28

Line Coding

28.1 Introduction
28.2 Common Line Coding Formats
Unipolar NRZ (Binary On-Off Keying) • Unipolar RZ • Polar NRZ • Polar RZ [Bipolar, Alternate Mark Inversion (AMI), or Pseudoternary] • Manchester Coding (Split Phase or Digital Biphase)
28.3 Alternate Line Codes
Delay Modulation (Miller Code) • Split Phase (Mark) • Biphase (Mark) • Code Mark Inversion (CMI) • NRZ (I) • Binary *N* Zero Substitution (BNZS) • High-Density Bipolar *N* (HDBN) • Ternary Coding
28.4 Multilevel Signalling, Partial Response Signalling, and Duobinary Coding
Multilevel Signalling • Partial Response Signalling and Duobinary Coding
28.5 Bandwidth Comparison
28.6 Concluding Remarks

Joseph L. LoCicero
Illinois Institute of Technology

Bhasker P. Patel
Illinois Institute of Technology

28.1 Introduction

The terminology **line coding** originated in telephony with the need to transmit digital information across a copper telephone *line*; more specifically, binary data over a digital repeatered line. The concept of line coding, however, readily applies to any transmission line or channel. In a digital communication system, there exists a known set of symbols to be transmitted. These can be designated as $\{m_i\}$, $i = 1, 2, ., N$, with a probability of occurrence $\{p_i\}$, $i = 1, 2, ., N$, where the sequentially transmitted symbols are generally assumed to be statistically independent. The conversion or *coding* of these abstract symbols into real, temporal waveforms to be transmitted in baseband is the process of line coding. Since the most common type of line coding is for binary data, such a waveform can be succinctly termed a direct format for serial bits. The concentration in this section will be line coding for binary data.

Different channel characteristics, as well as different applications and performance requirements, have provided the impetus for the development and study of various types of line coding [1,2]. For example, the channel might be AC coupled and, thus, could not support a line code with a DC component or large DC content. Synchronization or timing recovery requirements might necessitate a discrete component at the data rate. The channel bandwidth and **crosstalk** limitations might dictate the type of line coding employed. Even such factors as the complexity of the encoder and the economy of the decoder could determine the line code chosen. Each line code has its own distinct properties. Depending on the application, one property may be more important than the other. In what follows, we describe, in general, the most desirable features that are considered when choosing a line code.

It is commonly accepted [1,2,5,8] that the dominant considerations effecting the choice of a line code are: (1) timing, (2) DC content, (3) power spectrum, (4) performance monitoring, (5) probability of error, and (6) transparency. Each of these are detailed in the following paragraphs.

(1) *Timing:* The waveform produced by a line code should contain enough timing information such that the receiver can synchronize with the transmitter and decode the received signal properly. The timing content should be relatively independent of source statistics, i.e., a long string of **1**s or **0**s should not result in loss of timing or jitter at the receiver.

(2) *DC content:* Since the repeaters used in telephony are AC coupled, it is desirable to have zero DC in the waveform produced by a given line code. If a signal with significant DC content is used in AC coupled lines, it will cause **DC wander** in the received waveform. That is, the received signal baseline will vary with time. Telephone lines do not pass DC due to AC coupling with transformers and capacitors to eliminate DC ground loops. Because of this, the telephone channel causes a droop in constant signals. This causes DC wander. It can be eliminated by DC restoration circuits, feedback systems, or with specially designed line codes.

(3) *Power spectrum:* The power spectrum and bandwidth of the transmitted signal should be matched to the frequency response of the channel to avoid significant distortion. Also, the power spectrum should be such that most of the energy is contained in as small bandwidth as possible. The smaller the bandwidth, the higher the transmission efficiency.

(4) *Performance monitoring:* It is very desirable to detect errors caused by a noisy transmission channel. The error detection capability in turn allows performance monitoring while the channel is in use (i.e., without elaborate testing procedures that require suspending use of the channel).

(5) *Probability of error:* The average error probability should be as small as possible for a given transmitter power. This reflects the reliability of the line code.

(6) *Transparency:* A line code should allow all the possible patterns of **1**s and **0**s. If a certain pattern is undesirable due to other considerations, it should be mapped to a unique alternative pattern.

28.2 Common Line Coding Formats

A line coding format consists of a formal definition of the line code that specifies how a string of binary digits are converted to a line code waveform. There are two major classes of binary line codes: **level codes** and **transition codes**. Level codes carry information in their voltage level, which may be high or low for a full bit period or part of the bit period. Level codes are usually instantaneous since they typically encode a binary digit into a distinct waveform, independent of any past binary data. However, some level codes do exhibit memory. Transition codes carry information in the change in level appearing in the line code waveform. Transition codes may be instantaneous, but they generally have memory, using past binary data to dictate the present waveform. There are two common forms of level line codes: one is called **return to zero (RZ)** and the other is called **nonreturn to zero (NRZ)**. In RZ coding, the level of the pulse returns to zero for a portion of the bit interval. In NRZ coding, the level of the pulse is maintained during the entire bit interval.

Line coding formats are further classified according to the polarity of the voltage levels used to represent the data. If only one polarity of voltage level is used, i.e., positive or negative (in addition to the zero level), then it is called **unipolar** signalling. If both positive and negative voltage levels are being used, with or without a zero voltage level, then it is called **polar** signalling. The term **bipolar** signalling is used by some authors to designate a specific line coding scheme with positive, negative, and zero voltage levels. This will be described in detail later in this section. The formal definition of five common line codes is given in the following along with a representative waveform, the *power spectral density* (PSD), the probability of error, and a discussion of advantages and disadvantages. In some cases specific applications are noted.

Unipolar NRZ (Binary On-Off Keying)

In this line code, a binary **1** is represented by a non-zero voltage level and a binary **0** is represented by a zero voltage level as shown in Fig. 28.1(a). This is an instantaneous level code. The PSD of this code with equally likely **1**s and **0**s is given by [5,8]

$$S_1(f) = \frac{V^2T}{4}\left(\frac{\sin \pi f T}{\pi f T}\right)^2 + \frac{V^2}{4}\delta(f) \tag{28.1}$$

where V is the binary **1** voltage level, $T = 1/R$ is the bit duration, and R is the bit rate in bits per second. The spectrum of unipolar NRZ is plotted in Fig. 28.2a. This PSD is a two-sided even spectrum, although only half of the plot is shown for efficiency of presentation. If the probability of a binary **1** is p, and the probability of a binary **0** is $(1 - p)$, then the PSD of this code, in the most general case, is $4p(1 - p)S_1(f)$. Considering the frequency of the first spectral null as the bandwidth of the waveform, the bandwidth of unipolar NRZ is R in hertz. The error rate performance of this code, for equally likely data, with additive white Gaussian noise (AWGN) and optimum, i.e., matched filter, detection is given by [1,5]

$$P_e = \frac{1}{2}\operatorname{erfc}\left(\sqrt{\frac{E_b}{2N_0}}\right) \tag{28.2}$$

where E_b/N_0 is a measure of the signal-to-noise ratio (SNR) of the received signal. In general, E_b is the energy per bit and $N_0/2$ is the two-sided PSD of the AWGN. More specifically, for unipolar NRZ, E_b is the energy in a binary **1**, which is V^2T. The performance of the unipolar NRZ code is plotted in Fig. 28.3.

