1. **Frequency division multiplexing**

   (Taken from OWN, Prob. 8.33) Let us consider the frequency-division multiplexing of discrete-time signals $x_i[n], i = 0, 1, 2, 3$. Furthermore, each $x_i[n]$ potentially occupies the entire frequency band $(-\pi < \Omega < \pi)$. The sinusoidal modulation of upsampled versions of each of these signals may be carried out by using either double-sideband techniques or single-sideband techniques.

   (a) Suppose each signal $x_i[n]$ is upsampled by a factor of $L$ and then modulated with $\cos[i(\pi/4)n]$. What is the minimum $L$ required to ensure that the spectrum of the FDM signal does not have any aliasing? (10 pts)

   (b) If the upsampling of each $x_i[n]$ is restricted to be by a factor of 4, how would you use single-sideband techniques to ensure that the FDM signal does not have any aliasing? Include a sketch of the Fourier transform of the FDM signal, clearly delineating which part of the signal corresponds to which $x_i$. (That is, you might assume that $X_1(e^{j\Omega})$ is a triangle, $X_2(e^{j\Omega})$ a rect, $X_3(e^{j\Omega})$ a semicircle, etc, and you should have a key showing which shapes go with which input signal.) (10 pts)

   **Solution:**

   (a) The separation between the center frequencies of the cosines is $\pi/4$. Since the spectral occupancy of each $x_i[n]$ is $[-\pi/L, \pi/L]$, this requires that

   $$2\pi/L \geq \pi/4$$

   for no spectral overlap. This gives us that $L \geq 8$.

   (b) We can retain the upper sidebands. This way it may be verified that the modulated signals do not overlap, or in other words, the FDM signal does not have any aliasing.
2. **Quadrature Modulation and Complex Baseband Representations** (10 pts)

Consider a quadrature modulated signal \( s(t) = m_1(t) \cos(\omega_c t) + m_2(t) \sin(\omega_c t) \) from homework 3 for both analog communications (\( m_1(t) \) and \( m_2(t) \) analog signals) and digital communications (\( m_1(t) \) and \( m_2(t) \) baseband digital signals). The receiver for the quadrature system in both cases is as shown in the figure 1 and figure 2. Note that figure 1 has been updated to have the correct signal. Please use the signal as mentioned here. You may find the following trig identities useful:

\[
\cos(a) \cos(b) = \frac{1}{2} \left( \cos(a + b) + \cos(a - b) \right)
\]

\[
\cos(a) \sin(b) = \frac{1}{2} \left( \sin(a + b) - \sin(a - b) \right)
\]

Assume now that we have a digital system with the in-phase and quadrature signals consisting of a polar modulated signal. Suppose you send a “1” bit on the in-phase branch with amplitude \( A = 10 \) and a “0” bit on the quadrature branch with amplitude \( -A = -10 \). Find \( R_0, R_1, \) and \( a_0 \) in this case such that in the absence of noise, \( R_I(nT_b) = R_0 \) if a “0” is sent and \( R_I(nT_b) = R_1 \) if a “1” is sent on the in-phase branch, and similarly on the quadrature branch, with \( a_0 = \frac{(R_1 + R_0)}{2} \). Assume the noise introduced by the channel that is input to the decision device is \( N_I \) on the in-phase branch and \( N_Q \) on the quadrature branch. For \( \phi = \pi/8 \), find all values of the noise \( N_I \) and \( N_Q \) that will cause an error in detecting these bits.

**Solution:**

* For this part, answers that do not assume a phase difference between the transmitter and receiver will also be accepted.

If a “1” is sent: \( R_I(nT_b) = (10) \cos(\phi) = 10 \cos(\phi) \)

If a “0” is sent: \( R_I = (-10) \cos(\phi) = -10 \cos(\phi) \)

\( a_0 = 0 \)

In order to have a bit error, we must have either \( N_I < a_0 - 5 \cos(\phi) + \frac{1}{2}m_2(t) \sin(\frac{\phi}{2}) \) or \( N_I > a_0 + 5 \cos(\phi) + \frac{1}{2}m_2(t) \sin(\frac{\phi}{2}) \). For the quadrature branch, this range is: \( N_Q < a_0 + m_1(t) \cos(\frac{\phi}{2}) - 10 \sin(\frac{\phi}{2}) \) or \( N_Q > a_0 + m_1(t) \cos(\frac{\phi}{2}) + 10 \sin(\frac{\phi}{2}) \).
3. **Finite Impulse Response Filter**

We are given an ideal continuous-time highpass filter with frequency response below:

![Ideal Highpass Filter](image)

- **Solution:**

a) Assuming $M + 1$ is large, sketch the frequency response $H_a(e^{j\Omega})$ for $-\pi < \Omega < \pi$ (can draw real and imaginary parts or magnitude and phase). Does the Gibbs phenomenon occur? Why or why not?

b) Give an expression for $h_a[n]$. 

