1 Labwork

1.1 Sampling

a. Construct a signal $x[n] = \sum_{i=1}^{C} A_i sin[2\pi f_i n + \theta_i]$ using the following parameters with a sampling frequency of $F_s = 5$ KHz and signal duration of 0.2 seconds.

- $C = 3$ (Number of frequency components)
- $A_1 = 1, A_2 = 3, A_3 = 2$
- $f_1 = 60, f_2 = 20, f_3 = 150$
- $\theta_1 = 0, \theta_2 = \pi/2, \theta_3 = \pi/4$

Fig 1

Define the parameters in a vector form so that you can add new components by simply increasing the vector size. Plot the signal $x[n]$ using 2D continuous plot function *plot()*.

b. Downsample the signal $x[n]$ by 10 and 20 to obtain signals $x_{s1}[n]$ and $x_{s2}[n]$.

Fig 2

c. Plot the downsampled signals $x_{s1}[n]$ and $x_{s2}[n]$ using two *stem()* function inside 2×1 *subplot()*.

d. Compute and plot the normalized frequency spectrum of the signals $x_{s1}[n]$ and $x_{s2}[n]$ using two *plot()* functions inside 2×1 *subplot()*. Display only the magnitude.
Comment on the followings in your code just before the line where you call *fft()*.

Fig 3

- Explain how did you decide the frequency axis while plotting frequency spectrum.
- Why did we normalize frequency spectrum?

Answer the following questions in your report.

Report-Q1. What are the factors that affect the frequency resolution of the spectrum.

Report-Q2. Why do we observe the 3^{rd} component ($150Hz$) of $x_{s2}[n]$ on ($100Hz$). Is it possible to predict observed location of another component which is at $180Hz$ without adding such component. Give your prediction with sufficient explanation.

Fig 4
Fig 5.

e. Reconstruct the original signal by using linear and cubic interpolation techniques. Plot original function and signals reconstructed from $x_{s1}[n]$ by using linear and cubic interpolation in the same axis. In another figure, plot original function and signals reconstructed from $x_{s2}[n]$ by using linear and cubic interpolation in the same axis.

Report-Q3 Compare the resulting signals for each method and write your comments on your report.

Report-Q4 How would you compute linear interpolation without using any built-in interpolation function?

1.2 Quantization

a. Construct a signal $x_2[n] = \sum_{i=1}^{C} A_i sin[2\pi f_i n + \theta_i]$ using the following parameters with a sampling frequency of $F_s = 5$ KHz and signal duration of 0.2 seconds.

- $C = 5$ (Number of frequency components)
- $A_1 = 5, A_2 = 4, A_3 = 3, A_4 = 2, A_5 = 1$

Downsample Sal??

6

- $f_1 = 50, f_2 = 60, f_3 = 70, f_4 = 80, f_5 = 90$
- $\theta_1 = 0, \theta_2 = \pi/5, \theta_3 = 2\pi/5, \theta_4 = 3\pi/5, \theta_5 = 4\pi/5$

1x.0

Define the parameters again in a vector form. Plot the signal $x_2[n]$ using 2D continuous plot function *plot()*.

b. Apply uniform quantization with a number of bits of N=4 and N=8 to generate quantized signals x_{q4} and x_{q8} using the following formula

$$X_{qN} = \left\lfloor \frac{x-a}{b-a} * (2^N - 1) \right\rfloor * \frac{b-a}{2^N - 1} + a \tag{1}$$

where a, b and N stand for minimum value of the signal, maximum value of the signal and number of bits to represent each value, respectively.

1x
4x.0

c. Plot x_{q4} and x_{q8} on top of $x_2[n]$ in 2x1 *subplot()* using proper legends labels and coloring.

d. Compute Mean Squared Error (MSE) and Signal to Quantization-Noise Ratio (SQNR) in linear scale and in decibels (dB), respectively. Compute MSE and SQNR terms for both 4-bit and 8-bit quantized signals. Use following formulas

$$MSE = E[(X - X_Q)^2] \tag{2}$$

$$SQNR = \frac{E[(X^2)]}{E[(X - X_Q)^2]} \tag{3}$$

where E, X, X_Q represent expected value, original signal and quantized signal, respectively. **Do not forget to convert ~~SQNR in dB.~~**

Report-Q5 Comment on the results of MSE and SQNR values of x_{q4} and x_{q8}.

