WES_268A_preLab 4

Theory Problems

1. Consider a receiver with noise power -160 dBm (within the signal bandwidth of interest). Assume a simplified path-loss model with $d_0 = 1$ m, and K obtained from the free-space model path loss formula(given below with omnidirectional antennas) and $f_c = 1$ GHz, and $\gamma = 4$. For a transmitter power of $P_t = 10$ mW, **find the maximum distance** between the transmitter and receiver **such that the received signal-to-noise power ratio is 20dB**. The received power for the simplified path-loss model is given by the following:

$$P_r = P_t K \left[\frac{d_0}{d}\right]^{\gamma}$$

$$P_r \text{ dBm} = P_t \text{ dBm} + K \text{ dB} - 10\gamma \log_{10}\left[\frac{d}{d_0}\right]$$

where P_r is the received power, P_t is the transmit power, d is the distance in meters, d_0 is a reference distance for the antenna far field, and γ is the path-loss exponent with K given below:

$$K \models \left(\frac{\lambda}{4\pi d_0}\right)^2$$

where λ is the wavelength of the carrier signal given by

$$\lambda = \frac{c}{f_c}$$

and c is the speed of light $c \approx 3.0 \times 10^8$ m/s. Values for γ depend on the environment and are given in the table below.

Environment	γ range
Urban macrocells	3.7-6.5
Urban microcells (up to 35km)	2.7-3.5
Office building (same floor)	1.6-3.5
Office building (multiple floors)	2-6
Store	1.8-2.2
Factory	1.6-3.3
Home	3

Table 1: Typical path-loss exponents

$\lambda = (3 * 10^8\ m/s)$/(1*10^9/s) = 0.3m

K = (0.3m/4πd_0) ^{2}

For 20dB SNR Pr= -140dBm = 10^-14mW

-10^-14mW = 10mW*(0.3m/4π $* 1^{\ 2}$) ^{2} $\left[\frac{1m}{d(m)}\right]^{4}$

d ^{4} = (10*/-10^-14) *(0.3m/4π) ^{2}

= 10^15*(0.3m/4π) ^{2}

d = 868.87m

2. *Phase noise*

A circularly gaussian probability density function $f(x, y)$ can be written as

$$f(x,y) \quad = \quad \left[\frac{1}{\sqrt{2\pi}\sigma}e^{-(x-\langle \underline{x}\rangle)^2/2\sigma^2}\right]\left[\frac{1}{\sqrt{2\pi}\sigma}e^{-(y-\langle \underline{y}\rangle)^2/2\sigma^2}\right], \qquad (1)$$

where $\langle x\rangle 2$ and $\langle y\rangle 2$ are the squares of the expected values of the independent gaussian random variables x and y that have the same variance σ^2. Let $A = \sqrt{\langle \underline{x}\rangle^2 + \langle \underline{y}\rangle^2}$.

Using the following change of variables, $x = A\cos\theta$, $y = A\sin\theta$, $dxdy = dr\, rd\theta$, $r = \sqrt{x^2+y^2}$ $x = r\cos\varphi$, $y = r\sin\varphi$, and redefining φ with respect to the phase θ of the constant amplitude signal so that $\theta - \varphi$ is replaced by φ. Using these substitutions (1) becomes

$$f_{\underline{r\phi}}(r,\phi) \quad = \quad \frac{r}{2\pi\sigma^2}e^{-(r^2-2Ar\cos\phi+A^2)/2\sigma^2}$$