*Solution:*
a) Yes, the ideal highpass filter response $H(j\omega)$ has discontinuities and these discontinuities lead in to $H_a(e^{j\omega})$ so the Gibbs phenomenon does occur.

b) Because this is a finite impulse response filter, $h_a[n]$ is only defined as non-zero in the range $n \in [-M/2, M/2]$. We can write:

$$h_a[n] = \begin{cases} h_d[n] & \text{for } |n| \leq M/2 \\ 0 & \text{otherwise} \end{cases}$$

where we define $h_d[n]$ as the impulse response of an ideal highpass filter with $\Omega_c = \frac{3\pi}{4}$:

$$h_d[n] = \frac{\pi - \Omega_c}{\pi}(-1)^n \text{sinc} \left( \frac{\pi - \Omega_c}{\pi} \right) = \frac{1}{4}(-1)^n \text{sinc} \left( \frac{1}{4}n \right)$$

4. **Causal filter through time delay/linear phase shift**

Consider the discrete time filter specified by the following DTFT:

$$H(e^{j\Omega}) = \cos(M\Omega) + \sin(M\Omega) \cot \left( \frac{\Omega}{2} \right).$$

(a) Show that the filter is non causal. (5 pts)

*Hint: You may use the fact that the transform is real and symmetric.*

*Solution:* The inverse transform of the filter has to be real and symmetric. There are two possibilities:

- The impulse response is zero for all $n$ with $|n| > 0$,
- the impulse response is non zero for some $n$ with $|n| > 0$.

Since this is clearly not a delta function in the time domain, the first option cannot be true and the filter is necessarily non causal.

(b) Is there an integer $n_0$ such that $e^{jn_0\Omega}H(e^{j\Omega})$ is causal? Find an $n_0$ if it exists. (10 pts)
Hint: Compute the Fourier transform of a discrete time rect and simplify it using $e^{j\theta} = \cos(\theta) + j\sin(\theta)$. You may want to use the following identities: $\sum_{r=0}^{M} a^r = (1 - a^{M+1})/(1 - a)$, $\cos(2A) = 1 - \sin^2(A)$, $\sin(A + B) = \sin(A)\cos(B) + \cos(A)\sin(B)$.

Solution: We note that for a positive integer $M$,

$$
\sum_{n=-M}^{M} e^{-j\Omega n} = e^{jM\Omega} \sum_{n=0}^{2M} e^{-j\Omega n}
$$

$$
= e^{jM\Omega} \frac{1 - e^{-j(2M+1)\Omega}}{1 - e^{-j\Omega}}
$$

$$
= e^{jM\Omega} \frac{1 - \cos((2M + 1)\Omega) + j\sin((2M + 1)\Omega)}{1 - \cos(j\Omega) + j\sin(j\Omega)}
$$

$$
= e^{jM\Omega} \frac{2\sin^2((M + 0.5)\Omega) + 2j\sin((M + 0.5)\Omega)\cos((M + 0.5)\Omega)}{2\sin^2(0.5\Omega) + 2j\sin(0.5\Omega)\cos(0.5\Omega)}
$$

$$
= e^{jM\Omega} \frac{\sin((M + 0.5)\Omega)}{\sin(0.5\Omega)} e^{j(\pi - (M + 0.5)\Omega)} e^{(-\pi + 0.5\Omega)}
$$

$$
= \frac{\sin((M + 0.5)\Omega)}{\sin(0.5\Omega)} = H(e^{j\Omega}).
$$

The given DTFT corresponds exactly to the transform of a discrete time rect function. Thus a value of $n_0$ which will guarantee causality is $n_0 = -M$. 