To Do

✓Figürleri Düzelt

✓Comment Yaz

✓Değerleri kontrol et

✓Kodu kontrol et sınırları ok..

Downsample lol?

1 Labwork

a. Construct a message signal $m(t) = A_m \cos(2\pi f_m t)$ where $A_m = 2$ V and $f_m = 4\,\text{Hz}$. The sampling rate is $F_s = 80\,\text{Hz}$ and the duration of the signal is 1 s.

b. Compute the minimum required step size (Δ_{min}) to avoid **slope overload distortion**.
<u>Hint:</u> Remember the condition about the maximum slope of the message signal.

c. Write a code which performs the block diagram in Figure 1. Use the step size $\Delta = \Delta_{min}$ that you have just computed in the previous step.

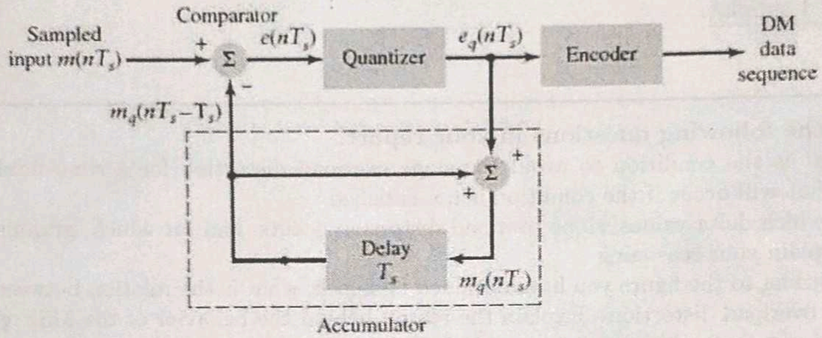

Figure 1: Delta Modulator

The set of equations related to the delta modulator are given by

$$\begin{aligned} m_q(nT_s) &= m_q(nT_s - T_s) + e_q(nT_s), \\ e(nT_s) &= m(nT_s) - m_q(nT_s - T_s), \\ e_q(nT_s) &= \Delta\, sgn\,(e(nT_s))\,. \end{aligned} \tag{1}$$

where T_s is the sampling period, $m_q(nT_s)$ is the staircase approximation of the sampled input $m(nT_s)$, $e(nT_s)$ is the error signal, $e_q(nT_s)$ is the quantized error signal, and $sgn(.)$ is the signum function.
<u>Hint:</u> Assume that the initial value of the latest staircase approximation $m_q(nT_s - T_s)$ is zero. In addition, encoder output will be 1 if the encoder input is greater than zero, otherwise it will be 0.

d. Plot the message signal $m(t)$ and its staircase approximation $m_q(t)$ on the same axis.
<u>Hint:</u> Use the function **stairs(.)** to plot $m_q(t)$. Pay attention to the dimension of $m_q(t)$, and make the necessary regulation to plot it.
<u>Remark:</u> **Do not forget to insert title, legend, and axis labels!**

e. Compute the average granular noise power by

$$\text{P} = \frac{\Delta^2 f_m}{3F_s}. \tag{2}$$

f. Compute the mean squared error (MSE) between the message signal and its staircase approximation by

$$\text{MSE} = \text{E}\left\{(m(t) - m_q(t))^2\right\} \tag{3}$$

g. Perform the steps **c**, **d**, **e**, and **f** for three different step size values $\Delta_1 = 0.2$, $\Delta_2 = 0.4$, and $\Delta_3 = 1.4$. As a result of this step, you should obtain three different figures including $m(t)$ and $m_q(t)$ plots for Δ_1, Δ_2, Δ_3, and the MSE and average granular noise power values for each step size.