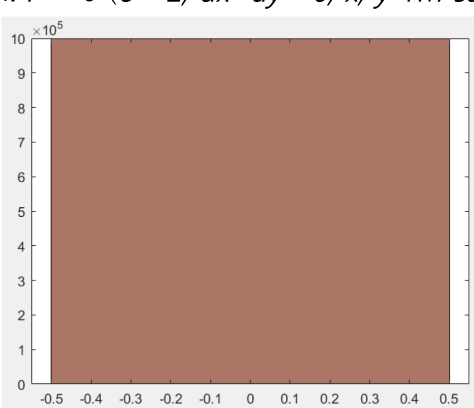
This is a circularly symmetric gaussian random variable in polar coordinates. The marginal phase distribution $f_\varphi(\varphi)$ is of joint distribution is (see Eq. 2.2.35 in the text)

$$f_{\underline{\phi}}(\phi) \quad = \quad \frac{1}{2\pi}\left(e^{-F} + \sqrt{\pi F}\cos\phi e^{-F\sin^2\phi}\left(1+\mathrm{erf}\left(\sqrt{F}\cos\phi\right)\right)\right)$$

where $F = A^2/2\sigma^2$, and erf is the error function.

a) Plot $f_\varphi(\varphi)$ for the following values of F:

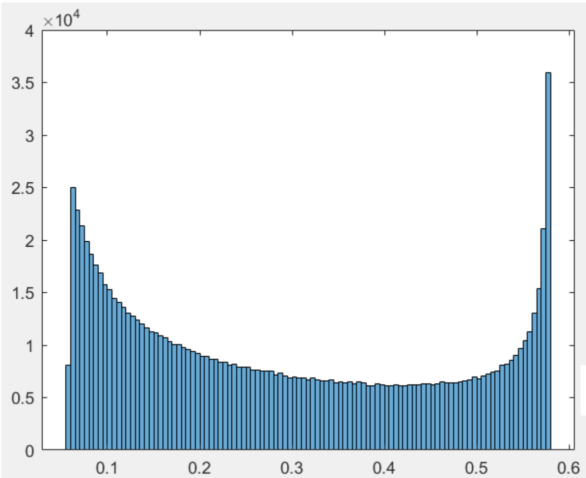
i. $F = 0$ (*σ =2, ux=uy= 0, x, y 1M samples each case*)



(radian)

```
sigma =2;
mux = sqrt(0)*sigma;
muy = sqrt(0)*sigma;
A = sqrt(mux^2+muy^2)
x = normrnd( mux, sigma, 1, 1000000);
y = normrnd( muy, sigma, 1, 1000000);
```

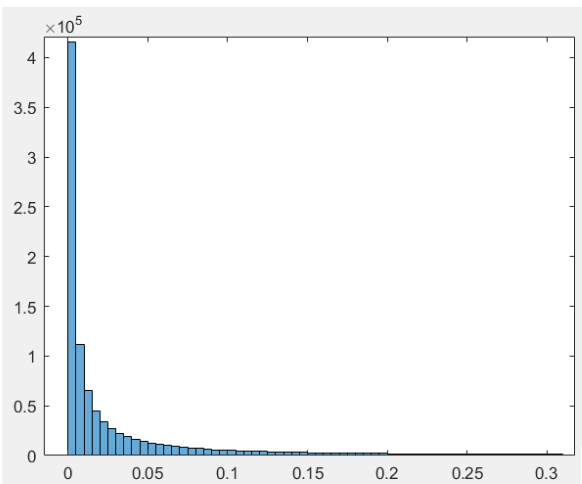
ii. $F = 1$ (ux=uy=σ=2, *x, y 1M samples each case*)




```
sigma =2;
mux = sqrt(1)*sigma;
muy = sqrt(1)*sigma;
A = sqrt(mux^2+muy^2)
x = normrnd( mux, sigma, 1, 1000000);
y = normrnd( muy, sigma, 1, 1000000);
```

(radian)