5
5. **FIR Causal Differentiation** (35 pts)

Consider the CT system

\[ x(t) \longrightarrow H(j\omega) \rightarrow \frac{d}{dt}x(t) \]

We wish to design a DT FIR approximation using the frequency response matching technique for CT signals sampled at rate \( \omega_s = 2\pi/T_s \). Since we have \( H(j\omega) = j\omega \) and since we saw that sampling without aliasing leads to the substitution \( \Omega = \omega T_s \) for the DTFT of the sampled signal (over a single period \( |\Omega| \leq \pi \)), the desired frequency response of our DT differentiator is

\[ H_d(e^{j\Omega}) = j\Omega/T_s, \quad |\Omega| \leq \pi. \]

In this question we explore the performance the FIR approximation to \( H_d(e^{j\Omega}) \).

a) Sketch \( |H_d(e^{j\Omega})| \) and \( \angle H_d(e^{j\Omega}) \) and give an expression for the impulse response \( h_d[n] \). Simplify this expression as much as you can. (8 pts)

**Solution:**

\[
   h_d[n] = \frac{1}{2\pi T_s} \int_{-\pi}^{\pi} j\Omega e^{j\Omega n} d\Omega
\]

\[
   = \begin{cases} 
   0 & n = 0 \\
   \frac{j}{2\pi T_s} \left[ \frac{e^{j\Omega n} (j\Omega n - 1)}{(jn)^2} \right] & \Omega = \pi, \quad n \neq 0 \\
   0 & n = 0 \\
   \frac{j}{2\pi T_s} \left[ 2j(n\pi - n\pi \cos(n\pi)) \right] & n \neq 0 \\
   0 & n = 0 \\
   \frac{(-1)^n}{nT_s} & n \neq 0.
   \end{cases}
\]

b) For an even \( M \), give an expression for the filter \( h_a[n] \), i.e., the non-causal FIR approximation of duration \( M + 1 \) that minimizes the frequency-domain (or time-domain) error. Compute the DTFT \( H_a(e^{j\Omega}) \) and determine whether \( H_a(e^{j\Omega}) \) is real, imaginary or complex? (8 pts)

**Solution:**

\[
   h_a[n] = \begin{cases} 
   h_d[n], & |n| \leq M/2 \\
   0, & |n| > M/2.
   \end{cases}
\]

\[
   H_a(e^{j\Omega}) = T_s^{-1} \sum_{n=-M/2}^{M/2} \frac{(-1)^n}{n} e^{-jn\Omega} + T_s^{-1} \sum_{n=1}^{M/2} \frac{(-1)^n}{n} e^{-jn\Omega}
\]

\[
   = T_s^{-1} \sum_{n=1}^{M/2} \frac{(-1)^n}{n} e^{jn\Omega} + T_s^{-1} \sum_{n=1}^{M/2} \frac{(-1)^n}{n} e^{-jn\Omega}
\]

\[
   = -\frac{2j}{T_s} \sum_{n=1}^{M/2} \frac{(-1)^n}{n} \sin(n\Omega)
\]
c) Find the impulse response $h_{cu}[n]$ and the frequency response $H_{cu}(e^{j\Omega})$ of a causal FIR filter such that $|H_{cu}(e^{j\Omega})| = |H_0(e^{j\Omega})|$. What is the group delay of $H_{cu}(e^{j\Omega})$? (8 pts)

Solution:

$$h_{cu}[n] = h_0[n - M/2] = \begin{cases} (-1)^{n-M/2} & n = 0, 1, \ldots, M/2 - 1, M/2 + 1, \ldots, M \\ 0 & n = M/2, n < 0, n > M. \end{cases}$$

A causal DTFT is obtained by multiplying $H_0(e^{j\Omega})$ by $e^{-j\Omega M/2}$, leading to

$$H_{cu}(e^{j\Omega}) = - \frac{2j}{T_s} e^{-j\Omega M/2} \sum_{n=1}^{M/2} (-1)^n \sin(\Omega n) = - \frac{2j}{T_s} e^{-j\Omega M/2 - j\pi/2} \sum_{n=1}^{M/2} (-1)^n \frac{\sin(\Omega n)}{n}. $$

