Experiment 8: Line Coding

Aim: This experiment is intended to make the students generate line codes corresponding to random bit sequences and examine their time- and frequency-domain properties.

Basic Theory

Logic-0 and logic-1 in digital systems are represented by assigned voltages. For example, the TTL logic-0 is represented by 0V and the TTL logic-1 is represented by 5V (or by acceptable voltages relatively close to 0V and 5V). The voltage levels for other logic families such as CMOS and ECL are not necessarily 0V and 5V. This tells us that logic levels can be represented by any pair of voltages we like. That said, the choice of which voltages to use is not as arbitrary as that may seem. It is usually an engineering decision made to confer an advantage.

Importantly, this is also true for the choice of voltages used when sending digital signals over transmission lines like telephone lines. Standard TTL and CMOS voltages are less than ideal for this purpose. Moreover, even the basic premise of holding the voltage at a particular value for the entire duration of the logic state's value can be disadvantageous. For these reasons, digital signals within systems are often conditioned for transmission line communications and this is called *line coding*.

There are quite a few line codes. We will generate and study the following:

- **Non-return to zero – level (bipolar) (NRZ-L):** As you can see from Figure 1 on the next page, this code is a simple scale and level shift of the original digital signal.

- **Bi-phase – level (BiΦ-L also known as Manchester code):** Figure 1 shows that this code changes state from +V to –V in the middle of the bit period for all logic-1s and changes from –V to +V in the middle of the bit period for all logic-0s. For consecutive bits with the same logic level, the voltage must invert after half a bit length in order to satisfy this rule for the next bit.

- **Return to zero – alternate mark inversion (RZ-AMI):** Figure 1 shows that this code uses 0V to represent logic-0 and a half-bit pulse to represent logic-1. Importantly, the polarity of the pulses alternates for every successive logic-1 (even if they're not consecutive bits).

- **Non-return to zero – mark (bipolar) (NRZ-M):** Figure 1 shows that this code changes state for each new logic-1 and doesn't change state for any logic-0s.