iii. $F = 10$(ux=uy=$\sqrt{10}\sigma$, *x, y 1M samples each case)*



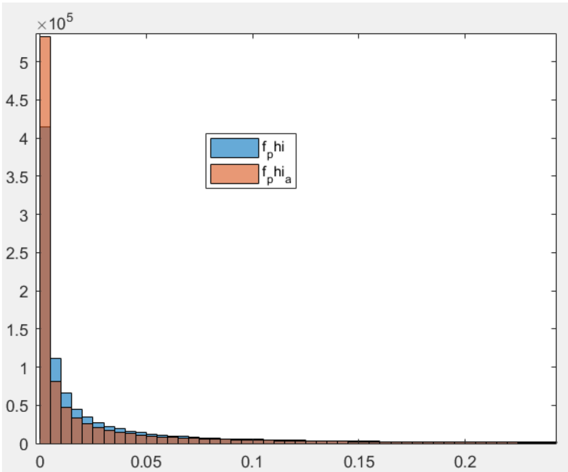

```
sigma =2;
mux = sqrt(10)*sigma;
muy = sqrt(10)*sigma;
A = sqrt(mux^2+muy^2)
x = normrnd( mux, sigma, 1, 1000000);
y = normrnd( muy, sigma, 1, 1000000);
```

(radian)

b) On the same curve as the plot for $F = 10$, plot the following approximate expression

$$f_{\underline{\phi}}(\phi) \quad \approx \quad \sqrt{\frac{F}{\pi}} e^{-F\phi^2},$$

$F = 10$(ux=uy=$\sqrt{10}\sigma$, *x, y 1M samples each case)*




```
sigma =2;
mux = sqrt(10)*sigma;
muy = sqrt(10)*sigma;
A = sqrt(mux^2+muy^2)
x = normrnd( mux, sigma, 1, 1000000);
y = normrnd( muy, sigma, 1, 1000000);
```

:approximation has higher at zero and lower around near zero phases. i.e., it`s more optimistic phase noise approximation.

3. *Fading*

Using the analytic expression for a Rayleigh fade (Slide 17 Lecture 4a)

$$\overline{P_e} = \frac{1}{2}\left(1 - \sqrt{\frac{z}{1+z}}\right)$$

determine the slope on log-log scale for large values of *z*.

a) Compare this slope to the slope to the unfaded case and comment. (See Figure on Slide 18 of Lecture 4a.) You may solve this problem numerically.

<Rayleigh-Faded case>

$$\log\bar{P}_e = \log(1/2)+\log\left(1-\sqrt{\frac{z}{(1+z)}}\right) = \log(1/2)+\log\left((1+z)^{\frac{1}{2}} - z^{\frac{1}{2}}\right) - \log(1+z)^{\frac{1}{2}}$$

$$\frac{d}{dz}\log\bar{P}_e = \frac{d}{dz}\left(\log(1/2)+\log\left((1+z)^{\frac{1}{2}} - z^{\frac{1}{2}}\right) - \log(1+z)^{\frac{1}{2}}\right)$$

$$= 0 + \frac{d}{dz}\log\left(z(1+z)^{\frac{1}{2}} - z^{\frac{1}{2}}\right) - ½\frac{d}{dz}\log(1+z)$$

$$\cong -½\frac{d}{dz}\log(z) = -½\frac{d}{dz}\ln(z)/\ln(10) = (1/4.6)*\ln(1/z)$$, with large z

< unfaded case>

$$\bar{P}_e = ½\,erfc\left(\left(\frac{Eb}{No}\right)^{½}\right)$$

$$= \int_{\left(\frac{Eb}{No}\right)^{1/2}}^{\infty}\left(\frac{1}{\sqrt{2\pi}}e^{-r'^2/2}\right)dr', \quad r' = (r + A)/\sigma$$

$$\log\bar{P}_e = \int_{A/\sigma}^{\infty}\left(\log\left(\frac{1}{\sqrt{2\pi}}\right) - \frac{r'^2}{2}\right)dr'$$

$$\frac{d}{dz}\log\bar{P}_e = \int_{\sqrt{z}}^{\infty} \frac{d}{dz}\frac{r'^2}{2}\,dr'$$