The group delay is given by

$$- \frac{d}{d\Omega} \angle H_{cu}(e^{j\Omega}) = - \frac{d}{d\Omega} (-\Omega M/2 - \pi/2) = \frac{M}{2}. $$

d) We are given a continuous-time signal $x(t) = \sin \left( \frac{2\pi}{8Ts} t \right)$. Denote by $y(t)$ the CT process obtained by sampling $x(t)$ at rate $\omega_s = \frac{2\pi}{T_s}$, passing the resulting signal through the filter $H_{cu}(e^{j\Omega})$ and use optimal D/A reconstruction. Find an expression for the signal $y(t)$. Use MATLAB to evaluate the expression you obtained for $y(t)$ for $M = 8, 16$ and $32$ with $T_s = 1$ and compare it with the output of an ideal CT differentiator. Plot $|y(t)|$ and $\frac{d}{dt}x(t)$ on the same axis over the time range $0 \leq t \leq 10$. (11 pts)

Solution:

We have $x_d[n] = \sin(\pi n/4)$. The result of passing $x_d[n]$ through $H_{cu}(e^{j\Omega})$ leads to

$$Y_d(e^{j\Omega}) = 2\pi H_{cu}(e^{j\pi/4}) \delta(\Omega - \frac{\pi}{2}) - H_{cu}(e^{-j\pi/4}) \delta(\Omega + \frac{\pi}{4}), \quad |\Omega| \leq \pi. $$

Using an ideal D/A we obtain

$$y[t] = \frac{H_{cu}(e^{j\pi/4}) e^{-j\frac{\pi}{4} t} - H_{cu}(e^{-j\pi/4}) e^{j\frac{\pi}{4} t}}{2j} $$

$$= -e^{-j\pi M/8 \sum_{n=1}^{M/2} (-1)^n \sin(n/4) e^{-j\frac{\pi}{4} t}} + e^{j\pi M/8 \sum_{n=1}^{M/2} (-1)^n \sin(n/4) e^{j\frac{\pi}{4} t}} $$

An ideal differentiator yields

$$\frac{d}{dt} \sin \left( \frac{2\pi}{8Ts} t \right) = \frac{2\pi}{8Ts} \cos \left( \frac{2\pi}{8Ts} t \right) $$

$$= \frac{\pi}{8Ts} e^{-j\frac{\pi}{4} t} + \frac{\pi}{8Ts} e^{j\frac{\pi}{4} t} $$

Comparing the latter to (1), we see that a good approximation to the differentiator implies

$$- \frac{\pi}{8} \approx e^{j\pi M/8 \sum_{n=1}^{M/2} (-1)^n \sin(n/4) \frac{n}{n}}. $$

You are invited to proof that this is indeed the case as $M$ goes to infinity.
Consider the FSK demodulator shown below, where \( s(t) = A_c \cos(\omega_1 t) \) for \( \omega_1 = \omega_1 \) ("1" sent) or \( \omega_2 = \omega_2 \) ("0" sent). The decision device outputs a “1” if \( x(nT_b) > T \) and a “0” if \( x(nT_b) \leq T \) where \( T \) is a threshold level. Given the bit rate, \( T_b \), you can assume that \( 1/T_b \ll \omega_1, \omega_2 \).

![FSK Demodulation Diagram](image)

**Figure 4: FSK Demodulation**

a) Assume that \( \omega_1 - \omega_2 = \Delta \omega = 2\pi/T_b \). Show that if \( s(t) = A_c \cos(\omega_1 t) \) and we neglect very small terms, \( r_1(nT_b) = .5A_c T_b \) and \( r_2(nT_b) = 0 \). Show that if \( s(t) = A_c \cos(\omega_2 t) \) and we neglect very small terms, \( r_1(nT_b) = 0 \) and \( r_2(nT_b) = .5A_c T_b \). Explain why \( T = 0 \) is the correct threshold.

b) What is the smallest \( \Delta \omega \) for which the answer from part (a) holds and why? What is \(|r_1(nT_b) - r_2(nT_b)|\) when \( \Delta \omega = 0 \)?