h. Plot the following figures

1) MSE values $\mathbf{MSE} = [\mathrm{MSE}_1, \mathrm{MSE}_2, \mathrm{MSE}_{min}, \mathrm{MSE}_3]$ with respect to the corresponding step sizes $\mathbf{\Delta} = [\Delta_1, \Delta_2, \Delta_{min}, \Delta_3]$,
2) Average granular noise powers $\mathbf{P} = [\mathrm{P}_1, \mathrm{P}_2, \mathrm{P}_{min}, \mathrm{P}_3]$ with respect to the corresponding step sizes $\mathbf{\Delta} = [\Delta_1, \Delta_2, \Delta_{min}, \Delta_3]$

in a 2×1 subplot.

Answer the following questions in your report.

Q1. What is the condition to avoid the slope overload distortion for a sinusoidal wave? Briefly explain what will occur if the condition is not satisfied.

Q2. For which delta values, slope overload distortion occurs, and for which, granular noise occurs? Briefly explain your reasoning.

Q3. According to the figure you have obtained in step **h**, what is the relation between granular noise and slope overload distortion? Explain the reason behind the behavior of the MSE graph?

Q4. What can be further applied to avoid both granular noise and slope overload distortion?

Q5. How many bits are used to represent a sample value in delta modulation?

3 Labwork

3.1 Transmitter

a. Construct a signal to transmit the bit sequence $N = [1001100101]$.

b. Set the sampling frequency F_s, and the time duration of the signal as $1kHz$ and $1s$, respectively.

c. Generate symbol waveforms $s_0(t)$ and $s_1(t)$ given in Figure 1 that represent 0 and 1 bits for a binary digital communication system, respectively. T_b is the bit duration.

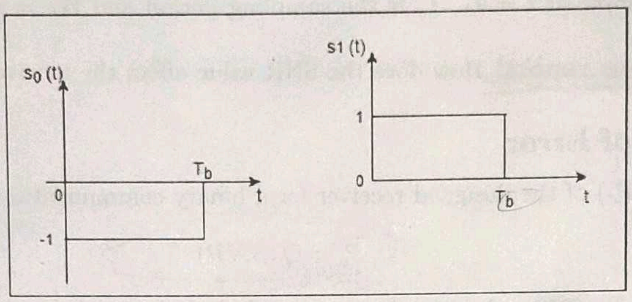

Figure 1: Symbol waveforms $s_0(t)$ and $s_1(t)$

d. Construct the transmitted signal $s(t)$ that includes $s_0(t)$ and $s_1(t)$ according to the bit sequence N.

3.2 Channel

a. $s(t)$ is transmitted over an additive white Gaussian noise (AWGN) channel. You can use MATLAB's built in function **awgn()**.

b. Pass the transmitted signal through an AWGN channel having a 15dB signal-to-noise ratio (SNR) to obtain the received signal, $r(t)$.

3.3 Receiver

a. Design an optimum receiver AWGN channel that finds the correlation between symbol waveforms ($s_0(t)$, $s_1(t)$) that generated the transmitted signal, $s(t)$, and the received signal, $r(t)$.

b. The block diagram of a correlator receiver is given in Figure 2.

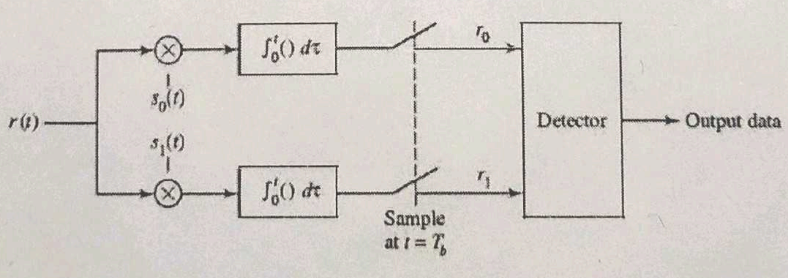

Figure 2: Block diagram of correlator receiver for a binary communication system

c. The equation for correlator receiver is given in 1 & 2.