$$= 2\int_{\sqrt{z}}^{\infty} r'\frac{dr'}{dz}\,dr'$$

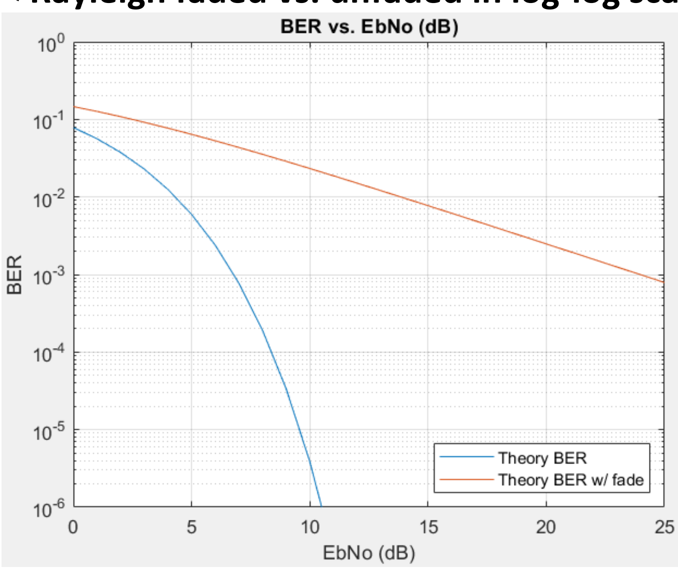
$$= 2\int_{\sqrt{z}}^{\infty} \left(\frac{r}{\sigma} + z^{\frac{1}{2}}\right)\left(\frac{r}{2\sqrt{\sigma}}z + z^{-\frac{3}{2}}\right)^2 dz$$

$$\cong 2\int_{\sqrt{z}}^{\infty} z^{\frac{1}{2}} z^{-3}\,dz, \qquad with\ large\ z$$

$$= 2\int_{\sqrt{z}}^{\infty} z^{-\frac{5}{2}}\,dz = -4/3[z^{-\frac{3}{2}}]_{\sqrt{z}}^{\infty} = 4/3(Z^{-\frac{3}{4}})$$

Log-scaled Rayleigh faded Pe is linearly inverse proportional to log(Eb/No) while unfaded Pe is non-linearly more inverse proportional to Eb/No as follows

< Rayleigh faded vs. unfaded in log-log scale>




```
z_db = 0:26;
z = 10 .^ (z_db/10);
theory_bpsk = 0.5*erfc(sqrt(z));
theory_bpsk_fade = 0.5*(1 - sqrt( z ./ (z
+ 1 ) ) );

figure(1)
semilogy( z_db, theory_bpsk, z_db,
theory_bpsk_fade);

grid on;
xlabel('EbNo (dB)');
ylabel('BER');
title('BER vs. EbNo (dB)');
legend('Theory BER','Theory BER w/
fade','location','SouthEast');
axis([0 26-1 1e-6 1]);
```

4. *Multipath Fading and Equalization*

Consider a received sequence of signal values $y(n)$ for a transmitted sequence of signal values $x(n)$. The received sequence $y(n)$ is the summation of multiple scaled and delayed copies of $x(n)$.

$$y(n) = \sum_{k=0}^{L-1} \alpha_k x(n-k)$$

The values of $\alpha_k = |\alpha_k| e^{j\phi(k)}$ are in general complex.

a) Find the z-transform $Y(z)$ of $y(n)$. (You may need to review Z-transforms.)

$y(n) = \sum_{k=0}^{L-1} \alpha_k x(n-k) = \alpha_0 x(n) + \alpha_1 x(n-1) + \cdots \alpha_{L-1} x(n-L+1)$

$= x(n) \sum_{k=0}^{L-1} \alpha_k \delta(n-k)$

$Y(z) = \Sigma_{n=0}^{N} \sum_{k=0}^{L-1} \alpha_k x(n-k) z^{-n}$

$= \Sigma_{n-k=0}^{N-k} \sum_{k=0}^{L-1} \alpha_k x(n-k) z^{-(n-k)} z^{-k}$

$= X(z) \sum_{k=0}^{L-1} \alpha_k\, z^{-k} = X(z)\alpha(z)$

$= X(z)[\alpha_0 + \alpha_1\, z^{-1} + \cdots \alpha_{L-1} z^{-L+1}]$

$= X(z)[\alpha_0 z^{L-1} + \alpha_1\, z^{L-2} + \cdots \alpha_{L-1}]/z^{L-1}$

b) Find the transfer function of the channel $H(z)$ by writing $Y(z)$ in terms of $X(z)$ (i.e.