The principle advantages of unipolar NRZ are ease of generation, since it requires only a single power supply, and a relatively low bandwidth of R Hz. There are quite a few disadvantages of this line code. A loss of synchronization and timing jitter can result with a long sequence of **1**s or **0**s because no pulse transition is present. The code has no error detection capability and, hence, performance cannot be monitored. There is a significant DC component as well as a DC content. The error rate performance is not as good as that of polar line codes.

Unipolar RZ

In this line code, a binary **1** is represented by a nonzero voltage level during a portion of the bit duration, usually for half of the bit period, and a zero voltage level for rest of the bit duration. A binary **0** is represented by a zero voltage level during the entire bit duration. Thus, this is an instantaneous level code. Figure 28.1(b) illustrates a unipolar RZ waveform in which the **1** is represented by a nonzero voltage level for half the bit period. The PSD of this line code, with equally likely binary digits, is given by [5,6,8]

$$S_2(f) = \frac{V^2T}{16}\left(\frac{\sin \pi f T/2}{\pi f T/2}\right)^2 + \frac{V^2}{4\pi^2}\left[\frac{\pi^2}{4}\delta(f) + \sum_{n=-\infty}^{\infty}\frac{1}{(2n+1)^2}\delta(f-(2n+1)R)\right] \tag{28.3}$$

where again V is the binary **1** voltage level, and $T = 1/R$ is the bit period. The spectrum of this code is drawn in Fig. 28.2a. In the most general case, when the probability of a **1** is p, the continuous portion of the PSD in Eq. (28.3) is scaled by the factor $4p(1 - p)$ and the discrete portion is scaled by the factor $4p^2$. The first null bandwidth of unipolar RZ is $2R$ Hz. The error rate performance of this line code is the same as that of the unipolar NRZ, provided we increase the voltage level of this code such that the energy in binary **1**, E_b, is the same for both codes. The probability of error is given by Eq. (28.2) and identified in Fig. 28.3. If the voltage level and bit period are the same for unipolar NRZ and unipolar RZ, then the energy in a binary **1** for unipolar RZ will be $V^2T/2$ and the probability of error is worse by 3 dB.

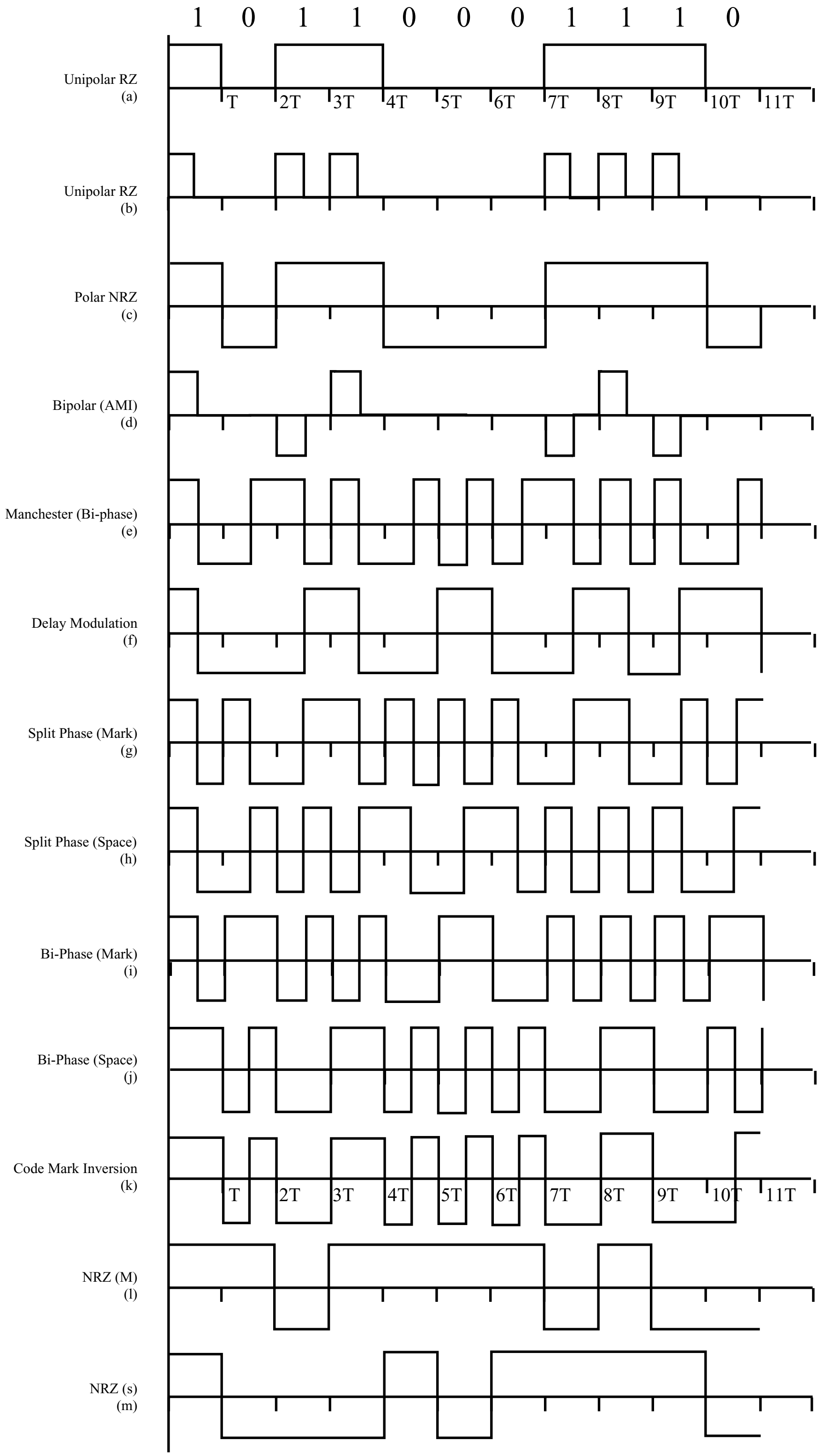

FIGURE 28.1 Waveforms for different line codes.

