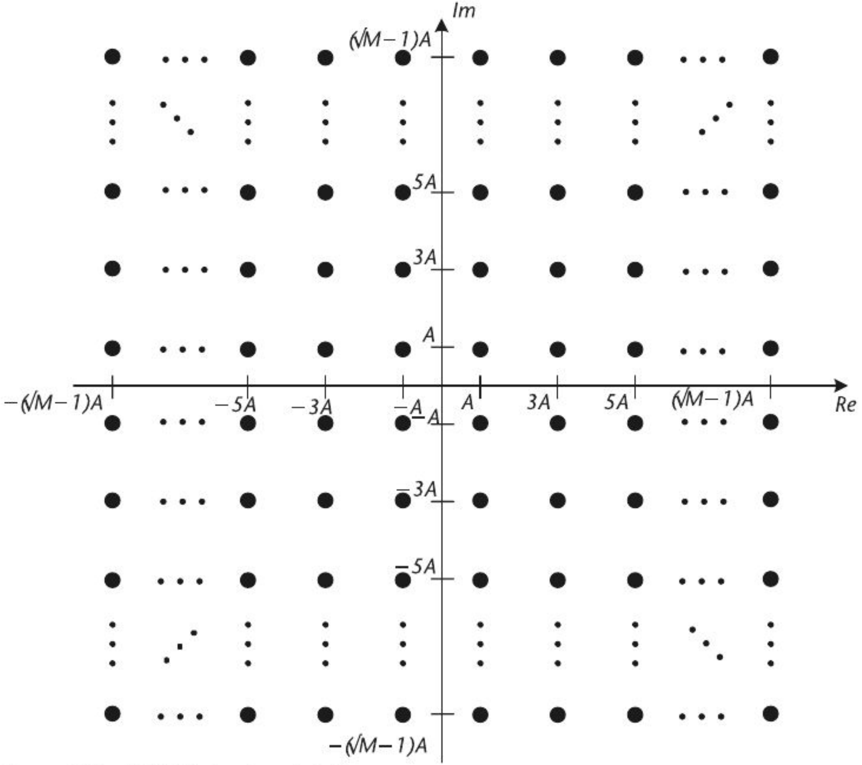

Figure 4.10 M-QAM signal constellation.

4.2.3 Quadrature Amplitude Modulation (QAM)

Power Efficiency

$$\varepsilon_{p,\mathrm{M-QAM}} = \frac{3!b}{2^b - 1}$$

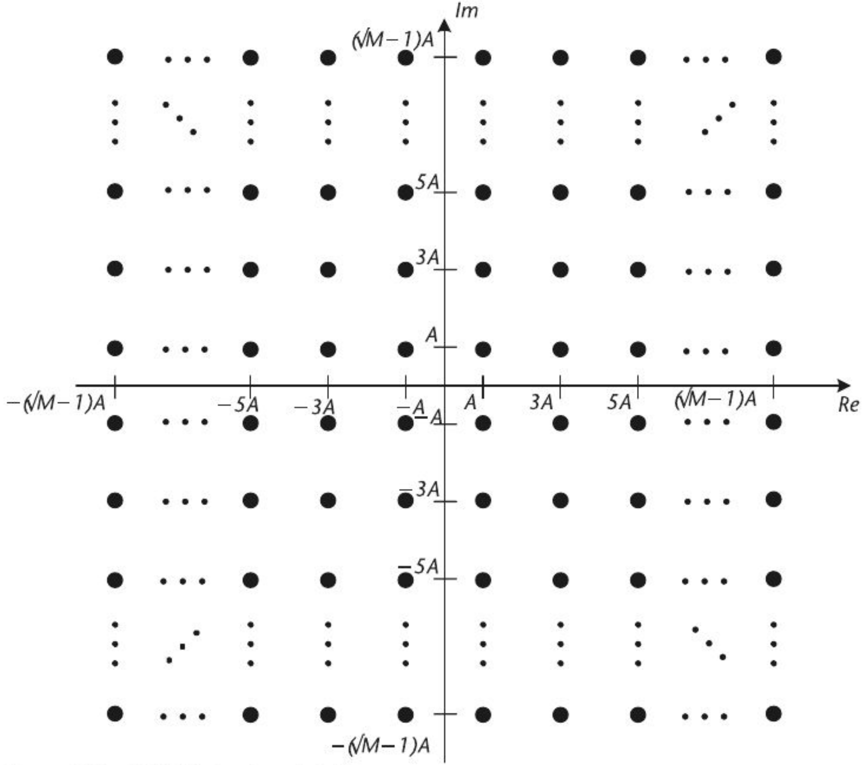

Figure 4.10 M-QAM signal constellation.

4.2.3 Quadrature Amplitude Modulation (QAM)

Let’s look at some MATLAB code on *decode_qam.m* file.

4.2.4 Phase Shift Keying (PSK)

Is a digital modulation scheme that conveys data by changing or modulating the phase of a reference signal.

PSK uses a finite number of phases, each assigned a unique pattern of binary digits. Usually, each phase encodes an equal number of bits.

The demulator which is designed specifically for the symbol set used by the modulator, determines the phase of the received signal, and maps it back to the symbol it represents, thus recovering the original data.

The receiver needs to be able to compare the phase of the received signal to a reference signal. Such a system is termed coherent.

4.2.4 Phase Shift Keying (PSK)

Mathematically, a PSK signal waveform is represented by

$$s_i(t) = A\cos\left(2\pi f_c t + (2i-1)\frac{\pi}{m}\right), \quad \text{for} \quad i = 1, \ldots, \log_2 m$$

Where

- A is the amplitude,
- f_c is carrier frequency, and
- $(2i-1)\frac{\pi}{m}$ is the phase offset of each symbol

4.2.4 Binary Phase Shift Keying (B-PSK)

One of the most popular and most robust phase shift keying modulations.

The modulation rules are as follows:

$$
\begin{aligned}
\text{“1”} \rightarrow s_1(t) &= A \cdot \cos(\omega_c t + \theta) \\
\text{“0”} \rightarrow s_2(t) &= -A \cdot \cos(\omega_c t + \theta) \\
&= A \cdot \cos(\omega_c (t) + \theta + \pi) \\
&= -s_1(t).
\end{aligned}
$$

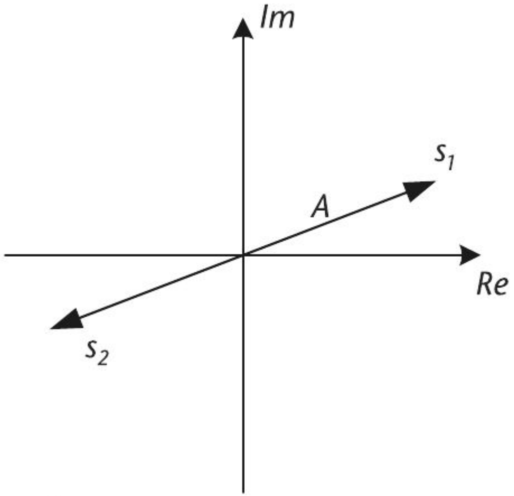

Figure 4.12 BPSK signal constellation.

