Part 1
Performance of Matched filters and correlators

Objective

(1) Investigate the matched filter and correlator receivers

(2) Modify the system of simple detector to use a matched filter receiver instead of simple sampler.

(3) Compare the performance of simple detector and matched filter receiver.

Theoretical Background

(1) Matched filter receivers and correlators:

Matched filter is baseband filter which is designed as function of transmitted waveforms to maximize the received SNR at the designed sampling instant and hence minimizes the resultant probability of error and considered the optimal receiver in case of AWGN channel.
The matched filter is designed in case of AWGN channel as

$$h_{mf}(t) = c\ (\ S_1(T-t) - \ S_2(T-t)\ )$$

And the output of the matched filter is the convolution between $h_{mf}(t)$ and received signal to have a MF output that will be further sampled to be the input of the simple detector.
To ease the design of implementing of the MF, the correlator is proposed. The correlator consists of multiplier and integrator, it will multiply by g(t) which equals to

$$g(t) = S_1(t) - \ S_1(t)$$

The received signal is multiplied by this g(t) and then accumulated to have a value that will be compared with the threshold.
The correlator receiver output equals the MF output at the sampling instant. The threshold at this case can be calculated as

$$V_{th} = \frac{S_1(T) + S_1(T)}{2} = \text{c}\ \frac{E_1 - E_2}{2}$$

(2) Modeling Matched filter receiver

To model the Matched filter receiver using MATLAB, you should use the sampled MF version, That will result in sampling $h_{mf}(t)$ by specified number of samples (example:10 samples), the transmitted signal itself should be represented by same number of samples, then you can perform convolution at the receiver and choose the sample at any time (but not the middle) which represents the sampling process.

Procedure

(1) Simulation parameters

a. Number of bits/SNR=1e5 bits

b. Signal to noise ratio range=0 to 30 dB with 2 dB steps.

c. Number of samples that represents waveform $m = 20$

d. Sampling instant =20

e. Receiver type: Matched filter and correlator.

f. $s_1(t)$ is rectangular signal with Amp=1 and $s_2(t)$ is zero signal.

(2) Generate random binary data vector (you can make use of `randint` or `randi`).

(3) Represent each bit with proper waveform (hint: use concatenation, notice that the resultant vector will be 10e6 samples)

(4) Apply noise to samples (Hint: For fair comparison between MF and simple detector you should equate the average power in the two cases, and hence you should notice that the power of signal in case of MF is calculated by adding the squares of all samples that represents $s_1(t)$ and hence SNR will change).

```
Rx_sequence=bits+noise
```

Or

```
Rx_sequence=awgn(bits,snr,'measured')
```

(5) Apply convolution process in the receiver (Hint: the convolution process will be performed in bit-by-bit basis i.e. 20 samples by 20 samples) You may use `conv`
In case of correlator, you will apply element by element multiplication and then add the resultant samples together.

(6) Sample the output of the Matched filter (i.e. choose the middle sample, you may use indexing tools)

(7) Decide whether the Rx_sequence is '1' or '0' by comparing the samples with threshold (Hint: try to use relational operators and indexing to make the code more efficient)

(8) Compare the original bits with the detected bits and calculate number of errors (you can make use of `xor` or `biterr`).

(9) Save the probability of error of each SNR in matrix, BER

(10) Plot the BER curve against SNR (use `semilogy`)

Report requirement

(1) Well commented M-file.

(2) Softcopy report containing required figures (MF and correlator and its comparison to simple detector), comment.

(3) Calculation of transmitted signal power

(4) At which value of SNR the system is nearly without error (for the given frame)?

(5) The code should run in a reasonable time (around few minutes) (Hint: try to reduce the number of loops as much as possible and try to consider preallocation).

(6) Generalize the program to acquire any number of samples that represents $s_1(t)$, and any sampling instant, compare performance of having 10,20,100 samples

(7) Generalize program to have $s_1(t)$ and $s_2(t)$ as general waveforms or user defined.

Part 2

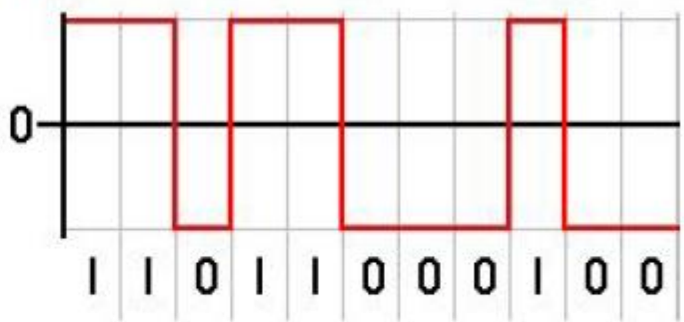
Line code

Objective:
Compare the different types of line codes used in digital communications.