**Solution:**

a) Start with the case where \( s(t) = A_c \cos(\omega_1 t) \) and we want to find \( r_1(nT_b) \). To do this, we take \( s(t) \cos(\omega_1 t) = \frac{1}{2}A_c(\cos(0) + \cos(2\omega_1 t)) \) and integrate from \( 0 \) to \( T_b \). Since \( \omega_1 \gg \frac{1}{T_b} \), the value of \( \cos(2\omega_1 t) \) over time \( T_b \) can be approximated as 0 and ignored. We are then left with \( \int_0^{T_b} \frac{1}{2}A_c(\cos(0)) = \frac{1}{2}A_c T_b \). In the case where we look at the same channel to find \( r_1(nT_b) \), but \( s(t) = A_c \cos(\omega_2 t) \), we take \( s(t) \cos(\omega_2 t) = \frac{1}{2}A_c(\cos(\omega_2 - \omega_1) + \cos(\omega_2 + \omega_1)) \). We can again see that on the time scale of \( 0 \) to \( T_b \), \( \cos(\omega_2 \pm \omega_1 t) \) is negligible. The question then becomes what the value of \( \int_0^{T_b} \frac{1}{2}A_c(\cos(\omega_2 - \omega_1)) \). To solve for this, we notice that since the difference in the frequencies is \( \frac{2\pi}{T_b} \), this signal exactly finishes a period within the time \( 0 \) to \( T_b \) and thus the integral is 0. We can use the exact same process to show the values for \( r_2(nT_b) \). From the symmetry of these values around 0, it is clear why \( T = 0 \) is the correct threshold.

b) In part a, we saw that when the signal corresponding to \( \cos(\Delta \omega) \) finished one full period in the time from \( 0 \) to \( T_b \), then the integral was 0. The same can be said if the signal finishes exactly one half period, which corresponds to \( \Delta \omega = \frac{\pi}{T_b} \). When \( \Delta \omega = 0 \), there will be no difference in the received signals based on what was sent, and \(|r_1(nT_b) - r_2(nT_b)| = 0 \).
MATLAB Assignment

General Instructions
Answer all questions asked. Your submission should include all m-file listings and plots requested. All plots should have a title and x- and y-axes properly labeled.

1 Phase Shift Keying (PSK)

(25 pts)

In this problem you will explore the technique of Phase Shift Keying (PSK). You will modulate a signal to encode a bit stream and then demodulate the same signal to recover the original information. To start, create and store a bit stream with random binary values. To do this, use the command \( \text{rand}(1,T)>0.5 \) with \( T=10 \) bits. Assume that \( T_b = 1 \) second and \( w_1 = 2\pi*5 \) radians/second. When creating sinusoids, use a sampling period of 1/100 second. This will lead to a total of \( N=1000 \) samples in total.

a) Phase Shift Keying: On the interval \( i=1:T \), perform PSK to send the bit stream stored in \( b \) by sending (for bit “1”) either a sinusoid with angular frequency \( w_1 \) and amplitude \( \sqrt{0.5} \) or this same sinusoid offset by a phase difference of \( \pi \) (for bit “0”). Store this modulated signal as \( \text{psk} \). Display \( \text{psk} \).

b) Create a function to demodulate the signal stored in \( \text{psk} \) and return the vector of binary values. This may be done by multiplying the signal stored in \( \text{psk} \) by an appropriate sinusoid, integrating over bit time intervals (can be done using \( \text{sum} \)), and using a thresholder on the integrator output every bit time, as described in Problem 2.

c) Extra credit: Create a function which evaluates the bit error rate (BER) as a function of the noise power \( a \). To do this, create a vector \( n = \sqrt{a} \cdot \text{randn}(0,N) \) with \( N=1000 \) samples where each sample is drawn from a normal (Gaussian) distribution. Add this noise vector to \( \text{psk} \) and demodulate the signal. Create the random noise vector 100 times (since the random vector will be different each time) and repeat the process. Calculate the total number of errors in these 10000 trials and divide it by the total number of bits (there are 100000 bits in total) to get the BER. Run this function for \( a=.2, 10, 100 \) and plot \( \log_{10}(\text{BER}) \) vs. the inverse of the noise power in dB, given by \( -10 \cdot \log_{10}(a) \).

2 Discrete-Time Differentiation

(40 pts)

In this problem we consider a DT differentiator as defined in problem 5 with frequency response:

\[
H_d(e^{j\Omega}) = j\Omega / T_i, \quad |\Omega| \leq \pi.
\]

Throughout, assume that \( T_i \) is equal to 1. We will refer to the answer for problem 5(a) as \( h_a[n] \). You should note that \( H_d(e^{j\Omega}) \) is purely imaginary and \( h_a[n] \) is antisymmetric about \( n = 0 \). If \( H_d(e^{j\Omega}) \) is multiplied by a linear phase term, \( H_d(e^{j\Omega}) e^{-j\alpha\Omega} \), the resulting impulse response will be antisymmetric about \( n = \alpha \), assuming \( 2\alpha \) is an integer. If \( \alpha \) is an integer, the impulse response will be \( h_a[n-\alpha] \).
a) Generate and plot $h[n]$ for $-256 < n < 256$ (i.e., use $M = 512$).