4.2.4 Binary Phase Shift Keying (B-PSK)

The Minimum Euclidean Distance

$$
\begin{aligned}
d_{\min}^2 &= \int_0^T (s_1(t) - s_2(t))^2 dt \\
&= 4A^2 \int_0^T \cos^2(\omega_c t + \theta) dt \\
&= \frac{4A^2 T}{2} + \frac{4A^2}{2} \int_0^T \cos(2\omega_c t + 2\theta) dt \\
&= 2A^2 T.
\end{aligned}
$$

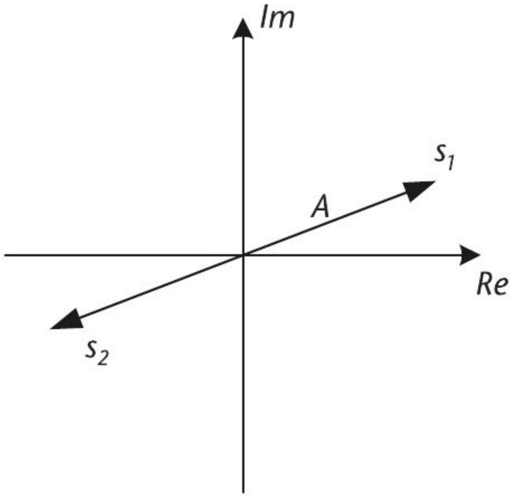

Figure 4.12 BPSK signal constellation.

4.2.4 Binary Phase Shift Keying (B-PSK)

Average Energy per Bit

$$E_{s_1} = \int_0^T s_1^2(t)dt = A^2 \int_0^T \cos^2(\omega_c t + \theta)dt$$

$$= \frac{A^2T}{2} + \frac{A^2}{2}\int_0^T \cos(2\omega_c t + 2\theta)dt$$

$$= \frac{A^2T}{2}$$

$$E_{s_2} = \frac{A^2T}{2}$$

$$\bar{E}_b = P(0)\cdot E_{s_2} + P(1)\cdot E_{s_1} = \frac{A^2T}{2}.$$

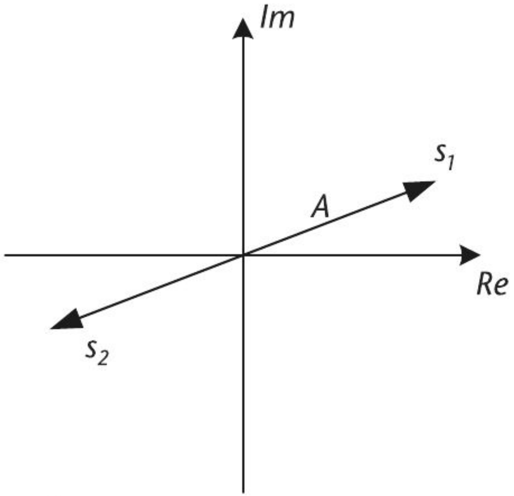

Figure 4.12 BPSK signal constellation.

4.2.4 Binary Phase Shift Keying (B-PSK)

Power Efficiency

$$\varepsilon_{p,\mathrm{BPSK}} = \frac{d_{\min}^2}{\bar{E}_b} = 4.$$

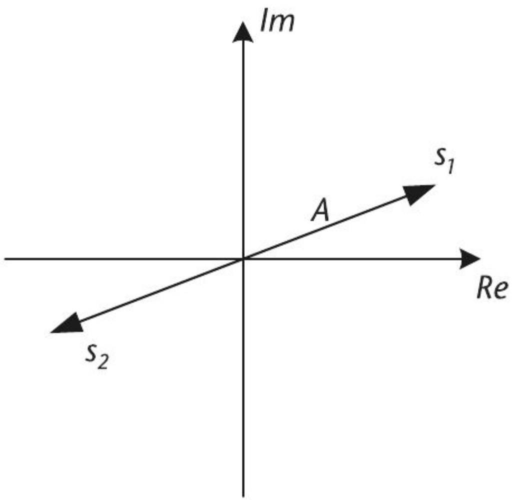

Figure 4.12 BPSK signal constellation.

4.2.4 Binary Phase Shift Keying (B-PSK)

A note on the Euclidean Distance:

Another way for computing $d^2_{\min}$ is to use the concept of correlation.

We can express the minimum Euclidean distance as

$$d^2_{\min} = \int_0^T (s_2(t) - s_1(t))^2 dt = E_{s_1} + E_{s_2} - 2\rho_{12} = 2(E - \rho_{12})$$

where the symbol energy for symbol i, E_{si} , and the correlation between symbols 1 and 2, $\rho 12$, are given by

$$E_{s_i} = \int_0^T s_i^2(t)dt \quad \text{and} \quad \rho_{12} = \int_0^T s_1(t)s_2(t)dt$$

4.2.4 Binary Phase Shift Keying (B-PSK)

A note on the Euclidean Distance:

In order to get a large ε_p , we need to maximize $d^2_{\min}$, which means we want

ρ12< 0.

which means $d^2_{\min} = 2(E - \rho_{12})$ and consequently $\rho_{12} = -E$.

4.2.4 Quadrature Phase Shift Keying (Q-PSK)

Uses two bits per symbol.

A signal waveform possesses the following representation:

$$s_i(t) = \pm A \cdot \cos(\omega_c t + \theta) \pm A \cdot \sin(\omega_c t + \theta),$$

where each signal waveform possesses the same amplitude but one of four possible phase values.

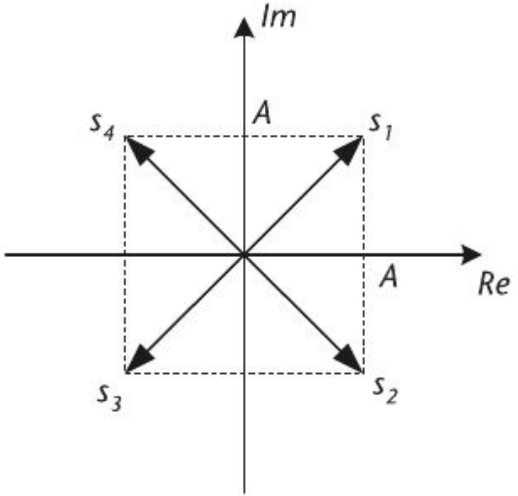

Figure 4.13 QPSK signal constellation.

4.2.4 Quadrature Phase Shift Keying (Q-PSK)

The Minimum Euclidean Distance

$$d_{\min}^2 = \int_0^T \Delta s^2(t)dt = 2A^2T$$

Average Energy per Bit

$$\bar{E}_b = \frac{(E_{s_1} + E_{s_2} + E_{s_3} + E_{s_4})/4}{\log_2(M)} = \frac{A^2T}{2}$$

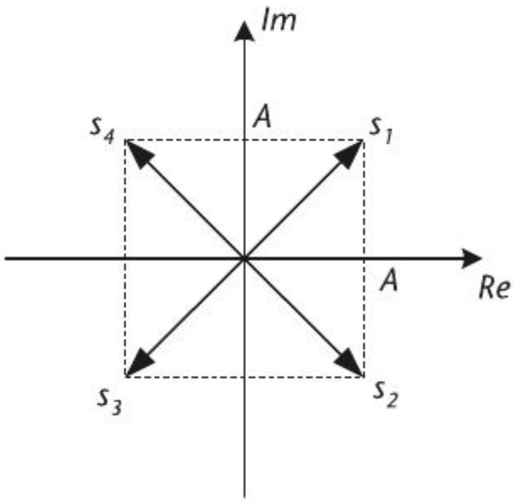

Figure 4.13 QPSK signal constellation.