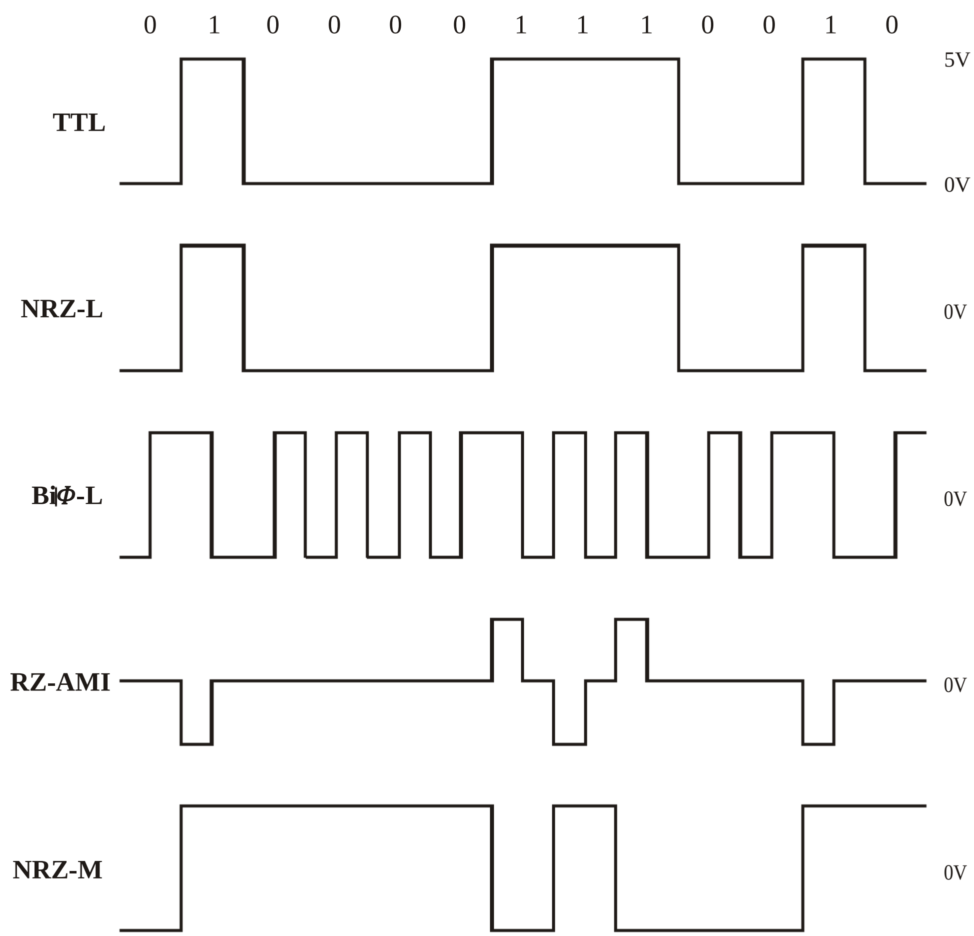

Figure 1

Table 1 compares the minimum bandwidth requirements for propagating these signals along transmission lines. It also shows the line code's usefulness for bit-clock regeneration. As you can see from the table, RZ-AMI offers the best compromise among the above between bandwidth and bit-clock regeneration (as well as other line code characteristics not mentioned here) and so it is widely used.

Table 1

Line code	**Minimum Bandwidth**	**Bit-clock regeneration**
NRZ-L	R_b	Poor
BiΦ-L	$2R_b$	Very good
RZ-AMI	R_b	Good
NRZ-M	R_b	Poor

A – Observations on the random bit pattern in both Time and Frequency domains

1. Generate a data string of 10 random bits. To generate random integer string of length N within the range I_{min} and I_{max}, use the MATLAB command

   randi([I_{min}, I_{max}],1,N)
2. To represent this bit string as a transmittable signal, let the bit rate (R_b) be 10 bits per second. What is the bit period?
3. Represent this data using NRZ-L, Bi*Φ*-L, RZ-AMI and NRZ-M line codes. Assume V = 1 Volt for each of these. Plot the line codes corresponding to the generated random data one below the other in a single figure.
4. For each of the line codes, plot the spectrum one below the other in a single figure. From this, for each line code, note down the frequency at which the first deep null occurs. If we call this the essential bandwidth of the line code, what is the bandwidth for each line code? Is this related to the bit rate? What is the ratio of the bandwidth to the bit rate for each line code? Find the power at 0 Hz and at the first deep null. Tabulate the results separately for each line coding using the table format given in Table 2.

Table 2: Time & Spectral Domain Properties of line codes

Digital Bit	**Line code Voltage**	**Bit Rate R_b**	**Power At 0 Hz**	**First Deep Null Frequency**	**Power at First deep null**	**Essential BW / R_b**
1						
0						

B – Effect of Bandwidth Limiting of channels

The bandwidth limiting in a channel can distort digital signals and upset the operation of the receiver. This part of the experiment demonstrates this using line coded digital data passing through a channel, represented by a low pass filter.

Pass each of the line codes through an LPF. For each line code, vary the cut-off frequency across a sufficiently large range of frequencies. You can use the cut-off frequencies in multiple of the bit rate. Plot the filtered output below the corresponding line coded signal. Describe your observations. For performing the filtering operation, use the "butter" command to obtain the filter coefficients of a Butterworth filter (of order 5 or above) and use these coefficients in the "filter" command. Do not use the "lowpass" command for this experiment.

C – Detection of line coded signals corrupted by bandlimited channels

Next, we decode the line-coded signal that is transmitted via a bandlimited channel. The detection mechanism varies from one line code to another. For NRZ-L and NRZ-M, we detect the bit by first sampling it at the middle of the bit period (that is, at time instant $n\,T_b$ for the n^{th} bit) and checking if the received signal at this time instant is greater than $\frac{V_{max}+V_{min}}{2}$ Volts or lesser. Here, V_{max} and V_{min} are the voltage levels assigned to logic-1 and logic-0 respectively. In contrast, due to split-phase coding used in BiΦ-L and RZ-AMI, the signal needs to be sampled twice in a bit period. Specifically, for the n^{th} bit, it needs to be sampled at $\frac{n\,T_b}{4}$ and $\frac{3\,n\,T_b}{4}$. As before, each of these samples are compared with $\frac{V_{max}+V_{min}}{2}$ before a detection decision is appropriately taken.

For each of the line code, detect the line-coded signal and note down the error rate, which is the ratio of wrongly decoded bits to the total number of transmitted bits. Compute the error rate for various cut-off frequencies of the LPF. Note down your observations.

D – Conclusions:

List out your learnings from the experiments.