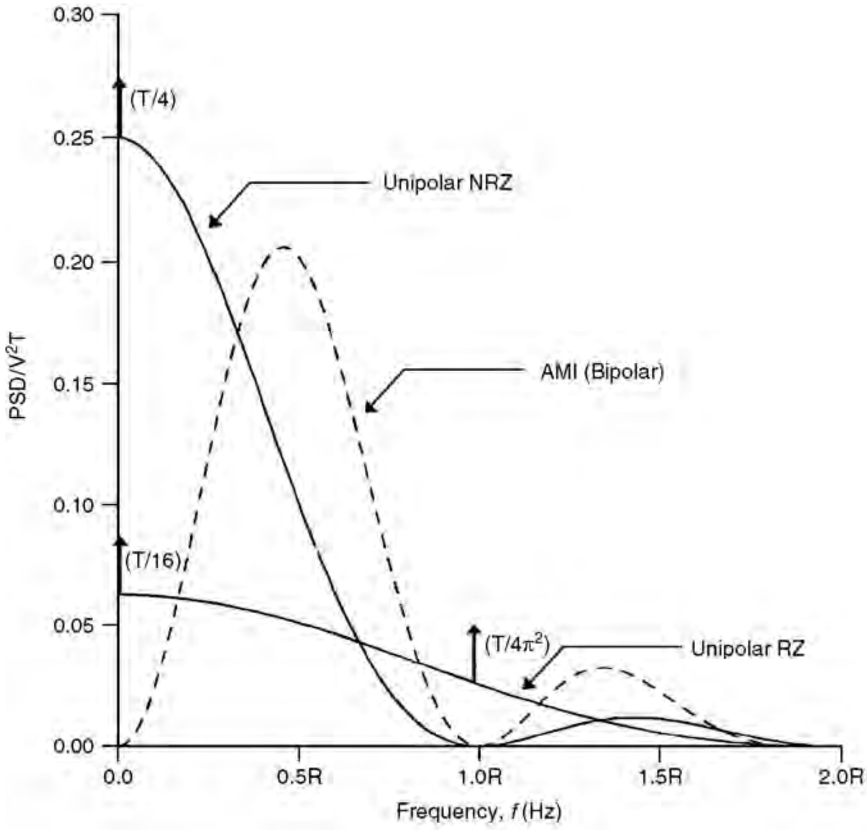

FIGURE 28.2a Power spectral density of different line codes, where $R = 1/T$ is the bit rate.

The main advantages of unipolar RZ are, again, ease of generation since it requires a single power supply and the presence of a discrete spectral component at the symbol rate, which allows simple timing recovery. A number of disadvantages exist for this line code. It has a nonzero DC component and nonzero DC content, which can lead to DC wander. A long string of **0**s will lack pulse transitions and could lead to loss of synchronization. There is no error detection capability and, hence, performance monitoring is not possible. The bandwidth requirement ($2R$ Hz) is higher than that of NRZ signals. The error rate performance is worse than that of polar line codes.

Unipolar NRZ as well as unipolar RZ are examples of pulse/no-pulse type of signalling. In this type of signalling, the pulse for a binary **0**, $g_2(t)$, is zero and the pulse for a binary **1** is specified generically as $g_1(t) = g(t)$. Using $G(f)$ as the Fourier transform of $g(t)$, the PSD of pulse/no-pulse signalling is given as [6,7,10]

$$S_{\text{PNP}}(f) = p(1-p)R|G(f)|^2 + p^2R^2\sum_{n=-\infty}^{\infty}|G(nR)|^2\delta(f-nR) \tag{28.4}$$

where p is the probability of a binary **1**, and R is the bit rate.

Polar NRZ

In this line code, a binary **1** is represented by a positive voltage $+V$ and a binary **0** is represented by a negative voltage $-V$ over the full bit period. This code is also referred to as NRZ (L), since a bit is represented by maintaining a level (L) during its entire period. A polar NRZ waveform is shown in Fig. 28.1(c). This is again an instantaneous level code. Alternatively, a **1** may be represented by a $-V$

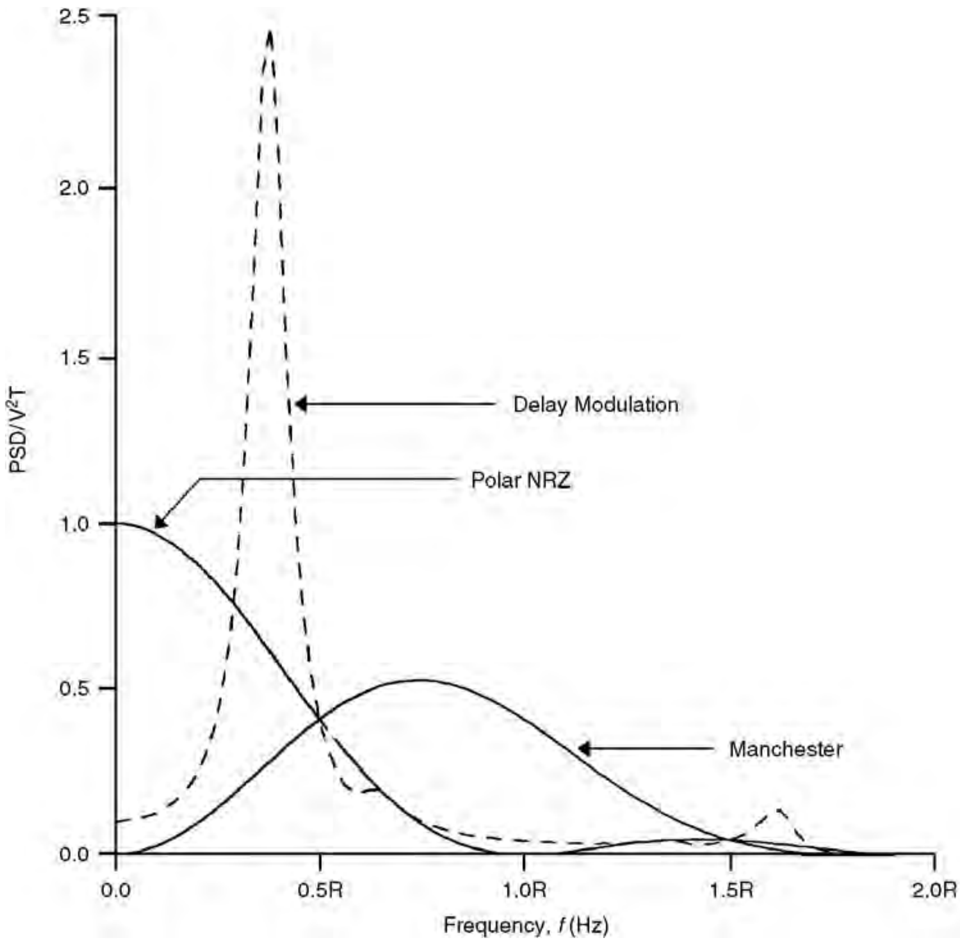

FIGURE 28.2b Power spectral density of different line codes, where $R = 1/T$ is the bit rate.

voltage level and a **0** by a $+V$ voltage level, without changing the spectral characteristics and performance of the line code. The PSD of this line code with equally likely bits is given by [5,8]

$$S_3(f) = V^2T\left(\frac{\sin \pi f T}{\pi f T}\right)^2 \tag{28.5}$$

This is plotted in Fig. 28.2b. When the probability of a **1** is p, and p is not 0.5, a DC component exists, and the PSD becomes [10]

$$S_{3p}(f) = 4V^2Tp(1-p)\left(\frac{\sin \pi f T}{\pi f T}\right)^2 + V^2(1-2p)^2\,\delta(f) \tag{28.6}$$

The first null bandwidth for this line code is again R Hz, independent of p. The probability of error of this line code when $p = 0.5$ is given by [1,5]

$$P_e = \frac{1}{2}\,\mathrm{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right) \tag{28.7}$$