4.2.4 Quadrature Phase Shift Keying (Q-PSK)

Power Efficiency

$$\varepsilon_{p,\text{QPSK}} = \frac{d_{\min}^2}{\bar{E}_b} = 4$$

Same as BPSK but with 2 bits per symbol.

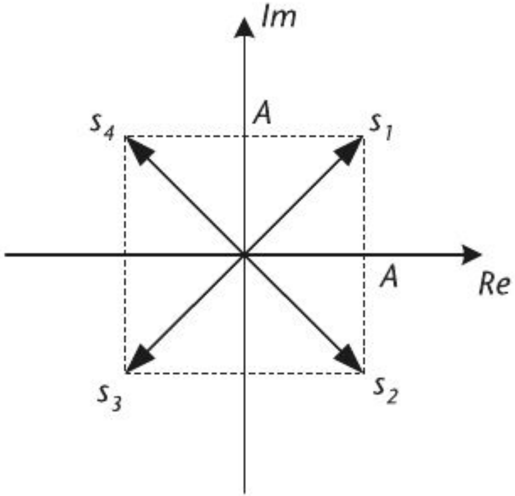

Figure 4.13 QPSK signal constellation.

4.2.4 M- Phase Shift Keying (M-PSK)

A signal waveform can be mathematically represented as

$$s_i(t) = A \cdot \cos\left(\omega_c t + \frac{2\pi i}{M}\right), \text{ for } i = 0, 1, 2, \ldots, M-1$$

4.2.4 M- Phase Shift Keying (M-PSK)

The Minimum Euclidean Distance

$$d_{\min}^2 = E_{s_1} + E_{s_2} - 2\rho_{12}$$

Symbol energy

$$E_{s_i} = \int_0^T s_i^2(t)dt = \frac{A^2T}{2}, \text{ for } i = 1, 2,$$

Correlation between the two signal waveforms

$$\rho_{12} = \int_0^T s_1(t)s_2(t)dt = \frac{A^2T}{2}\cos\left(\frac{2\pi}{M}\right)$$

4.2.4 M- Phase Shift Keying (M-PSK)

The Minimum Euclidean Distance

$$d_{\min}^2 = E_{s_1} + E_{s_2} - 2\rho_{12} = A^2T\left(1 - \cos\left(\frac{2\pi}{M}\right)\right)$$

Symbol energy

$$E_{s_i} = \int_0^T s_i^2(t)dt = \frac{A^2T}{2}, \text{ for } i = 1, 2,$$

Correlation between the two signal waveforms

$$\rho_{12} = \int_0^T s_1(t)s_2(t)dt = \frac{A^2T}{2}\cos\left(\frac{2\pi}{M}\right)$$

4.2.4 M- Phase Shift Keying (M-PSK)

Average Energy per Bit

$$\bar{E}_b = \frac{\bar{E}_s}{\log_2(M)} = \frac{\bar{E}_s}{b}, \text{ where } \bar{E}_s = A^2T/2$$

Power Efficiency

$$\varepsilon_{p,\mathrm{M-PSK}} = 2b\left(1 - \cos\left(\frac{2\pi}{M}\right)\right) = 4b\sin^2\left(\frac{\pi}{2^b}\right)$$

4.2.5 Power Efficiency Summary

We can summarize the following power efficiency calculations

$$\varepsilon_{p,\mathrm{M-PAM}} = \frac{12b}{2^{2b}-1} \qquad \varepsilon_{p,\mathrm{M-PSK}} = 4b\sin^2\left(\frac{\pi}{2^b}\right) \qquad \varepsilon_{p,\mathrm{M-QAM}} = \frac{3!b}{2^b-1}$$

4.2.5 Power Efficiency Summary

We can summarize the following power efficiency calculations

$$\varepsilon_{p,\mathrm{M-PAM}} = \frac{12b}{2^{2b}-1} \qquad \varepsilon_{p,\mathrm{M-PSK}} = 4b\sin^2\left(\frac{\pi}{2^b}\right) \qquad \varepsilon_{p,\mathrm{M-QAM}} = \frac{3!b}{2^b-1}$$

We use the following expression

$$\delta\mathrm{SNR} = 10 \cdot \log_{10}\left(\frac{\varepsilon_{p,\mathrm{QPSK}}}{\varepsilon_{p,\mathrm{other}}}\right)$$

to determine how much power efficiency we are losing relative to $\varepsilon_{p,\mathrm{QPSK}}$, which possesses the best possible result.

4.2.5 Power Efficiency Summary

Two-dimensional modulation schemes perform better than the one-dimensional modulation schemes.

These modulation schemes are linear, which means they possess a similar level of receiver complexity

Table 4.1 δSNR Values of Various Modulation Schemes

M	b	M-ASK	M-PSK	M-QAM
2	1	0	0	0
4	2	4	0	0
8	3	8.45	3.5	—
16	4	13.27	8.17	4.0
32	5	18.34	13.41	—
64	6	24.4	18.4	8.45

4.2.5 Power Efficiency Summary

Let’s look at some MATLAB code on *digital_modulations.m* file.

4.3 Probability of Bit Error

BER is the probability that a bit transmitted will be decoded incorrectly.

Is very important when assessing whether the design of a digital communication system meets the specific error robustness requirements of the application to be supported.

Having a metric that quantifies error performance is helpful when comparing one digital communication design with another.

4.3 Probability of Bit Error

We will mathematically describe the probability of error by employing the concept of hypothesis testing.

The receiver can decide on whether $s_1(t)$ or $s_2(t)$ was sent based on the observation of the intercepted signal r(t).

The following hypothesis testing framework:

$$\mathcal{H}_1 : r(t) = s_1(t) + n(t), \; 0 \le t \le T$$
$$\mathcal{H}_0 : r(t) = s_2(t) + n(t), \; 0 \le t \le T$$

where $\mathcal{H}_0$ and $\mathcal{H}_1$ are Hypothesis 0 and Hypothesis 1

4.3 Probability of Bit Error

We now can establish a decision rule at the receiver such that it can select which waveform was sent based on the intercept signal.

A decision rule based on the correlation between two signals.

4.3 Probability of Bit Error

We now can establish a decision rule at the receiver such that it can select which waveform was sent based on the intercept signal.

A decision rule based on the correlation between two signals.

Assume that $s_1(t)$ was transmitted a decision rule on whether $s_1(t)$ or $s_2(t)$ was transmitted given that we observe r(t) is defined as

$$\int_0^T r(t)s_1(t)dt \geq \int_0^T r(t)s_2(t)dt$$

4.3 Probability of Bit Error

In the situation where a transmitted signal waveform is sufficiently corrupted such that it appears to be more correlated to another possible signal waveform, the receiver could potentially select an incorrect waveform, thus yielding an error event.

The error occurs when

$$\int_0^T r(t)s_1(t)dt \leq \int_0^T r(t)s_2(t)dt$$

4.3 Probability of Bit Error

In the situation where a transmitted signal waveform is sufficiently corrupted such that it appears to be more correlated to another possible signal waveform, the receiver could potentially select an incorrect waveform, thus yielding an error event.