$y(n) = x(n) \sum_{k=0}^{L-1} \alpha_k x(n-k)$

$Y(z) = X(z) \sum_{k=0}^{L-1} \alpha_k\, z^{-k}$

$H(z) = \sum_{k=0}^{L-1} \alpha_k\, z^{-k}$

$= \alpha_0 + \alpha_1\, z^{-1} + \cdots \alpha_{L-1} z^{-L+1}$

$= [\alpha_0 z^{L-1} + \alpha_1\, z^{L-2} + \cdots \alpha_{L-1}]/z^{L-1}$

c) Answer the following questions about the impulse response $h(n)$ of the channel:

i. Is the impulse response $h(n)$ linear?

From the above transfer function we can tell channel is linear. Gain or loss to input signal get`s the same gain or loss atfter the transfer function. It`s just like different harmonics with different coefficients.

ii. Is $h(n)$ an FIR (finite-impulse-response) or IIR (infinite-impulse-response) filter?

IIR filter as the linear channel response is like FIR filter and it`s inverse makes IIR.

d) In general, $H(z)$ causes unwanted distortion in the received sequence of signal values $y(n)$. Suppose we wish to “undo” the effects of $H(z)$ by “equalizing” $Y(z)$. To do so, we want to find a equalization filter $G(z)$ such that

$$G(z)\,Y(z) = X(z)$$

Find the form of the equalization filter $G(z)$ in terms of the transfer function $H(z)$

$G(z) = X(z)/Y(z) = 1/H(z) = (1 - z^{-1})/(\sum_{k=0}^{L-1} \alpha_k\, z^{k})$

e) Answer the following questions about the impulse response $g(n)$ of the equalization filter:

i. Is the impulse response $g(n)$ linear?

Yes. As it`s just inverse of the linear FIR filter.

ii. Is $g(n)$ an FIR (finite-impulse-response) or IIR (infinite-impulse-response) filter?

It`s IIR filter with polls generated by inverse of FIR filter’s zeros.

5. *Two Component Fading Model and Equalization*

Suppose you have the following difference equation for the received sequence of signal values

$$y(n): y(n) = a_0 x(n) + a_1 x(n - n_0)$$

$$n_0 = 8$$

a) Find the z-transform for this specific form of $y(n)$, which consists of the “direct path” signal $x(n)$ and a single “image” path signal $x(n - n_0)$. The coefficients for each of these signals are $a_0 = 1$ and $a_1 = -A$ where $0 < A < 1$.

$y(n) = x(n) - Ax(n - n_0)$

$Y(z) = \Sigma_{n=0}^{\infty}(x(n) - Ax(n-8))z^{-n}$

$= X(z)(1\text{-}Az^{-8})$

b) Determine the transfer function $H(z)$ of the channel.

$H(z) = 1\text{-}Az^{-8} = (z^4 - Az^{-4})/z^{-4},$

c) Evaluate $H(z)$ on the unit circle using ($|z| = |e^{jw}| = 1$) so that $z = e^{jw}$.

(* $H(e^{jw}) = (2/e^{-j4w})(e^{j4*w} - Ae^{-j4*w})/2$)

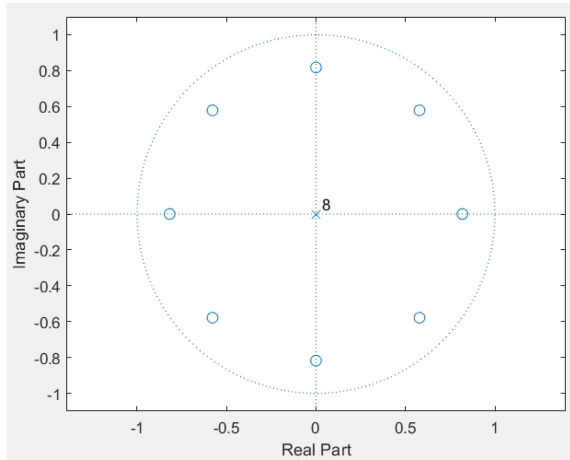
($for\ simplicity\ if\ set\ A = 1$)