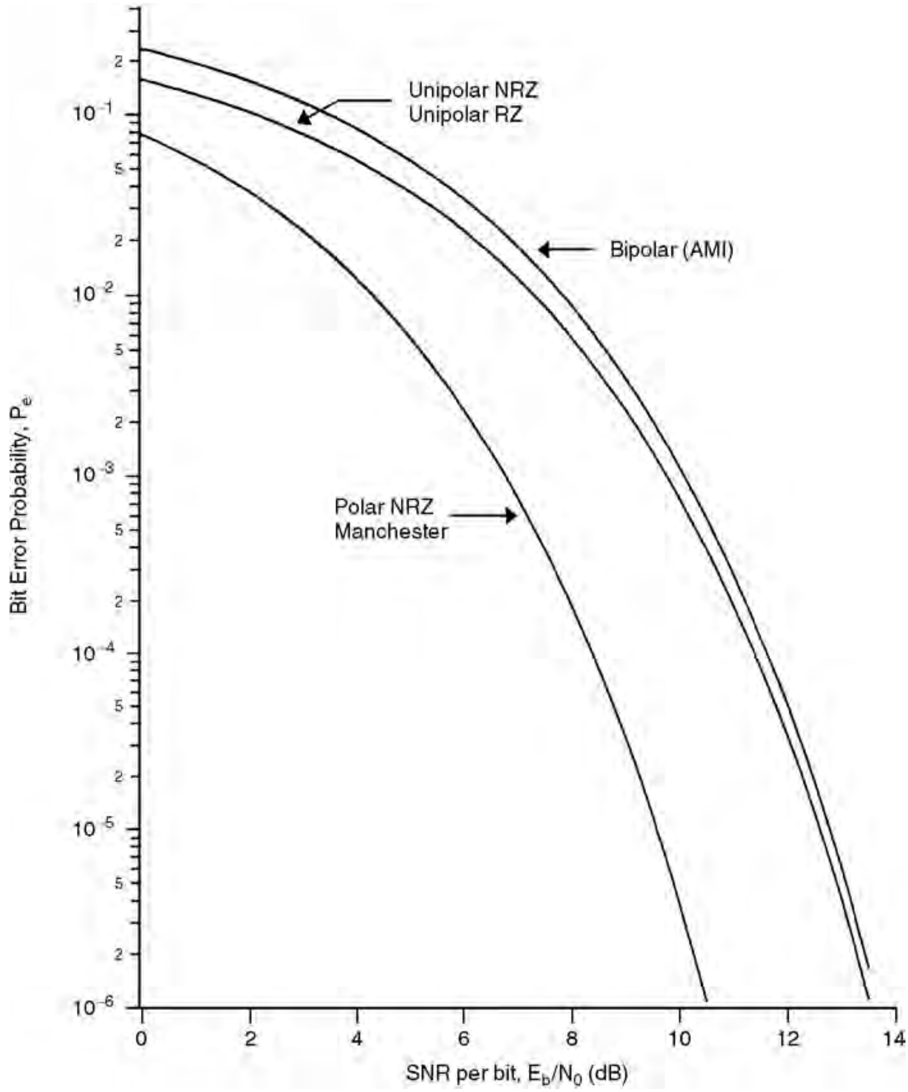

FIGURE 28.3 Bit error probability for different line codes.

The performance of polar NRZ is plotted in Fig. 28.3. This is better than the error performance of the unipolar codes by 3 dB.

The advantages of polar NRZ include a low-bandwidth requirement, *R* Hz, comparable to unipolar NRZ, very good error probability, and greatly reduced DC because the waveform has a zero DC component when $p = 0.5$ even though the DC content is never zero. A few notable disadvantages are that there is no error detection capability, and that a long string of **1**s or **0**s could result in loss of synchronization, since there are no transitions during the string duration. Two power supplies are required to generate this code.

Polar RZ [Bipolar, Alternate Mark Inversion (AMI), or Pseudoternary]

In this scheme, a binary **1** is represented by alternating the positive and negative voltage levels, which return to zero for a portion of the bit duration, generally half the bit period. A binary **0** is represented by a zero voltage level during the entire bit duration. This line coding scheme is often called **alternate mark inversion (AMI)** since **1**s (marks) are represented by alternating positive and negative pulses. It is also called *pseudoternary* since three different voltage levels are used to represent binary data. Some authors designate this line code as bipolar RZ (BRZ). An AMI waveform is shown in Fig. 28.1(d). Note that this is a level code with memory. The AMI code is well known for its use in telephony. The PSD of this line code with memory is given by [1,2,7]

$$S_{4p}(f) = 2p(1-p)R|G(f)|^2\left(\frac{1-\cos 2\pi f T}{1+(2p-1)^2+2(2p-1)\cos 2\pi f T}\right) \tag{28.8}$$

where $G(f)$ is the Fourier transform of the pulse used to represent a binary **1**, and p is the probability of a binary **1**. When $p = 0.5$ and square pulses with amplitude $\pm V$ and duration $T/2$ are used to represent binary **1**s, the PSD becomes

$$S_4(f) = \frac{V^2T}{4}\left(\frac{\sin \pi f T/2}{\pi f T/2}\right)^2 \sin^2(\pi f T) \tag{28.9}$$

This PSD is plotted in Fig. 28.2a. The first null bandwidth of this waveform is R Hz. This is true for RZ rectangular pulses, independent of the value of p in Eq. (28.8). The error rate performance of this line code for equally likely binary data is given by [5]

$$P_e \approx \frac{3}{4}\,\text{erfc}\left(\sqrt{\frac{E_b}{2N_0}}\right), \quad E_b/N_0 > 2 \tag{28.10}$$

This curve is plotted in Fig. 28.3 and is seen to be no more than 0.5 dB worse than the unipolar codes.

The advantages of polar RZ (or AMI, as it is most commonly called) outweigh the disadvantages. This code has no DC component and zero DC content, completely avoiding the DC wander problem. Timing recovery is rather easy since squaring, or full-wave rectifying, this type of signal yields a unipolar RZ waveform with a discrete component at the bit rate, R Hz. Because of the alternating polarity pulses for binary **1**s, this code has error detection and, hence, performance monitoring capability. It has a low-bandwidth requirement, R Hz, comparable to unipolar NRZ. The obvious disadvantage is that the error rate performance is worse than that of the unipolar and polar waveforms. A long string of **0**s could result in loss of synchronization, and two power supplies are required for this code.