Since $r(t) = s_1(t) + n(t)$, we can substitute this into the error event in order to obtain the decision rule

$$\int_0^T s_1^2(t)dt + \int_0^T n(t)s_1(t)dt \leq \int_0^T s_1(t)s_2(t)dt + \int_0^T n(t)s_2(t)dt$$

4.3 Probability of Bit Error

$$\int_0^T r(t)s_1(t)dt \leq \int_0^T r(t)s_2(t)dt$$

$$\int_0^T s_1^2(t)dt + \int_0^T n(t)s_1(t)dt \leq \int_0^T s_1(t)s_2(t)dt + \int_0^T n(t)s_2(t)dt$$

$$E_{s_1} - \rho_{12} \leq \int_0^T n(t)(s_2(t) - s_1(t))dt$$

$$E_{s_1} - \rho_{12} \leq z$$

With $z \sim \mathcal{N}(0, \sigma^2)$

4.3 Probability of Bit Error

We now need to express this inequality as a probability

$$E_{s_1} - \rho_{12} \leq z$$

In other words, we need to find

$$P(e|1) = P(z \geq E - \rho_{12})$$

4.3 Probability of Bit Error

We now need to express this inequality as a probability

$$E_{s_1} - \rho_{12} \leq z$$

In other words, we need to find

$$P(e|1) = P(z \geq E - \rho_{12})$$

The Q function is a convenient way to express right-tail probabilities for Gaussian random variables, P(X > x). Mathematically, this is equivalent to finding the complementary CDF of X;

$$Q(x) = 1 - F_X(x) = 1 - P(X \leq x)$$

$$= P(X > x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-t^2/2} dt$$

Where FX (x) is the CDF of X

4.3 Probability of Bit Error

We now need to express this inequality as a probability

$$E_{s_1} - \rho_{12} \leq z$$

In other words, we need to find

$$P(e|1) = P(z \geq E - \rho_{12})$$

$$P(z \geq E - \rho_{12}) = Q\left(\frac{E - \rho_{12}}{\sigma}\right)$$

Since $z \sim \mathcal{N}(0, \sigma^2)$
and $E - \rho_{12}$ is constant

The Q function is a convenient way to express right-tail probabilities for Gaussian random variables, P(X > x). Mathematically, this is equivalent to finding the complementary CDF of X;

$$Q(x) = 1 - F_X(x) = 1 - P(X \leq x)$$
$$= P(X > x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-t^2/2} dt$$

Where FX (x) is the CDF of X

4.3 Probability of Bit Error

We now need to express this inequality as a probability

$$E_{s_1} - \rho_{12} \leq z$$

In other words, we need to find

$$P(e|1) = P(z \geq E - \rho_{12})$$

$$P(z \geq E - \rho_{12}) = Q\left(\frac{E - \rho_{12}}{\sigma}\right)$$

The variance of z, σ^2, can be solved as follows

$$\sigma^2 = E\{z^2\} = \frac{N_0}{2}\int_0^T (s_1(t) - s_2(t))^2 dt$$

$$= \frac{N_0}{2}(E_{s_1} + E_{s_2} - 2\rho_{12}) \rightarrow \text{Assume } E_{s_1} = E_{s_2} = E$$

$$= N_0(E - \rho_{12})$$

4.3 Probability of Bit Error

We now need to express this inequality as a probability

$$E_{s_1} - \rho_{12} \leq z$$

In other words, we need to find

$$P(e|1) = P(z \geq E - \rho_{12})$$

$$P(z \geq E - \rho_{12}) = Q\left(\frac{E - \rho_{12}}{\sigma}\right)$$

$$= Q\left(\sqrt{\frac{(E - \rho_{12})^2}{\sigma^2}}\right)$$

$$= Q\left(\sqrt{\frac{E - \rho_{12}}{N_0}}\right),$$

The variance of z, σ^2, can be solved as follows

$$\sigma^2 = E\{z^2\} = \frac{N_0}{2}\int_0^T (s_1(t) - s_2(t))^2 dt$$

$$= \frac{N_0}{2}(E_{s_1} + E_{s_2} - 2\rho_{12}) \rightarrow \text{Assume } E_{s_1} = E_{s_2} = E$$

$$= N_0(E - \rho_{12})$$

4.3 Probability of Bit Error

We are interested in optimize the probability of bit error by optimizing the probability of error by minimizing P(e|1)

$$P(z \geq E - \rho_{12}) = Q\left(\sqrt{\frac{E - \rho_{12}}{N_0}}\right)$$

This can be achieved by setting ρ12 = −E.

$$P(e|1) = Q\left(\sqrt{\frac{2\bar{E}_b}{N_0}}\right)$$

4.3 Probability of Bit Error

We are interested in optimize the probability of bit error by optimizing the probability of error by minimizing P(e|1)

$$P(z \geq E - \rho_{12}) = Q\left(\sqrt{\frac{E - \rho_{12}}{N_0}}\right)$$

when $E_{s1} \neq E_{s2}$, we can then use $d^2_{min} = E_{s1} + E_{s2} - 2\rho_{12}$

$$P_e = Q\left(\sqrt{\frac{d^2_{\min}}{2N_0}}\right)$$

4.3 Probability of Bit Error

When dealing with a large number of signal waveforms that form a modulation scheme, the resulting probability of error, Pe , is expressed as a sum of pairwise error probabilities

$$Q\left(\frac{d_{ij}^2}{2N_0}\right)$$

4.3 Probability of Bit Error

When dealing with a large number of signal waveforms that form a modulation scheme, the resulting probability of error, Pe , is expressed as a sum of pairwise error probabilities

$$Q\left(\frac{d_{ij}^2}{2N_0}\right)$$

the complete expression for Pe can be expressed as

$$Q\left(\frac{d_{\min}^2}{2N_0}\right) \leq P_e \leq Q\left(\frac{d_{1j}^2}{2N_0}\right) + \ldots + Q\left(\frac{d_{Mj}^2}{2N_0}\right), \quad i \neq j;$$

4.3 Probability of Bit Error

Let's look at some MATLAB code on *euclidian_distance.m* file.

4.3.1 Error Bounding

An accurate estimate of P(e) can be computed from the following bounds.

These upper and lower bounds can be expressed as

$$Q\left(\frac{d_{min}^2}{2N_0}\right) \leq P(e) \leq \sum_{i \in I} Q\left(\frac{d_{ij}^2}{2N_0}\right)$$

where I is the set of all signal waveforms within the signal constellation that are immediately adjacent to the signal waveform j.

In order to accurately assess the performance of a communications system, it must be simulated until a certain number of symbol errors are confirmed. In most cases, 100 errors will give a 95% confidence interval.

4.3.1 Error Bounding

Let’s look at some MATLAB code on *monte_carlo.m* file.

4.4 Signal Space Concept

We will use signal vectors to characterize and analyze our modulation schemes.

Suppose we define $\varphi_j(t)$ as an orthonormal set of functions over the time interval $[0, T]$ such that

$$\int_0^T \phi_i(t)\phi_j(t)dt = \begin{cases} 1 & i = j \\ 0 & \text{otherwise} \end{cases}$$

Given that $s_i(t)$ is the *ith* signal waveform, we would like to represent this waveform as a sum of several orthonormal functions

$$s_i(t) = \sum_{k=1}^{N} s_{ik}\phi_k(t)$$

which can be equivalently represented by the vector

$$\mathbf{s}_i = (s_{i1},\ s_{i2},\ s_{i3}, \ldots s_{iN})$$

4.4 Signal Space Concept

In order to find the vector elements, s_{il}, we need to solve the expression

$$\int_0^T s_i(t)\phi_l(t)dt = \sum_{k=1}^{N} s_{ik} \int_0^T \phi_k(t)\phi_l(t)dt = s_{il}$$

which is essentially a dot product or projection of the signal waveform $s_i(t)$ on the orthonormal function $\varphi_l(t)$.