($H(e^{jw}) = (2/e^{-j4w})(e^{j4*w} - e^{-j4*w})/2 = 2e^{j4w}\sin(4w)$)

i. Sketch the pole-zeros plot for $H(z)$

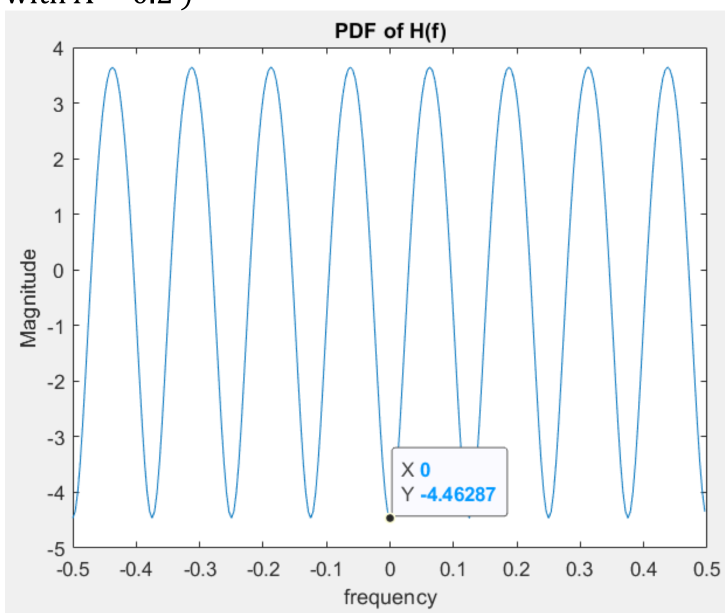
$H(z) = 1\text{-}Az^{-8}$

h(n) = [1 0 0 0 0 0 0 0 -A] in MatLab. When set A = 0.2



ii. Sketch $|H(e^{jw})|^2$.

(*with A = 0.2)

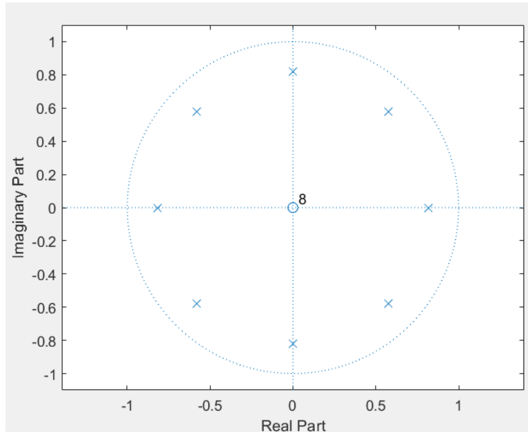


d) Find an expression for the transfer function $G(e^{jw})$ of the equalization filter .

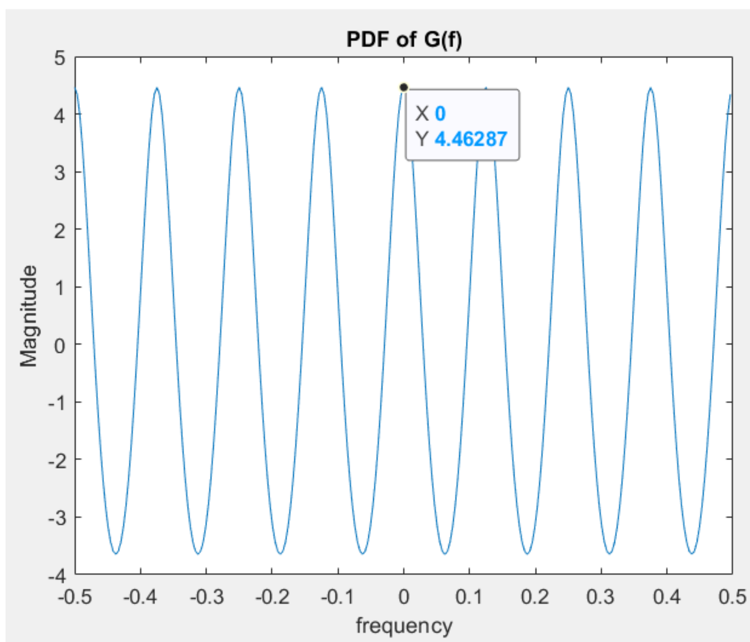
G(z) = $1/(1\text{-}Az^{-8}) = z^8/(z^8\text{-}A)$