Manchester Coding (Split Phase or Digital Biphase)

In this coding, a binary **1** is represented by a pulse that has positive voltage during the first-half of the bit duration and negative voltage during second-half of the bit duration. A binary **0** is represented by a pulse that is negative during the first-half of the bit duration and positive during the second-half of the bit duration. The negative or positive midbit transition indicates a binary **1** or binary **0**, respectively. Thus, a Manchester code is classified as an instantaneous transition code; it has no memory. The code is also called diphase because a square wave with a 0∞ phase is used to represent a binary **1** and a square wave with a phase of 180∞ used to represent a binary **0**; or vice versa. This line code is used in Ethernet local area networks (LANs). The waveform for Manchester coding is shown in Fig. 28.1(e). The PSD of a Manchester waveform with equally likely bits is given by [5,8]

$$S_5(f) = V^2T\left(\frac{\sin \pi f T/2}{\pi f T/2}\right)^2 \sin^2(\pi f T/2) \tag{28.11}$$

where $\pm V$ are used as the positive/negative voltage levels for this code. Its spectrum is plotted in Fig. 28.2b. When the probability p of a binary **1** is not equal to one-half, the continuous portion of the PSD is reduced in amplitude and discrete components appear at integer multiples of the bit rate, $R = 1/T$. The resulting

PSD is [6,10]

$$S_{5p}(f) = V^2T4p(1-p)\left(\frac{\sin\pi f T/2}{\pi f T/2}\right)^2 \sin^2\frac{\pi f T}{2} + V^2(1-2p)^2 \sum_{n=-\infty, n\pi 0}^{\infty}\left(\frac{2}{n\pi}\right)^2 \delta(f-nR \qquad (28.12)$$

The first null bandwidth of the waveform generated by a Manchester code is $2R$ Hz. The error rate performance of this waveform when $p = 0.5$ is the same as that of polar NRZ, given by Eq. (28.9), and plotted in Fig. 28.3.

The advantages of this code include a zero DC content on an individual pulse basis, so no pattern of bits can cause DC buildup; midbit transitions are always present making it is easy to extract timing information; and it has good error rate performance, identical to polar NRZ. The main disadvantage of this code is a larger bandwidth than any of the other common codes. Also, it has no error detection capability and, hence, performance monitoring is not possible.

Polar NRZ and Manchester coding are examples of the use of pure polar signalling where the pulse for a binary **0**, $g_2(t)$, is the negative of the pulse for a binary **1**, i.e., $g_2(t) = -g_1(t)$. This is also referred to as an antipodal signal set. For this broad type of polar binary line code, the PSD is given by [10]

$$S_{\mathrm{BP}}(f) = 4p(1-p)R|G(f)|^2 + (2p-1)^2R^2\sum_{n=-\infty}^{\infty}|G(nR)|^2\delta(f-nR) \qquad (28.13)$$

where $|G(f)|$ is the magnitude of the Fourier transform of either $g_1(t)$ or $g_2(t)$.

A further generalization of the PSD of binary line codes can be given, wherein a continuous spectrum and a discrete spectrum is evident. Let a binary **1**, with probability p, be represented by $g_1(t)$ over the $T = 1/R$ second bit interval; and let a binary **0**, with probability $1 - p$, be represented by $g_2(t)$ over the same T second bit interval. The two-sided PSD for this general binary line code is [10]

$$S_{\mathrm{GB}}(f) = p(1-p)R|G_1(f) - G_2(f)|^2 + R^2\sum_{n=-\infty}^{\infty}|pG_1(nR) + (1-p)G_2(nR)|^2\delta(f-nR) \qquad (28.14)$$

where the Fourier transform of $g_1(t)$ and $g_2(t)$ are given by $G_1(f)$ and $G_2(f)$, respectively.

28.3 Alternate Line Codes

Most of the line codes discussed thus far were instantaneous level codes. Only AMI had memory, and Manchester was an instantaneous transition code. The alternate line codes presented in this section all have memory. The first four are transition codes, where binary data is represented as the presence or absence of a transition, or by the direction of transition, i.e., positive to negative or vice versa. The last four codes described in this section are level line codes with memory.

Delay Modulation (Miller Code)

In this line code, a binary **1** is represented by a transition at the midbit position, and a binary **0** is represented by no transition at the midbit position. If a **0** is followed by another **0**, however, the signal transition also occurs at the end of the bit interval, that is, between the two **0**s. An example of delay modulation is shown in Fig. 28.1(f). It is clear that delay modulation is a transition code with memory.

This code achieves the goal of providing good timing content without sacrificing bandwidth. The PSD of the Miller code for equally likely data is given by [10]

$$S_6(f) = \frac{V^2 T}{2(\pi f T)^2 (17 + 8\cos 2\pi f T)}(23 - 2\cos \pi f T - 22\cos 2\pi f T - 12\cos 3\pi f T \\ + 5\cos 4\pi f T + 12\cos 5\pi f T + 2\cos 6\pi f T - 8\cos 7\pi f T + 2\cos 8\pi f T) \tag{28.15}$$

This spectrum is plotted in Fig. 28.2b. The advantages of this code are that it requires relatively low bandwidth and most of the energy is contained in less than 0.5*R*. However, there is no distinct spectral null within the 2*R*-Hz band. It has low DC content and no DC component. It has very good timing content, and carrier tracking is easier than Manchester coding. Error rate performance is comparable to that of the common line codes. One important disadvantage is that it has no error detection capability and, hence, performance cannot be monitored.

Split Phase (Mark)

This code is similar to Manchester in the sense that there are always midbit transitions. Hence, this code is relatively easy to synchronize and has no DC. Unlike Manchester, however, split phase (mark) encodes a binary digit into a midbit transition dependent on the midbit transition in the previous bit period [12]. Specifically, a binary **1** produces a reversal of midbit transition relative to the previous midbit transition. A binary **0** produces no reversal of the midbit transition. Certainly this is a transition code with memory. An example of a split phase (mark) coded waveform is shown in Fig. 28.1(g), where the waveform in the first bit period is chosen arbitrarily. Since this method encodes bits differentially, there is no 180º -phase ambiguity associated with some line codes. This phase ambiguity may not be an issue in most baseband links but is important if the line code is modulated. Split phase (space) is very similar to split phase (mark), where the role of the binary **1** and binary **0** are interchanged. An example of a split phase (space) coded waveform is given in Fig. 28.1(h); again, the first bit waveform is arbitrary.

Biphase (Mark)

This code, designated as Bi ϕ-M, is similar to a Miller code in that a binary **1** is represented by a midbit transition, and a binary **0** has no midbit transition. However, this code always has a transition at the beginning of a bit period [10]. Thus, the code is easy to synchronize and has no DC. An example of Bi ϕ-M is given in Fig. 28.1 (i), where the direction of the transition at $t = 0$ is arbitrarily chosen. Biphase (space) or Bi ϕ-S is similar to Bi ϕ-M, except the role of the binary data is reversed. Here a binary **0** (space) produces a midbit transition, and a binary **1** does not have a midbit transition. A waveform example of Bi ϕ-S is shown in Fig. 28.1(j). Both Bi ϕ-S and Bi ϕ-M are transition codes with memory.

Code Mark Inversion (CMI)

This line code is used as the interface to a Consultative Committee on International Telegraphy and Telephony (CCITT) multiplexer and is very similar to Bi ϕ-S. A binary **1** is encoded as an NRZ pulse with alternate polarity, +*V* or -*V*. A binary **0** is encoded with a definitive midbit transition (or square wave phase) [1]. An example of this waveform is shown in Fig. 28.1(k) where a negative to positive transition (or 180∞ phase) is used for a binary **0**. The voltage level of the first binary **1** in this example is chosen arbitrarily. This example waveform is identical to Bi ϕ-S shown in Fig. 28.1(j), except for the last bit. CMI has good synchronization properties and has no DC.