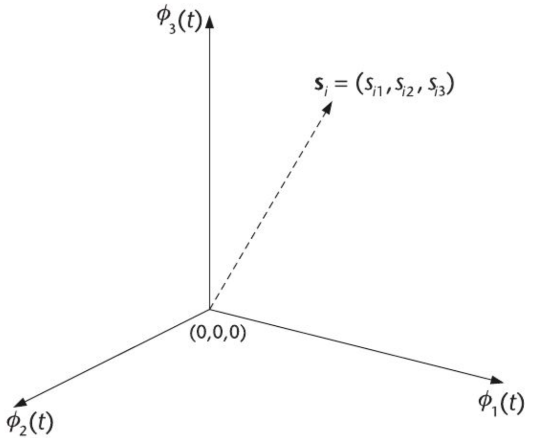

Figure 4.17 Sample vector representation of $s_i(t)$ in three-dimensional space using basis functions $\phi_1(t)$, $\phi_2(t)$, and $\phi_3(t)$.

4.4 Signal Space Concept

Correlation

$$\int_0^T s_i(t)s_j(t)dt = \mathbf{s}_i \cdot \mathbf{s}_j = \rho_{ij}$$

Signal Energy

$$E_{s_i} = \int_0^T s_i^2(t)dt = \mathbf{s}_i \cdot \mathbf{s}_i = ||\mathbf{s}_i||^2$$

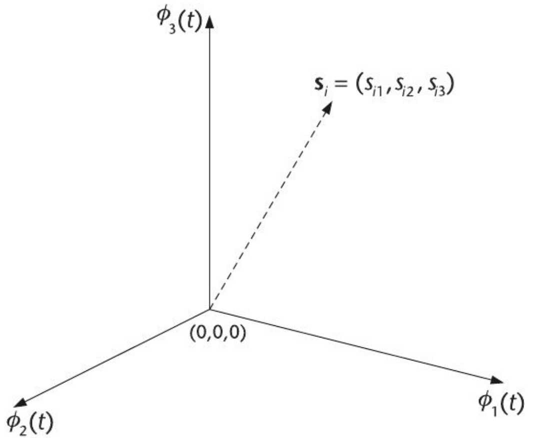

Figure 4.17 Sample vector representation of $s_i(t)$ in three-dimensional space using basis functions $\phi_1(t)$, $\phi_2(t)$, and $\phi_3(t)$.

4.4 Signal Space Concept

Minimum Euclidean Distance

$$d_{\min}^2 = \int_0^T \Delta s_{ij}^2(t)dt = \int_0^T (s_i(t) - s_j(t))^2 dt$$
$$= ||\mathbf{s}_i - \mathbf{s}_j||^2 = (\mathbf{s}_i - \mathbf{s}_j) \cdot (\mathbf{s}_i - \mathbf{s}_j)$$
$$= E_{s_i} + E_{s_j} - 2\rho_{ij}$$

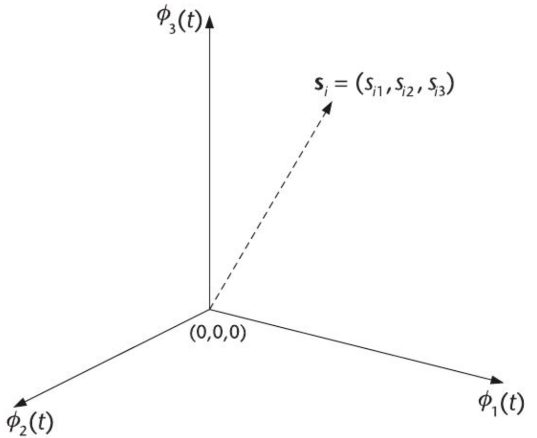

Figure 4.17 Sample vector representation of $s_i(t)$ in three-dimensional space using basis functions $\phi_1(t)$, $\phi_2(t)$, and $\phi_3(t)$.

4.4 Signal Space Concept

Average Symbol Energy

$$\bar{E}_s = \frac{1}{M}\sum_{i=1}^{M} ||\mathbf{s}_i||^2$$

Average Symbol Energy

$$\bar{E}_s = \frac{1}{M}\sum_{i=1}^{M} ||\mathbf{s}_i||^2$$

Power Efficiency

$$\varepsilon_p = d_{\min}^2/\bar{E}_b$$

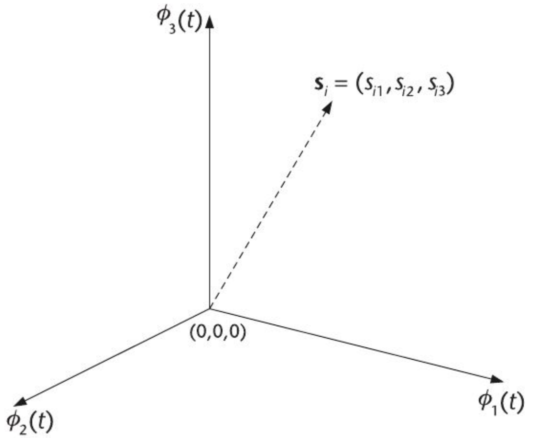

Figure 4.17 Sample vector representation of $s_i(t)$ in three-dimensional space using basis functions $\phi_1(t)$, $\phi_2(t)$, and $\phi_3(t)$.

4.5 Gram-Schmidt Orthogonalization

The Gram-Schmidt orthogonalization process is a method for creating an orthonormal set of functions in an inner product space such as the Euclidean space $\mathbb{R}^{\mathrm{n}}$.

The Gram-Schmidt orthogonalization process takes a finite set of signal waveforms $\{s_1(\mathrm{t}), \dots, s_{\mathrm{M}}(\mathrm{t})\}$ and generates from it an orthogonal set of functions $\{\phi_1(\mathrm{t}), \dots, \phi_{\mathrm{i}}(\mathrm{t})\}$ that spans the space $\mathbb{R}^{\mathrm{n}}$.