i. Sketch the pole-zeros plot for $G(z)$

G(z) = 1/H(z), can be plotted by zplane(1,H(z)).

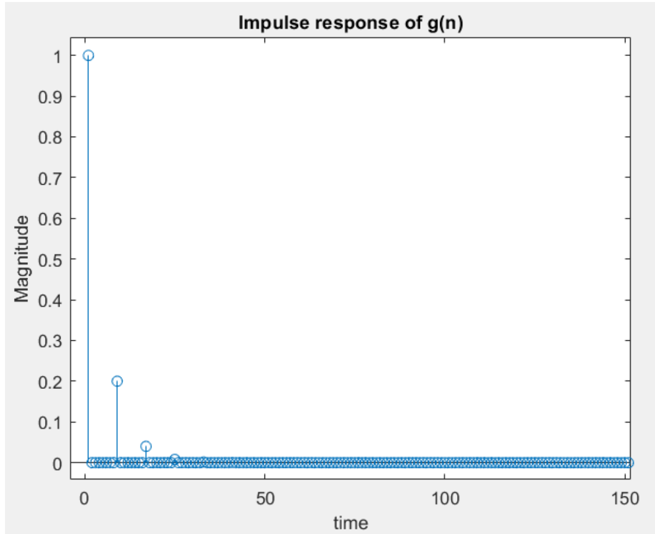


ii. Sketch $|G(e^{jw})|^2$.



e) Find an expression for the impulse response $g(n)$ of the equalization filter.

i. Sketch the impulse response $g(n)$



e) Comment on your results, how does A and n_0 affect the frequency response of $H(e^{jw})$?

Find zero forcing equalizer works OK by 1/h(n).

A: affects the poles & zeros location. Smaller A makes poles and zeros close to (0,0) while bigger A make them closer to circle of one.

n0: determines how many poles and zeros are in a Nyquist zone. Bigger n0 makes more poles or zeros proportional to it.

1. *Central Limit Theorem*

In this matlab problem, you will be validating the central limit theorem in simulation. Understanding the central limit theorem will be useful for analyzing multipath fading channel models in Lab 4. A summary of the central limit theorem is as follows:

$$Y = \frac{1}{N}\sum_{i=1}^{N} X_i$$

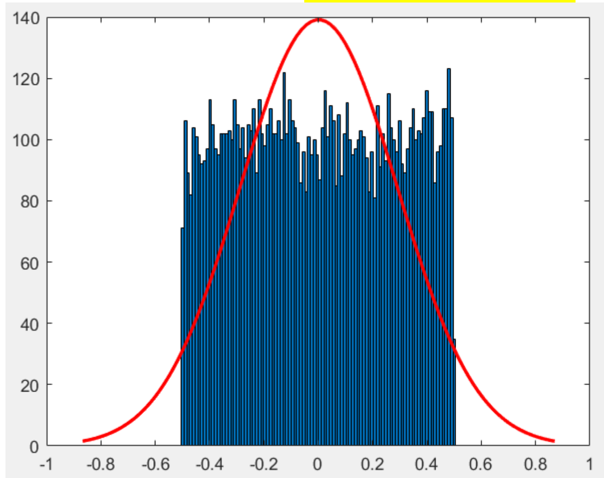
For a set of random variables X_i that are i.i.d (independent identically distributed), the sum random variable Y approaches a normal distribution. In the first part, we will be assuming that the random variables X_i are distributed uniformly (Note: In actuality, we could use any type of random variable and the fundamental result won't change).

a) Generate a two dimensional random vector of uniform random variables $X \sim U[-0.5\ 0.5]$.