NRZ (I)

This type of line code uses an inversion (I) to designate binary digits, specifically, a change in level or no change in level. There are two variants of this code, NRZ mark (M) and NRZ space (S) [5,12]. In NRZ (M), a change of level is used to indicate a binary **1**, and no change of level is used to indicate a binary **0**. In NRZ (S) a change of level is used to indicate a binary **0**, and no change of level is used to indicate a binary **1**. Waveforms for NRZ (M) and NRZ (S) are depicted in Fig. 28.1(l) and Fig. 28.1(m), respectively, where the voltage level of the first binary **1** in the example is chosen arbitrarily. These codes are level codes with memory. In general, line codes that use differential encoding, like NRZ (I), are insensitive to 180∞ phase ambiguity. Clock recovery with NRZ (I) is not particularly good, and dc wander is a problem as well. Its bandwidth is comparable to polar NRZ.

Binary *N* Zero Substitution (BNZS)

The common bipolar code AMI has many desirable properties of a line code. Its major limitation, however, is that a long string of zeros can lead to loss of synchronization and timing jitter because there are no pulses in the waveform for relatively long periods of time. **Binary *N* zero substitution (BNZS)** attempts to improve AMI by substituting a special code of length *N* for all strings of *N* zeros. This special code contains pulses that look like binary **1**s but purposely produce violations of the AMI pulse convention. Two consecutive pulses of the same polarity violate the AMI pulse convention, independent of the number of zeros between the two consecutive pulses. These violations can be detected at the receiver and the special code replaced by *N* zeros. The special code contains pulses facilitating synchronization even when the original data has a long string of zeros. The special code is chosen such that the desirable properties of AMI coding are retained despite the AMI pulse convention violations, i.e., DC balance and error detection capability. The only disadvantage of BNZS compared to AMI is a slight increase in crosstalk due to the increased number of pulses and, hence, an increase in the average energy in the code.

Choosing different values of *N* yields different BNZS codes. The value of *N* is chosen to meet the timing requirements of the application. In telephony, there are three commonly used BNZS codes: B6ZS, B3ZS, and B8ZS. All BNZS codes are level codes with memory.

In a B6ZS code, a string of six consecutive zeros is replaced by one of two the special codes according to the rule:

$$\text{If the last pulse was positive } (+)\text{, the special code is:} \quad 0 + - 0 - +.$$
$$\text{If the last pulse was negative } (-)\text{, the special code is:} \quad 0 - + 0 + -.$$

Here a zero indicates a zero voltage level for the bit period; a plus designates a positive pulse, and a minus indicates a negative pulse.

This special code causes two AMI pulse violations: in its second bit position and in its fifth bit position. These violations are easily detected at the receiver and zeros resubstituted. If the number of consecutive zeros is 12, 18, 24,…, the substitution is repeated 2, 3, 4,… times. Since the number of violations is even, the B6ZS waveform is the same as the AMI waveform outside the special code, i.e., between special code sequences.

There are four pulses introduced by the special code that facilitates timing recovery. Also, note that the special code is DC balanced. An example of the B6ZS code is given as follows, where the special code is indicated by the bold characters.

Original data:	0	1	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	1	1
B6ZS format:	0	+	**0**	+	-	**0**	-	+	-	+	0	-	**0**	-	+	**0**	+	-	+	-

The computation of the PSD of a B6ZS code is tedious. Its shape is given in Fig. 28.4, for comparison purposes with AMI, for the case of equally likely data.

TABLE 28.1 B3ZS Substitution Rules

Number of B Pulses Since Last Violation	Polarity of Last B Pulse	Substitution Code	Substitution Code Form
Odd	Negative (-)	**00−**	**00**V
Odd	Positive (+)	**00+**	**00**V
Even	Negative (-)	**+0+**	B**0**V
Even	Positive (+)	**−0−**	B**0**V

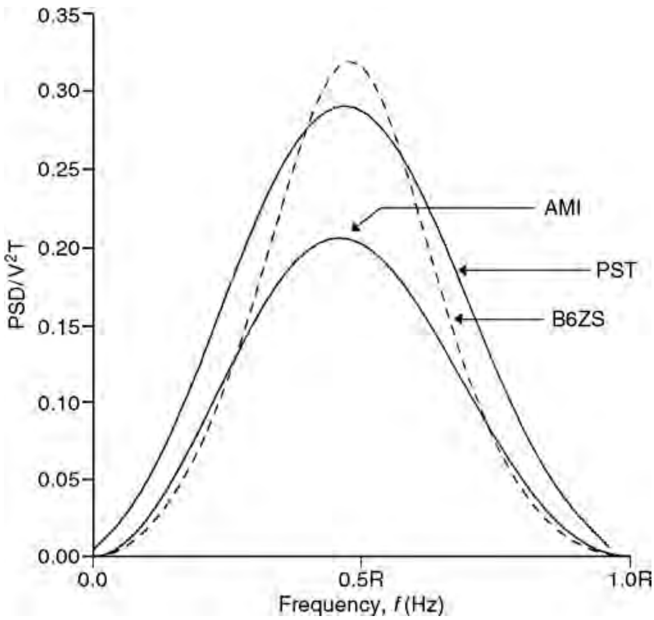

FIGURE 28.4 Power spectral density of different line codes, where $R = 1/T$ is the bit rate.

In a B3ZS code, a string of three consecutive zeros is replaced by either B**0**V or **00V**, where B denotes a pulse obeying the AMI (bipolar) convention and V denotes a pulse violating the AMI convention. B**0**V or **00**V is chosen such that the number of bipolar (B) pulses between the violations is odd. The B3ZS rules are summarized in Table 28.1.

Observe that the violation always occurs in the third bit position of the substitution code, and so it can be easily detected and zero replacement made at the receiver. Also, the substitution code selection maintains DC balance. There is either one or two pulses in the substitution code, facilitating synchronization. The error detection capability of AMI is retained in B3ZS because a single channel error would make the number of bipolar pulses between violations even instead of being odd. Unlike B6ZS, the B3ZS waveform between violations may not be the same as the AMI waveform. B3ZS is used in the digital signal-3 (DS-3) signal interface in North America and also in the long distance-4 (LD-4) coaxial transmission system in Canada. Next is an example of a B3ZS code, using the same symbol meaning as in the B6ZS code.

Original data:	1	0	0	1	0	0	0	1	1	0	0	0	0	1	0	0	0	1
B3ZS format:																		
Even No. of B pulses:	+	0	0	-	+	**0**	+	-	+	-	**0**	-	0	+	**0**	**0**	+	-
Odd No. of B pulses:	+	0	0	-	**0**	**0**	-	+	-	+	**0**	+	0	-	**0**	**0**	-	+

The last BNZS code considered here uses $N = 8$. A B8ZS code is used to provide transparent channels for the Integrated Services Digital Network (ISDN) on T1 lines and is similar to the B6ZS code. Here a

TABLE 28.2 HDB3 Substitution Rules

Number of B Pulses Since Last Violation	Polarity of Last B Pulse	Substitution Code	Substitution Code Form
Odd	Negative (-)	**000−**	**000**V
Odd	Positive (+)	**000+**	**000**V
Even	Negative (-)	**+00+**	B**00**V
Even	Positive (+)	**−00−**	B**00**V

string of eight consecutive zeros is replaced by one of two special codes according to the following rule:

If the last pulse was positive (+), the special code is: 0 0 0 + - 0 - +.