b) A rectangular window of length $N$ is defined to be equal to one for $0 \leq n < N$, and equal to zero elsewhere. Construct the impulse response $h[n] = w[n]h_a[n-n_d]$ where $w[n]$ is a rectangular window of length $N = 511$, created by $w = \text{boxcar}(511)$, and $n_d$ is an integer delay. Choose $n_d$ so that $h[n]$ is causal and store your result in the vector $h$. Use $H = \text{fft}(h,1024)$ to generate 1024 equally spaced samples of $H(e^{j\Omega})$. Plot the magnitude of $H$ versus $\Omega$ for $-\pi \leq \Omega < \pi$. How do you explain the behavior of the magnitude in the vicinity of $\Omega = \pm \pi$?

c) Generate and plot the magnitude of $H(e^{j\Omega})$ when $w[n]$ is a rectangular window of length $N = 17$. How does this frequency response compare to part (b)?

d) Generate samples on the interval $0 \leq t \leq 100$ of the signal $x_c(t) = \cos(\Omega_c(t))$ for $\Omega_c = \pi/10$. Store the result in $x_d$ and plot $x_d[n]$.

e) Take the derivative of the signal $x_c(t)$ by using the DT differentiator from part (c) using $y_d = \text{conv}(h,x_d)$. Since you are implementing the differentiator with a causal filter, and the ideal differentiator is non-causal, your resulting $y_d$ will be delayed by $n_d$ samples, where $n_d = 8$ for $N = 17$. Determine analytically the derivative of $x_c(t-n_d)$ and store samples of the derivative in the vector $y_{true}$. Plot $x_d$, $y_d$, and $y_{true}$ on the same set of axes for $20 \leq t \leq 100$. Does the DT differentiator give you roughly the same answer as the analytically determined derivative?

f) Repeat part (b)/(c) with a new window $w[n]$ defined as $w = \text{hamming}(17)$ (hamming filter) and repeat part (e) with your new impulse response.

g) Repeat parts (d)-(f) for $\Omega_c = \pi/4$, $3\pi/4$, $5\pi/4$. For which frequencies does the DT differentiator perform the proper differentiation?

3 Extra Credit: Intersymbol Interference in BPSK Systems

In a Binary Phase Shift Keying (BPSK) system, a discrete-time message is sent over a communication channel by modulating a periodically repeated pulse shape. Depending on the bandwidth limitations of the channel and the time characteristics of the pulse shape, each of the message samples transmitted by modulating the pulse shape may be received without interference between the adjacent message samples. However, if the bandwidth limitations of the channel result in temporal spreading of the transmitted pulses, then significant intersymbol interference (ISI) can occur. This exercise explores some of the issues within a BPSK system that give rise to ISI.

For this exercise, consider a simple binary communication system in which a “1” is represented by transmitting the pulse $p(t)$ and a “0” is represented by transmitting the negative of the pulse, $-p(t)$. This is POLAR signaling. Pulse $p(t)$ is given by:

$$p(t) = \begin{cases} 
1 - |t|/T & \text{if } |t| < T, \\
0 & \text{otherwise}.
\end{cases}$$

The pulses are transmitted at a rate of $f_b$ pulses per second, referred to as the bit rate of the BPSK system. For the remainder of the exercise, assume that $T = 0.1$. 

11
a) Generate and plot the pulse \( p(t) \) at time samples \([-t:dt:T]\) for \( dt = T/10 \) and store the result in vector \( p \).

b) Determine the maximum value of \( f_b \) so that there is no ISI in the transmitted waveform. That is, what is the maximum value of \( f_b \) such that \( N \) pulses in the signal \( y(t) = \sum_{k=1}^{N} p(t - k/f_b) \) do not overlap? What is the maximum value of \( f_b \) such that samples taken at the bit rate and located at the pulse peaks are not affected by intersymbol interference, i.e. \( y(n/f_b) = \sum_{k=1}^{N} p((n - k)/f_b) = p(0) \).
Homework 4

Table of Contents

Problem 1 ................................................................. 1
Problem 2 ................................................................. 4
Problem 3 ................................................................. 16