4.5 Gram-Schmidt Orthogonalization

An orthonormal function possesses the following property:

$$\int_0^T \phi_i(t)\phi_j(t)dt = \begin{cases} 1 & i = j \\ 0 & \text{otherwise} \end{cases}$$

It is possible to represent a signal waveform $s_i(t)$ as the weighted sum of these orthonormal basis functions

$$s_i(t) = \sum_{k=1}^{N} s_{ik}\phi_k(t)$$

4.5 Gram-Schmidt Orthogonalization

Algorithm:

Given the signals $\{s_1(t), s_2(t), \ldots, s_k(t)\}$

1. Calculate $\phi_1(t) = \frac{s_1(t)}{\sqrt{E_{s1}}}$

4.5 Gram-Schmidt Orthogonalization

Algorithm:

Given the signals $\{s_1(t), s_2(t), \ldots, s_k(t)\}$

1. Calculate $\phi_1(t) = \frac{s_1(t)}{\sqrt{E_{s1}}}$

2. Calculate $\phi_2(t) = \frac{g_2(t)}{\sqrt{\int_0^T g_2^2(t)dt}}$

4.5 Gram-Schmidt Orthogonalization

Algorithm:

Given the signals $\{s_1(t), s_2(t), \ldots, s_k(t)\}$

1. Calculate $\phi_1(t) = \frac{s_1(t)}{\sqrt{E_{s1}}}$

2. Calculate $\phi_2(t) = \frac{g_2(t)}{\sqrt{\int_0^T g_2^2(t)dt}}$

Where:

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

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

4.5 Gram-Schmidt Orthogonalization

Algorithm:

Given the signals $\{s_1(t), s_2(t), \ldots, s_k(t)\}$

3. Repeat with

$$\phi_i(t) = \frac{g_i(t)}{\sqrt{\int_0^T g_i^2(t)dt}}, \; i = 1, 2, \ldots, N.$$

Where:

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

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

4.5 Gram-Schmidt Orthogonalization

Homework:

Given the signals shown in the image. Perform the Gram-Schmidt orthogonalization procedure in the order $s_3(\mathrm{t}), s_1(\mathrm{t}), s_4(\mathrm{t}), s_2(\mathrm{t})$.

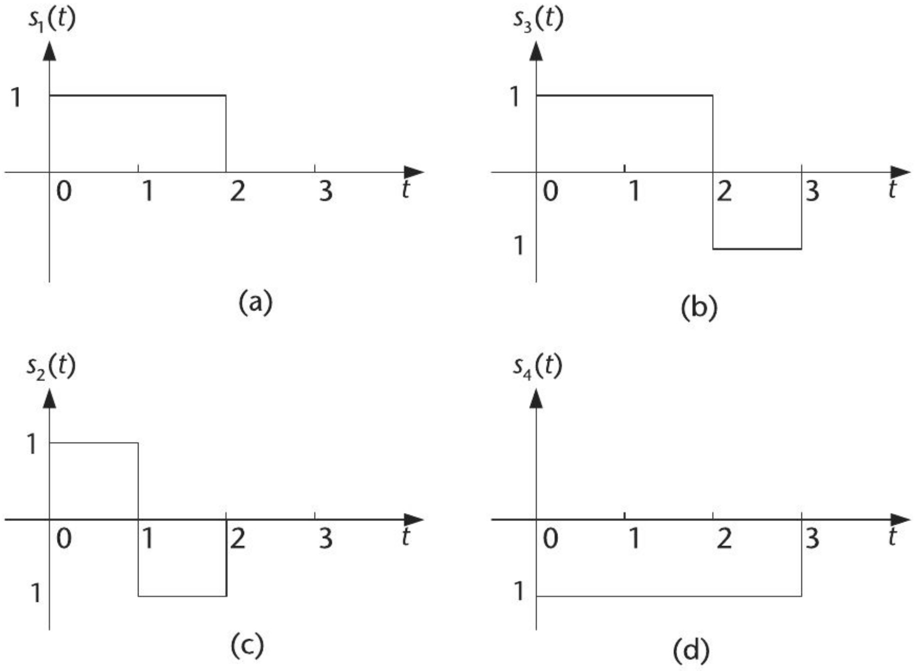

Figure 4.18 Example signal waveforms.

4.5 Gram-Schmidt Orthogonalization

Homework:

Given the signals shown in the image. Perform the Gram-Schmidt orthogonalization procedure in the order $s_3(t), s_1(t), s_4(t), s_2(t)$.

Result

$$\begin{aligned}
\mathbf{s}_1 &= (2/\sqrt{3}, \sqrt{6}/3, 0), \\
\mathbf{s}_2 &= (0, 0, \sqrt{2}), \\
\mathbf{s}_3 &= (\sqrt{3}, 0, 0), \\
\mathbf{s}_4 &= (-1/\sqrt{3}, -4/\sqrt{6}, 0)
\end{aligned}$$

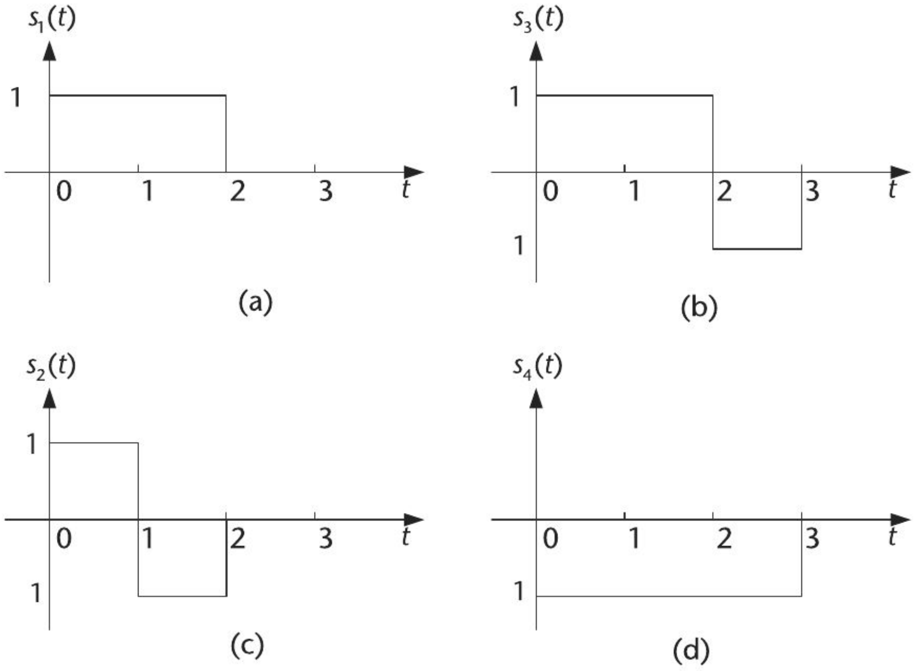

Figure 4.18 Example signal waveforms.

4.5 Gram-Schmidt Orthogonalization

Let’s look at some MATLAB code on *gram_schmidt.m* file.

4.6 Optimal Detection

Signal detection theory is used to discern between signal and noise.

Using this theory, we can explain how changing the decision threshold will affect the ability to discern between two or more scenarios, often exposing how adapted the system is to the task, purpose, or goal at which it is aimed.

4.6.1 Signal Vector Framework

The receiver only observes the corrupted version of $s_i(t)$ by the noise signal $n(t)$; namely, $r(t)$.

Our detection problem in this situation can be summarized as follows: Given $r(t)$ for $0 \leq t \leq T$, determine which $s_i(t)$, $i = 1, 2, \ldots, M$, is present in the intercepted signal $r(t)$.

4.6.1 Signal Vector Framework

Suppose we decompose the waveforms $s_i(t)$, $n(t)$, and $r(t)$ into a collection of weights applied to a set of orthonormal basis functions:

$$s_i(t) = \sum_{k=1}^{N} s_{ik}\phi_k(t), \quad r(t) = \sum_{k=1}^{N} r_k\phi_k(t), \quad n(t) = \sum_{k=1}^{N} n_k\phi_k(t)$$

We can rewrite the waveform model expression $r(t) = s_i(t) + n(t)$ into

$$\sum_{k=1}^{N} r_k\phi_k(t) = \sum_{k=1}^{N} s_{ik}\phi_k(t) + \sum_{k=1}^{N} n_k\phi_k(t)$$

$$\mathbf{r} = \mathbf{s}_i + \mathbf{n}.$$

4.6.1 Signal Vector Framework

Since the noise signal $n(t)$ is assumed to be a Gaussian random variable, the noise signal vector $\mathbf{n}$ is a Gaussian vector.