If the last pulse was negative (-), the special code is: 0 0 0 - + 0 + -.

There are two bipolar violations in the special codes, at the fourth and seventh bit positions. The code is DC balanced, and the error detection capability of AMI is retained. The waveform between substitutions is the same as that of AMI. If the number of consecutive zeros is 16, 24,…, then the substitution is repeated 2, 3,…, times.

High-Density Bipolar *N* (HDBN)

This coding algorithm is a CCITT standard recommended by the Conference of European Posts and Telecommunications Administrations (CEPT), a European standards body. It is quite similar to BNZS coding. It is thus a level code with memory. Whenever there is a string of $N + 1$ consecutive zeros, they are replaced by a special code of length $N + 1$ containing AMI violations. Specific codes can be constructed for different values of N. A specific **high-density bipolar *N* (HDBN)** code, HDB3, is implemented as a CEPT primary digital signal. It is very similar to the B3ZS code. In this code, a string of four consecutive zeros is replaced by either B**00**V or **000**V. B**00**V or **000*V*** is chosen such that the number of bipolar (B) pulses between violations is odd. The HDB3 rules are summarized in Table 28.2.

Here the violation always occurs in the fourth bit position of the substitution code, so that it can be easily detected and zero replacement made at the receiver. Also, the substitution code selection maintains DC balance. There are either one or two pulses in the substitution code facilitating synchronization. The error detection capability of AMI is retained in HDB3 because a single channel error would make the number of bipolar pulses between violations even instead of being odd.

Ternary Coding

Many line coding schemes employ three symbols or levels to represent only one bit of information, like AMI. Theoretically, it should be possible to transmit information more efficiently with three symbols, specifically the maximum efficiency is $\log_2 3 = 1.58$ bits per symbol. Alternatively, the redundancy in the code signal space can be used to provide better error control. Two examples of ternary coding are described next [1,2]: **pair selected ternary (PST)** and **4 binary 3 ternary (4B3T)**. The PST code has many of the desirable properties of line codes, but its transmission efficiency is still 1 bit per symbol. The 4B3T code also has many of the desirable properties of line codes, and it has increased transmission efficiency.

In the PST code, two consecutive bits, termed a binary pair, are grouped together to form a word. These binary pairs are assigned codewords consisting of two ternary symbols, where each ternary symbol can be +, -, or 0, just as in AMI. There are nine possible ternary codewords. Ternary codewords with identical elements, however, are avoided, i.e., ++, --, and 00. The remaining six codewords are transmitted using two modes called + mode and - mode. The modes are switched whenever a codeword with a single pulse is transmitted. The PST code and mode switching rules are summarized in Table 28.3.

PST is designed to maintain DC balance and include a strong timing component. One drawback of this code is that the bits must be framed into pairs. At the receiver, an *out-of-frame* condition is signalled

TABLE 28.3 PST Codeword Assignment and Mode Switching Rules

Binary Pair	Ternary Codewords +Mode	Ternary Codewords -Mode	Mode Switching
11	+ -	+ -	No
10	+ 0	- 0	Yes
01	0 +	0 -	Yes
00	- +	- +	No

TABLE 28.4 Modified PST Codeword Assignment and Mode Switching Rules

Binary Pair	Ternary Codewords + Mode	Ternary Codewords - Mode	Mode Switching
11	+ 0	0 -	Yes
10	0 -	0 -	No
01	- +	- +	No
00	0 +	- 0	Yes

TABLE 28.5 4B3T Codeword Assignment

Binary Words	Ternary Codewords Column 1	Ternary Codewords Column 2	Ternary Codewords Column 3
0000	- - -		+ + +
0001	- - 0		+ + 0
0010	- 0 -		+ 0 +
0011	0 - -		0 + +
0100	- - +		+ + -
0101	- + -		+ - +
0110	+ - -		- + +
0111	- 0 0		+ 0 0
1000	0 - 0		0 + 0
1001	0 0 -		0 0 +
1010		0 + -	
1011		0 - +	
1100		+ 0 -	
1101		- 0 +	
1110		+ - 0	
1111		- + 0	

when unused ternary codewords (++, --, and 00) are detected. The mode switching property of PST provides error detection capability. PST can be classified as a level code with memory.

If the original data for PST coding contains only **1**s or **0**s, an alternating sequence of +- +- … is transmitted. As a result, an out-of-frame condition cannot be detected. This problem can be minimized by using the modified PST code as shown in Table 28.4.

It is tedious to derive the PSD of a PST coded waveform. Again, Fig. 28.4 shows the PSD of the PST code along with the PSD of AMI and B6ZS for comparison purposes, all for equally likely binary data. Observe that PST has more power than AMI and, thus, a larger amount of energy per bit, which translates into slightly increased crosstalk.

In 4B3T coding, words consisting of four binary digits are mapped into three ternary symbols. Four bits imply $2^4 = 16$ possible binary words, whereas three ternary symbols allow $3^3 = 27$ possible ternary codewords. The binary-to-ternary conversion in 4B3T insures DC balance and a strong timing component. The specific codeword assignment is as shown in Table 28.5.

There are three types of codewords in Table 28.5, organized into three columns. The codewords in the first column have negative DC, codewords in the second column have zero DC, and those in the third column have positive DC. The encoder monitors the integer variable

$$I = N_p - N_n \tag{28.16}$$

where N_p is the number of positive pulses transmitted and N_n are the number of negative pulses transmitted. Codewords are chosen according to following rule:

If $I < 0$, choose the ternary codeword from columns 1 and 2.
If $I > 0$, choose the ternary codeword from columns 2 and 3.
If $I = 0$, choose the ternary word from column 2, and from column 1 if the previous $I > 0$ or from column 3 if the previous $I < 0$.

Note that the ternary codeword 000 is not used, but the remaining 26 codewords are used in a complementary manner. For example, the column 1 codeword for 0001 is --0, whereas the column 3 codeword is ++0. The maximum transmission efficiency for the 4B3T code is 1.33 bits per symbol compared to 1 bit per symbol for the other line codes. The disadvantages of 4B3T are that framing is required and that performance monitoring is complicated. The 4B3T code is used in the T148 span line developed by ITT Telecommunications. This code allows transmission of 48 channels using only 50% more bandwidth than required by T1 lines, instead of 100% more bandwidth.

28.4 Multilevel Signalling, Partial Response Signalling, and Duobinary Coding

Ternary coding, such as 4B3T, is an example of the use of more than two levels to improve the transmission efficiency. To increase the transmission efficiency further, more levels and/or more signal processing is needed. Multilevel signalling allows an improvement in the transmission efficiency at the expense of an increase in the error rate, i.e., more transmitter power will be required to maintain a given probability of error. In partial response signalling, intersymbol interference is deliberately introduced by using pulses that are wider and, hence, require less bandwidth. The controlled amount of interference from each pulse can be removed at the receiver. This improves the transmission efficiency, at the expense of increased complexity. **Duobinary coding**, a special case of partial response signalling, requires only the minimum theoretical bandwidth of $0.5R$ Hz. In what follows, these techniques are discussed in slightly more detail.