Problem 1

close all

% Initialization

% Initialize the bit stream
T = 10;
b = rand(1,T) > 0.5;

w = 2*pi*5; %Fundamental Frequency
T = .01:.01:T; % Time axis
n = 100; % Number of time samples per bit
N = T*n; %Total number of samples

% Part a, PSK
psk = zeros(1,N);
for i = 1:T
  if (b(i) == 1)
    psk((i-1)*n+1:i*n) = sqrt(.5)*cos(w*t((i-1)*n+1:i*n));
  else
    psk((i-1)*n+1:i*n) = sqrt(.5)*cos(w*t((i-1)*n+1:i*n)+pi);
  end
end

plot(1:N, psk)
title('Phase Shift Keying')

%part b  PSK Demodulation
b_psk = zeros(1,T);

psk_mod = psk.*cos(w*t); %Multiply the signal by a sinusoid

for i = 1:T
  r = 2*sum(psk_mod((i-1)*n+1:i*n));
  if (r > 0)
    b_psk(i) = 1;
  end
end
%compare b and b_psk
display(b)
display(b_psk)

%part c Extra Credit

%Additive Gaussian Noise for PSK

b_n_psk = zeros(1,T);
a = [0.2 10 100];
ber_psk = zeros(1,length(a));

for m = 1:length(a)
p = sqrt(a(m));
ber_total = 0;
for l = 1:10000
b_n_psk = zeros(1,T);
noise = p*randn(1,N);
psk_n = psk + noise;
psk_n_mod = psk_n.*cos(w*t);
for k = 1:T
r = 2*sum(psk_n_mod((k-1)*n+1:k*n));
if (r > 0)
b_n_psk(k) = 1;
end
end
ber_total = ber_total + sum(abs(b - b_n_psk));
end
ber_psk(m) = ber_total*(1/100000);
end
figure
plot(-10*log10(a), log10(ber_psk), 'g');
title('Bit Error Rate for PSK, (Accepted even with 100 trials)

b =

1x10 logical array

1 1 0 0 0 1 1 0 0 1

b_psk =

1 1 0 0 0 1 1 0 0 1
Problem 2

Part a

```matlab
ha = zeros(511,1);
for n = -255:1:255
    if (n == 0)
        ha(n+256) = 0;
    else
        ha(n+256) = (1/n)*(-1)^n;
    end
end
plot(-255:1:255, ha)
title('Impulse response h_a')
```

% Part b

```matlab
w = boxcar(511);
h = w.*ha;%this is technically already causal since ha was generated with a shift
H = fftshift(fft(h,1024));
figure
plot(-pi:pi/512:pi-(pi/512), abs(H))
title('Frequency Response H')
```

% Part c

```matlab
w = boxcar(17);
h = zeros(511, 1);
for n = 1:17
    h(n) = w(n).*ha(n+(255-8));
end
h1 = h; %boxcar impulse
H = fftshift(fft(h, 1024));
figure
plot(-pi:pi/512:pi-(pi/512), abs(H))
title('Frequency Response H with window length 17')
```

% Part d

```matlab
Omega_c = pi/10;
for i = 1:101
    xd(i) = cos(Omega_c*(i-1));
end
figure
stem(0:100, xd)
title('Continuous Time Cosine')
```

% Part e

```matlab
yd = conv(h,xd);
for i = 1:101
    ytrue(i) = -Omega_c*sin(Omega_c*(i-1));
end
figure
hold on;
```
stem(20:100, xd(20:100), 'b');
stem(20:100, yd(20:100), 'g');
%shifting ytrue by 8 here while plotting
stem(20:100, ytrue(20-8:100-8), 'r');
legend('xd', 'yd', 'ytrue')
title('Derivative signal for boxcar(17)')

% Part f
w = hamming(17);
h = zeros(511, 1);
for n = 1:17
    h(n) = w(n).*ha(n+(255-8));
end
H = fftshift(fft(h,1024));
figure
plot(-pi:pi/512:pi-(pi/512), abs(H))
h2 = h;%hamming impulse
yd = conv(h, xd);
for i = 1:101
    ytrue(i) = -Omega_c*sin(Omega_c*(i-1));
end
figure
hold on;
stem(20:100, xd(20:100), 'b');
stem(20:100, yd(20:100), 'g');
stem(20:100, ytrue(20-8:100-8), 'r');
legend('xd', 'yd', 'ytrue')
title('Derivative signal for hamming(17)')