$$r_0[k] = \sum_{n=(k-1)W_b+1}^{kW_b} r(nT_s)\, s_0\left((n-(k-1)W_b)T_s\right), \quad k = 1,2,....,N \tag{1}$$

$$r_1[k] = \sum_{n=(k-1)W_b+1}^{kW_b} r(nT_s)\, s_1\left((n-(k-1)W_b)T_s\right), \quad k = 1,2,....,N \tag{2}$$

where k is the sample at $t = T_b$, T_s is the sampling period and $W_b = \frac{T_b}{T_s}$ is the pulse width in these equations.
Comment in your report! How does the SNR value affect the received signal, $r(t)$?

3.4 Probability of Error

The probability error (P_e) of the designed receiver for a binary communcation system is given as 3.

$$P_e = Q(SNR) \tag{3}$$

a. Calculate the P_e for SNR values in the range of $[-15, 15]$ dB from simulation results by comparing transmitted signal ($s(t)$) and received signal ($r(t)$) iteratively.

b. Calculate the P_e for SNR values in the range of $[-15, 15]$ dB. (**Hint:** You can use the ***qfunc()*** built-in MATLAB function.)

Comment in your report! Compare the theoretical and simulation results of P_e?

3.5 Plotting

a. Repeat the sections 3.2 and 3.3 for SNR values 0 and -15 dB.

b. Plot $s(t)$ and $r(t)$ in the same figure by using subplot(21x) function. **Comment in your report!** How does the SNR value affect the received signal ?

c. Plot the correlation receiver outputs, $r_0(t)$ and $r_1(t)$, on the same axis by using **scatter()**. **Comment in your report!** How does the SNR value affect the correlator receiver outputs ?

d. Plot the P_e vs. SNR values of the theoretical and simulation results so that the vertical axis of the graph would be on a logarithmic scale. **Comment in your report!** How does the SNR value affect the P_e?
Please use title(),xlabel(), ylabel() and legend() function effectively to explain your figures!

bit rate — 1 = bit dr.

symbol ar.

Symbol dur ⟷ symbol rate.

EE 451: COMMUNICATION SYSTEMS II - LABORATORY
Laboratory 4: Inter-Symbol Interference (ISI)

Creating the message signal

bit duration = $\frac{1}{R_b}$

1. Set the sampling frequency $F_s = 27$kHz and the bit rate $R_b = 18$kbits/s.
2. Generate a random 4-ary PAM signal with 1000 symbols.
3. Calculate the symbol rate R_s according to the modulation technique. Also calculate the number of samples per symbol.

$R_s = ?$

Pulse shaping the data

4. Design a Square-Root Raised Cosine filter using **rcosdesign(.)** and specify a roll-off factor of 0, 0.5 and 1, respectively. Truncate the filter to 10 symbols.
5. **Create a 2x3 subplot** for the changing roll-off factors of 0, 0.5 and 1, respectively then **plot** the outputs of Square-Root Raised Cosine filters in the time domain and also **plot** the frequency spectrum of each.

Figure.

Hint: The upper part of subplot includes time domain plots, lower part includes the frequency spectrum of each.

Repeat all of the following steps for all previously specified roll-off factors.

The transmitted/received signal

Suppose that the channel is modeled as a linear filter having an equivalent lowpass frequency response $C(f)$ which is given as follows:

$$C(f) = \begin{cases} 1, & \text{if } |f| \leq W \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

6. Use **upfirdn(.)** function in order to upsample and filter the modulated signal for pulse shaping. So you obtained the transmitted signal.
7. Then, pass the transmitted signal through an AWGN channel having a 20dB signal-to-noise ratio (SNR). So, you obtained the received signal.
8. Filter and downsample the received signal. So, you obtained y_{out}.
9. Remove the first N and the last N elements from y_{out} to account for the filter delay.

   Hint: N is the number of symbols predefined above.