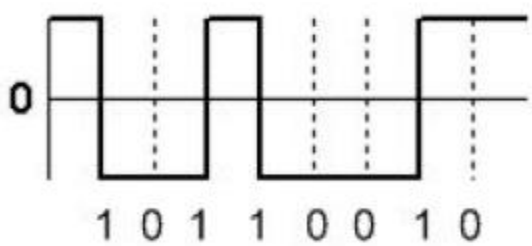
Theoretical Background:
(1) Line codes are also called digital baseband modulation.
(2) There are many different types of line codes and each one of them has an advantage from a certain point of view.

(3) The main types are:

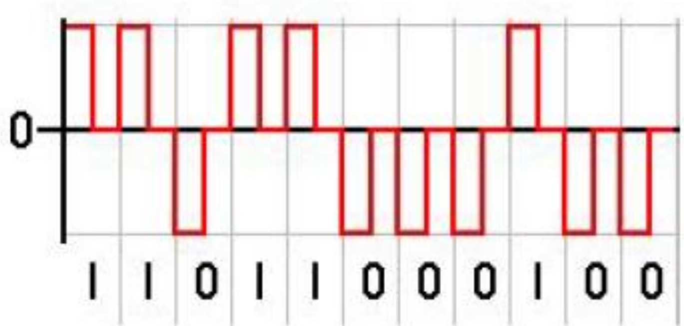
i- Non-return to zero.



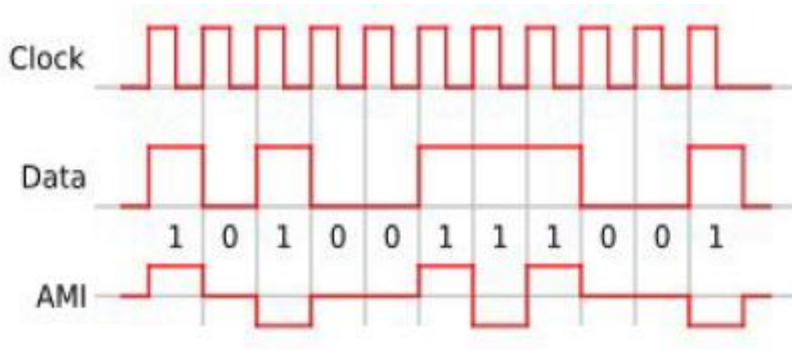
ii- Non-return to zero inverted.



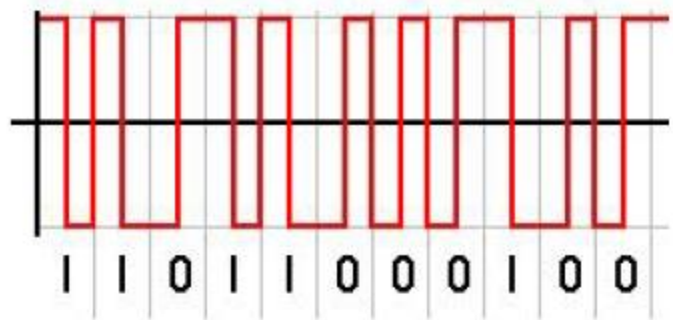
iii- Return to zero.



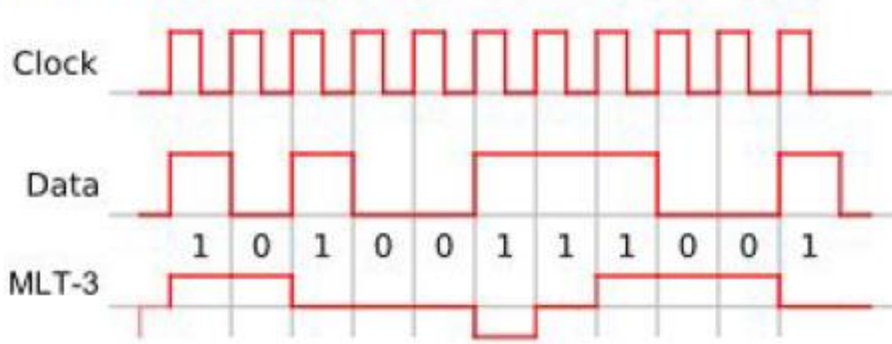
(iv) Alternative mark inversion (AMI)



v- Manchester coding



vi- Multi-level transmission 3



Procedure:

(1) Generate random bits of zeros and ones.
(2) Modulate this same vector using the different types of line codes.
(3) Plot a sample of all previous line code modulation and plot them under each other on the same figure using subplot, ex subplot (6,1), (make sure that all the types have the same periodic time Ts).
(4) Find the power spectrum density of each code, and plot them in the same figure using subplot as previous.
(5) Comment on each of the figures.

Report requirement:

(1) Well commented M-file.
(2) Softcopy report containing required figures and the comments on each figure.
(3) Which type of signal has the highest bandwidth? Comment.
(4) Mention the advantages and disadvantages of each line code.
(5) Mention in the report 2 other used line codes and explain them mentioning the main advantages or disadvantages.