% Part g
%pi/4
Omega_c = pi/4;
for i = 1:101
    xd(i) = cos(Omega_c*(i-1));
end
figure
stem(0:100, xd)
title('Continuous Time Cosine')
yd = conv(h1, xd);
for i = 1:101
    ytrue(i) = -Omega_c*sin(Omega_c*(i-1));
end
figure
hold on;
stem(20:100, xd(20:100), 'b');
stem(20:100, yd(20:100), 'g');
%shifting ytrue by 8 here while plotting
stem(20:100, ytrue(20-8:100-8), 'r');
legend('xd', 'yd', 'ytrue')
title('Derivative signal for boxcar(17) for pi/4')
yd = conv(h2, xd);
for i = 1:101
    ytrue(i) = -Omega_c*sin(Omega_c*(i-1));
end
figure
hold on;
stem(20:100,xd(20:100), 'b');
stem(20:100,yd(20:100), 'g');
stem(20:100,ytrue(20-8:100-8), 'r');
legend('xd', 'yd', 'ytrue')
title('Derivative signal for hamming(17) for pi/4')

%3pi/4
Omega_c = 3*pi/4;
for i = 1:101
    xd(i) = cos(Omega_c*(i-1));
end
figure
stem(0:100, xd)
title('Continuous Time Cosine')
yd = conv(h1,xd);
for i = 1:101
    ytrue(i) = -Omega_c*sin(Omega_c*(i-1));
end
figure
hold on;
stem(20:100,xd(20:100), 'b');
stem(20:100,yd(20:100), 'g');
stem(20:100,ytrue(20-8:100-8), 'r');
legend('xd', 'yd', 'ytrue')
title('Derivative signal for boxcar(17) for 3*pi/4')
yd = conv(h2,xd);
for i = 1:101
    ytrue(i) = -Omega_c*sin(Omega_c*(i-1));
end
figure
hold on;
stem(20:100,xd(20:100), 'b');
stem(20:100,yd(20:100), 'g');
stem(20:100,ytrue(20-8:100-8), 'r');
legend('xd', 'yd', 'ytrue')
title('Derivative signal for hamming(17) for 3*pi/4')

%5pi/4
Omega_c = 5*pi/4;
for i = 1:101
    xd(i) = cos(Omega_c*(i-1));
end
figure
stem(0:100, xd)
title('Continuous Time Cosine')
yd = conv(h1,xd);
for  i = 1:101
    ytrue(i) = -Omega_c*sin(Omega_c*(i-1));
end
figure
hold on;
stem(20:100,xd(20:100), 'b');
stem(20:100,yd(20:100), 'g');
%shifting ytrue by 8 here while plotting
stem(20:100,ytrue(20-8:100-8), 'r');
legend('xd', 'yd', 'ytrue')
title('Derivative signal for boxcar(17) for 5*pi/4')

yd = conv(h2,xd);
for  i = 1:101
    ytrue(i) = -Omega_c*sin(Omega_c*(i-1));
end
figure
hold on;
stem(20:100,xd(20:100), 'b');
stem(20:100,yd(20:100), 'g');
stem(20:100,ytrue(20-8:100-8), 'r');
legend('xd', 'yd', 'ytrue')
title('Derivative signal for hamming(17) for 5*pi/4')
Problem 3

%Extra Credit

close all
T = 0.1;
t = -T: T/10:T;
p = 1- abs(t)/T;
figure
plot(t, p);
title('p(t)')
ylabel('t')

display('The maximum value of fb so that there is no ISI in the
 transmitted waveform is 1/2T. This follows from the relationship that
 k/fb + T <= (k+1)/fb -T')
display('The maximum value of fb such that samples taken at the bit
 rate and located at the pulse peaks are not affected by intersymbol
 interference is 1/T. This follows from the relationship that k/fb + T
 <= (k+1)/fb')

The maximum value of fb so that there is no ISI in the transmitted
 waveform is 1/2T. This follows from the relationship that k/fb + T
 <= (k+1)/fb -T
The maximum value of fb such that samples taken at the bit rate
 and located at the pulse peaks are not affected by intersymbol
 interference is 1/T. This follows from the relationship that k/fb + T
 <= (k+1)/fb
Published with MATLAB® R2018a