The mean of these vector elements

$$
\begin{aligned}
E\{n_k\} &= E\left\{\int_0^T n(t)\phi_k(t)dt\right\} \\
&= \int_0^T E\{n(t)\}\phi_k(t)dt \\
&= 0
\end{aligned}
$$

4.6.1 Signal Vector Framework

Since the noise signal $n(t)$ is assumed to be a Gaussian random variable, the noise signal vector $\mathbf{n}$ is a Gaussian vector.

The variance of these vector elements

$$E\{n_k n_l\} = \frac{N_0}{2}\delta(k-l)$$

As a result, the matrix equivalent of this outcome is equal to

$$E\{\mathbf{n}\mathbf{n}^T\} = \frac{N_0}{2}\mathbf{I}_{N\times N}$$

4.6.1 Signal Vector Framework

Since the noise signal $n(t)$ is assumed to be a Gaussian random variable, the noise signal vector $\mathbf{n}$ is a Gaussian vector.

The joint probability density function

$$p(\mathbf{n}) = p(n_1, n_2, \ldots, n_N) = \frac{1}{(2\pi\sigma^2)^{N/2}} e^{-||\mathbf{n}||^2/2\sigma^2}$$

4.6.2 Decision Rules

We can define a rule for the receiver that can be used to determine which signal waveform is being intercepted given the presence of some noise introduced by the channel.

Two types of detectors:

- Maximum a Posteriori (MAP) detector

$$P(\mathbf{s}_i|\mathbf{r} = \rho) = \max_{\mathbf{s}_i} p(\rho|\mathbf{s}_i)P(\mathbf{s}_i)$$

- Maximum Likelihood (ML) detector

$$P(\mathbf{s}_i|\mathbf{r} = \rho) = \max_{\mathbf{s}_i} p(\rho|\mathbf{s}_i)$$

4.6.3 Maximum Likelihood Detection in an AWGN Channel

A maximum likelihood approach selects values of the model parameters that produce the distribution that are most likely to have resulted in the observed data.

The conditional probability of the received vector $\mathrm{r} = \rho$ can be obtained as

$$p(\rho|\mathbf{s}_i) = \prod_{k=1}^{N} p(\rho_k|s_{ik}), \text{ for } i = 1, 2, \ldots, M$$

Where

$$p(\rho_k|s_{ik}) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(\rho_k - s_{ik})^2/2\sigma^2}$$

4.6.3 Maximum Likelihood Detection in an AWGN Channel

This product of multiple elemental probability density functions will yield the following expression

$$p(\rho|\mathbf{s}_i) = \frac{1}{(2\pi\sigma^2)^{N/2}} e^{-||\rho-\mathbf{s}_i||^2/2\sigma^2}$$

we would like to solve for $\max_{\mathbf{s}_i} p(\rho|\mathbf{s}_i)$

4.6.3 Maximum Likelihood Detection in an AWGN Channel

Apply logarithm to both sides

$$\ln(p(\rho|\mathbf{s}_i)) = \frac{N}{2}\ln\left(\frac{1}{2\pi\sigma^2}\right) - \frac{||\rho - \mathbf{s}_i||^2}{2\sigma^2}$$

We can derive the following

$$\begin{aligned}\max_{\mathbf{s}_i} \ln(p(\rho|\mathbf{s}_i)) &= \max_{\mathbf{s}_i}\left(\frac{N}{2}\ln\left(\frac{1}{2\pi\sigma^2}\right) - \frac{||\rho - \mathbf{s}_i||^2}{2\sigma^2}\right) \\ &= \max_{\mathbf{s}_i}\left(-\frac{||\rho - \mathbf{s}_i||^2}{2\sigma^2}\right) \\ &= \max_{\mathbf{s}_i}\left(-||\rho - \mathbf{s}_i||^2\right) \\ &= \min_{\mathbf{s}_i} ||\rho - \mathbf{s}_i||.\end{aligned}$$

4.6.3 Maximum Likelihood Detection in an AWGN Channel

We can rewrite this decision rule as

$$\mathbf{s}_k = \arg\min_{\mathbf{s}_i} ||\rho - \mathbf{s}_i|| \rightarrow \hat{\mathbf{m}} = \mathbf{m}$$

A maximum likelihood detector is the equivalent of a minimum distance detector.

4.7 Basic Receiver Realizations

We will study two types of receivers

- Matched Filter
- Correlator

4.7.1 Matched Filter Realization

We are interested in detecting a pulse transmitted over a channel corrupted by noise.

Suppose we employ the following transmission model:

$$x(t) = g(t) + w(t), 0 \leq t \leq T$$

Where:

- $g(t)$ is a pulse signal,
- $w(t)$ is a white noise process with mean $\mu = 0$ and power spectral density equal to $N_0/2$,
- and $x(t)$ is the observed received signal

4.7.1 Matched Filter Realization

The optimal value for $\mathrm{H(f)}$ should be equal to

$$H_{\mathrm{opt}}(f) = K \cdot G^*(f) e^{-j2\pi fT}$$

The time domain representation is

$$h_{\mathrm{opt}}(t) = K \cdot \int_{-\infty}^{\infty} G^*(f) e^{-j2\pi fT} e^{-j2\pi ft} df = K \cdot g(T - t).$$

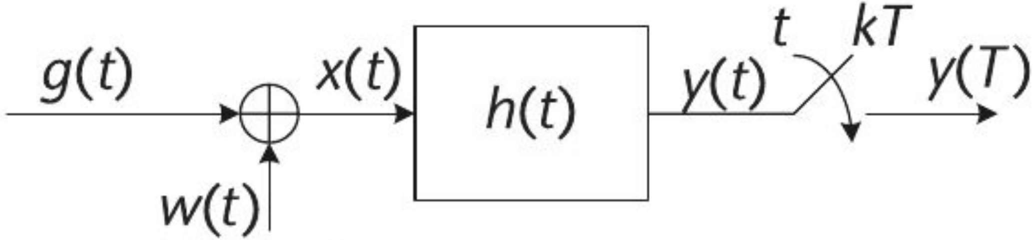

Figure 4.22 Filtering process for detecting $g(t)$.

4.7.1 Matched Filter Realization

When we are performing a matched filtering operation, we are convolving the time-flipped and time-shifted version of the transmitted pulse with the transmitted pulse itself in order to maximize the SNR.

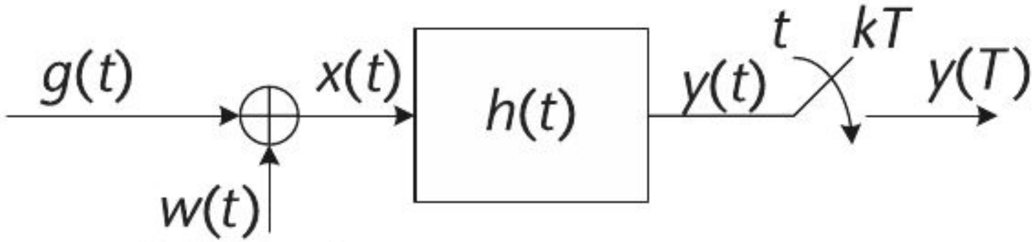

Figure 4.22 Filtering process for detecting $g(t)$.