Eye diagrams

10. **Plot** the eyediagrams for the filter outputs for the period $1/F_s$ with zero offset.

    Hint: Use **eyediagram(.)**

⚠ **The figures must be shown in the defined format to get full credit. Do not forget to add labels and legends in order to have clear figures.**

Answer the following questions in your report.

Q1. What are the major approaches to deal with ISI ?

Q2. What is the effect of small and large roll-off factor considering the figure which is obtained in Step 5?

Q3. How can we observe the effect of the excess bandwidth in the eye-diagram?

Q4. Determine and calculate the channel bandwidth to achieve a zero-ISI condition considering R_s?

$T_b = \frac{1}{2R_s}$ $\quad W = \frac{1}{2T_b} = R_s$

$B_T = W(1+\alpha) = R_s(1+\alpha)$

5 Labwork

5.1 BASK/BFSK/BPSK Modulation Techniques

a. Generate a random message bit stream of length 10 and modulate the message bits using **BASK** where $F_s = 45$kHz and $f_c = 3$kHz. Note that each bit duration is 2 ms.

Hint1: *Use randi() ~~and repmat()~~*

*Ex: b is a **column** vector which contains randomly generated 10 bits → use randi()*

BitSampleSize is the size of one bit → calculate

b2 is a vector contains the bit sequence

```
b2 = repmat(b, 1, BitSampleSize)';
b2 = b2(:)';
```

Fig 1.

b. Create a code to demodulate the **BASK** modulated signal and compare the bit stream and output of the demodulator. Plot the bitstream, modulated and demodulated signals together using 3x1 subplot.

Hint2: *For demodulation use two path correlation receiver ~~given in Figure 1. x(t) is the modulated signal,~~ $s_0(t)$ and $s_1(t)$ are the same signals used to modulate the bitstream.*

Hint3: *To correlate the two signals x and y, simply use the command c_{xy} = sum(x.*y)*

Fig 2

c. Using the same bit stream repeat steps (a) and (b) for **BFSK**. Use the carrier frequencies $f_1 = 3$kHz and $f_2 = 1.5$kHz.

Fig 3.

d. Using the same bit stream repeat steps (a) and (b) for **BPSK**. Use the carrier frequency $f_c = 3$kHz.

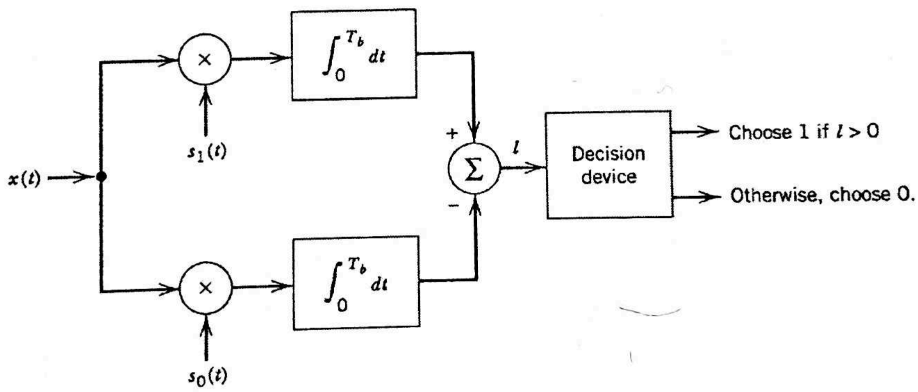

Figure 1: Two path correlation receiver for the general case

Answer the following questions in your report.

a. Explain BASK, BFSK and BPSK briefly by giving their modulator block diagrams (including $s_0(t)$ and $s_1(t)$) and write down the mathematical expression of these modulation techniques.

b. Explain BASK, BFSK and BPSK briefly by giving their demodulator block diagrams (including $s_0(t)$ and $s_1(t)$) and write down the mathematical expression of demodulation technique.

c. Compare BASK, BFSK and BPSK by considering their modulated signals.

Note: MATLAB's built-in functions for modulation and demodulation cannot be used.