Multilevel Signalling

The number of levels that can be used for a line code is not restricted to two or three. Since more levels or symbols allow higher transmission efficiency, multilevel signalling can be considered in bandwidth-limited applications. Specifically, if the signalling rate or baud rate is R_s and the number of levels used is L, the equivalent transmission bit rate R_b is given by

$$R_b = R_s \log_2 [L] \tag{28.17}$$

Alternatively, multilevel signalling can be used to reduce the baud rate, which in turn can reduce crosstalk for the same equivalent bit rate. The penalty, however, is that the SNR must increase to achieve the same error rate. The T1G carrier system of AT&T uses multilevel signalling with $L = 4$ and a baud rate of 3.152 mega-symbols/s to double the capacity of the T1C system from 48 channels to 96 channels. Also, a four level signalling scheme at 80-kB is used to achieve 160 kb/s as a basic rate in a digital subscriber loop (DSL) for ISDN.

Partial Response Signalling and Duobinary Coding

This class of signalling is also called *correlative* coding because it purposely introduces a controlled or correlated amount of intersymbol interference in each symbol. At the receiver, the known amount of interference is effectively removed from each symbol. The advantage of this signalling is that wider pulses can be used requiring less bandwidth, but the SNR must be increased to realize a given error rate. Also, errors can propagate unless *precoding* is used.

There are many commonly used partial response signalling schemes, often described in terms of the delay operator D, which represents one signalling interval delay. For example, in (1 + D) signalling the current pulse and the previous pulse are added. The T1D system of AT&T uses (1 + D) signalling with precoding, referred to as duobinary signalling, to convert binary (two level) data into ternary (three level) data at the same rate. This requires the minimum theoretical channel bandwidth without the deleterious effects of intersymbol interference and avoids error propagation. Complete details regarding duobinary coding are found in Lender [9] and Schwartz [11]. Some partial response signalling schemes, such as $(1 - D)$, are used to shape the bandwidth rather than control it. Another interesting example of duobinary coding is a $(1 - D^2)$, which can be analyzed as the product $(1 - D)(1 + D)$. It is used by GTE in its modified T carrier system. AT&T also uses $(1 - D^2)$ with four input levels to achieve an equivalent data rate of 1.544 Mb/s in only a 0.5-MHz bandwidth.

28.5 Bandwidth Comparison

We have provided the PSD expressions for most of the commonly used line codes. The actual bandwidth requirement, however, depends on the pulse shape used and the definition of bandwidth itself. There are many ways to define bandwidth, for example, as a percentage of the total power or the sidelobe suppression relative to the main lobe. Using the first null of the PSD of the code as the definition of bandwidth, Table 28.6 provides a useful bandwidth comparison.

The notable omission in Table 28.6 is delay modulation (Miller code). It does not have a first null in the $2R$-Hz band, but most of its power is contained in less than $0.5R$ Hz.

28.6 Concluding Remarks

An in-depth presentation of line coding, particularly applicable to telephony, has been included in this chapter. The most desirable characteristics of line codes were discussed. We introduced five common line codes and eight alternate line codes. Each line code was illustrated by an example waveform. In most cases expressions for the PSD and the probability of error were given and plotted. Advantages and disadvantages of all codes were included in the discussion, and some specific applications were noted. Line codes for optical fiber channels and networks built around them, such as fiber distributed data interface (FDDI), were not included in this section. A discussion of line codes for optical fiber channels and other new developments in this topic area can be found in [1,3,4].

TABLE 28.6 First Null Bandwidth Comparison

Bandwidth	Codes	
R	Unipolar NRZ Polar NRZ Polar RZ (AMI)	BNZS HDBN PST
$2R$	Unipolar RZ Manchester	Split Phase CMI

Defining Terms

Alternate mark inversion (AMI): A popular name for bipolar line coding using three levels: zero, positive, and negative.

Binary *N* zero substitution (BNZS): A class of coding schemes that attempts to improve AMI line coding.

Bipolar: A particular line coding scheme using three levels: zero, positive, and negative.

Crosstalk: An unwanted signal from an adjacent channel.

DC wander: The DC level variation in the received signal due to a channel that cannot support DC.

Duobinary coding: A coding scheme with binary input and ternary output requiring the minimum theoretical channel bandwidth.

4 Binary 3 Ternary (4B3T): A line coding scheme that maps four binary digits into three ternary symbols.

High-density bipolar *N* (HDBN): A class of coding schemes that attempts to improve AMI.

Level codes: Line codes carrying information in their voltage levels.

Line coding: The process of converting abstract symbols into real, temporal waveforms to be transmitted through a baseband channel.

Nonreturn to zero (NRZ): A signal that stays at a nonzero level for the entire bit duration.

Pair selected ternary (PST): A coding scheme based on selecting a pair of three level symbols.

Polar: A line coding scheme using both polarity of voltages, with or without a zero level.

Return to zero (RZ): A signal that returns to zero for a portion of the bit duration.

Transition codes: Line codes carrying information in voltage level transitions.

Unipolar: A line coding scheme using only one polarity of voltage, in addition to a zero level.

References

1. Bellamy, J., *Digital Telephony,* John Wiley & Sons, New York, NY, 1991.
2. Bell Telephone Laboratories Technical Staff Members. *Transmission Systems for Communications,* 4th ed., Western Electric Company, Technical Publications, Winston-Salem, NC, 1970.
3. Bic, J.C., Duponteil, D., and Imbeaux, J.C., *Elements of Digital Communication,* John Wiley & Sons, New York, NY, 1991.
4. Bylanski, P., *Digital Transmission Systems,* Peter Peregrinus, Herts, England, 1976.
5. Couch, L.W., *Modern Communication Systems: Principles and Applications,* Prentice-Hall, Englewood Cliffs, NJ, 1994.
6. Feher, K., *Digital Modulation Techniques in an Interference Environment,* EMC Encyclopedia Series, Vol. IX. Don White Consultants, Germantown, MD, 1977.
7. Gibson, J.D., *Principles of Analog and Digital Communications,* MacMillan Publishing, New York, NY, 1993.
8. Lathi, B.P., *Modern Digital and Analog Communication Systems,* Holt, Rinehart and Winston, Philadelphia, PA, 1989.
9. Lender, A., Duobinary Techniques for High Speed Data Transmission, *IEEE Trans. Commun. Electron.*, CE-82, 214–218, May 1963.
10. Lindsey, W.C. and Simon, M.K., *Telecommunication Systems Engineering,* Prentice-Hall, Englewood Cliffs, NJ, 1973.
11. Schwartz, M., *Information Transmission, Modulation, and Noise,* McGraw-Hill, New York, NY, 1980.
12. Stremler, F.G., *Introduction to Communication Systems,* Addison-Wesley Publishing, Reading, MA, 1990.