4.7.1 Matched Filter Realization

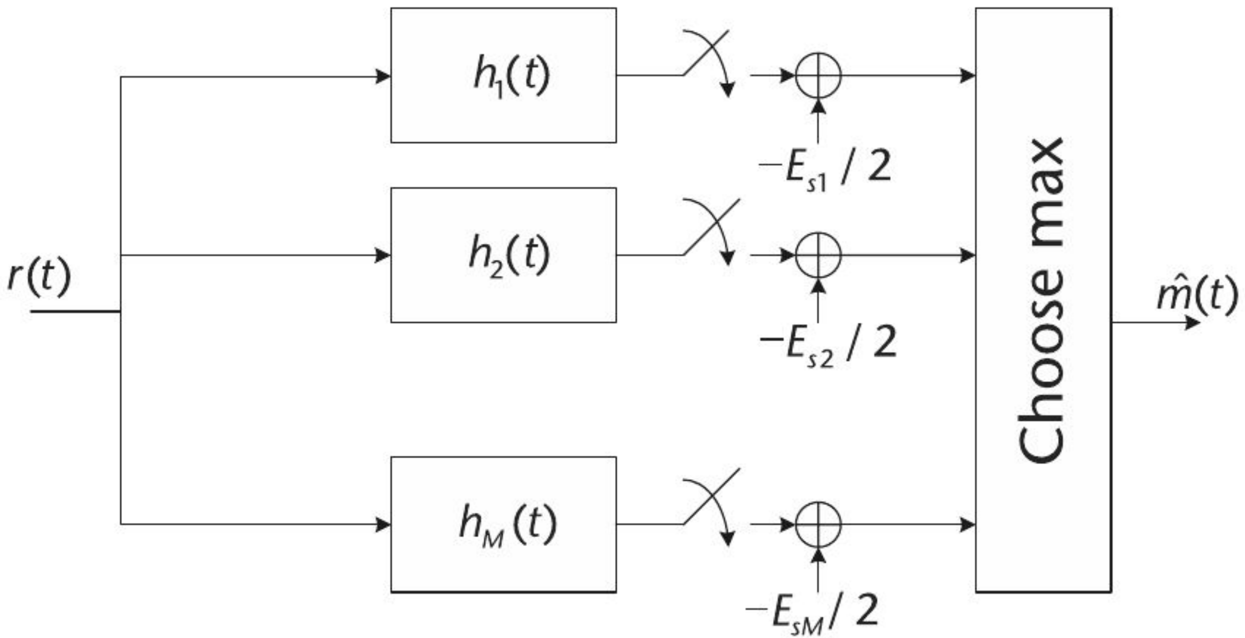

Figure 4.23 Schematic of matched filter realization of receiver structure.

4.7.2 Correlator Realization

We can employ the concept of correlation such that we only need to assume knowledge about the waveforms themselves.

Suppose we start with the decision rule

$$\min_{\mathbf{s}_i} ||\rho - \mathbf{s}_i||^2 = \min_{\mathbf{s}_i} (\rho - \mathbf{s}_i) \cdot (\rho - \mathbf{s}_i)$$
$$= \rho \cdot \rho - 2\rho \cdot \mathbf{s}_i + \mathbf{s}_i \cdot \mathbf{s}_i$$

Since $\rho \cdot \rho$ is common to all the decision metrics for different values of the signal waveforms $\mathbf{s}_i$, we can conveniently omit it from the expression

4.7.2 Correlator Realization

$$\min_{\mathbf{s}_i}(-2\rho\cdot\mathbf{s}_i+\mathbf{s}_i\cdot\mathbf{s}_i)=\max_{\mathbf{s}_i}(2\rho\cdot\mathbf{s}_i-\mathbf{s}_i\cdot\mathbf{s}_i)$$

where $\rho\cdot\mathbf{s_i}$ and $\mathbf{s_i}\cdot\mathbf{s_i}$ are defined by

$$\rho\cdot\mathbf{s}_i=\int_0^T\rho(t)s_i(t)dt \qquad \mathbf{s}_i\cdot\mathbf{s}_i=\int_0^T s_i^2(t)dt=E_{s_i}$$

When $s_k(t)$ is present in $r(t)$ the optimal detector is equal to

$$reliable\mathbf{s}_k=\arg\max_i\left(\int_0^T\rho(t)s_i(t)dt-\frac{E_{s_i}}{2}\right)$$

4.7.2 Correlator Realization

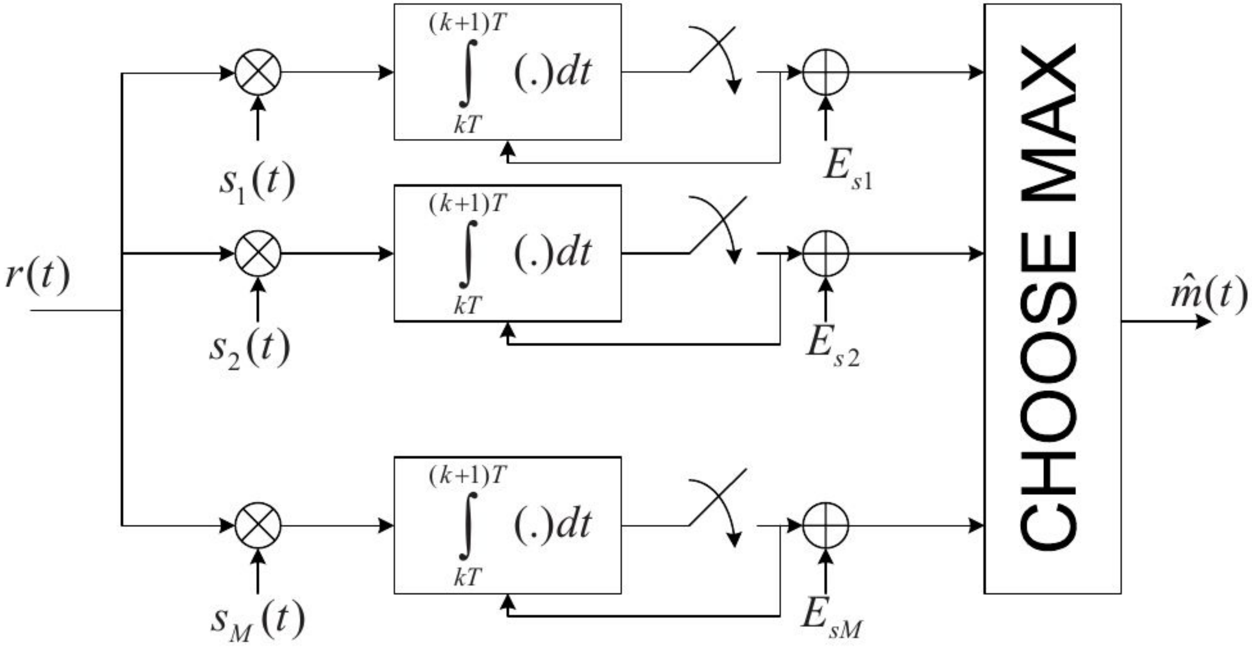

Figure 4.25 Correlator realization of a receiver structure assuming perfect synchronization.

4.7.2 Correlator Realization

Let's look at some MATLAB code on correlator_receiver.*m* file.