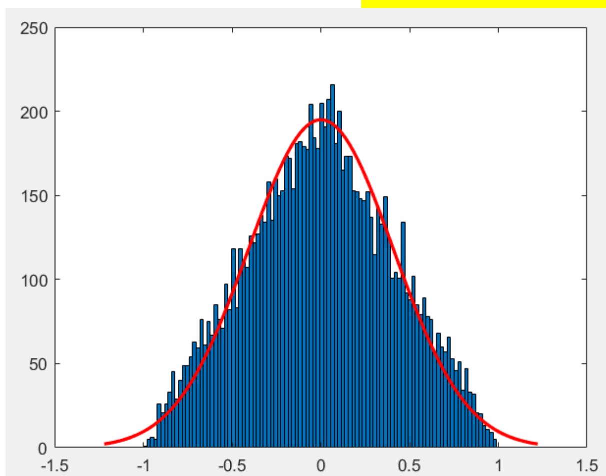
$X = rand(1000, 10000) - 0.5$

this will create a 1000 by 10000 matrix of uniform random numbers.

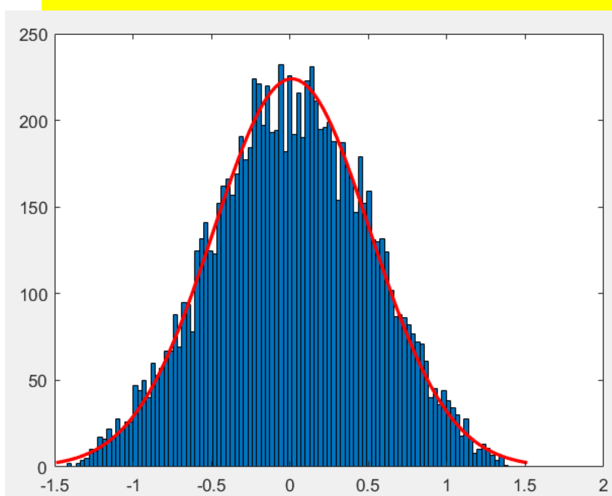
i. Plot the histogram (i.e. use the function ***histfit(.)***) of one row of this random vector (i.e. $Y = X_1$). What kind of distribution is this? Uniform distribution.



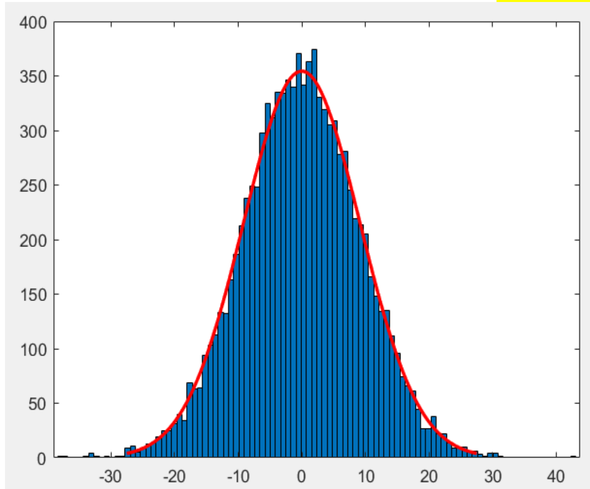
ii. Plot the histogram of the sum of any two rows of X (i.e. this matlab command models: $Y = X_1 + X_2$). What kind of distribution is this? Normal distribution.



iii. Plot the histogram of the sum of any three rows of X (i.e. this matlab command models: $Y = X_1 + X_2 + X_3$). More fit to normal distribution than the summation of two.



iv. Plot the histogram of the sum of all 1000 rows of X (i.e. this matlab command models: $Y = X_1 + X_2 + \ldots + X_{1000}$). What kind of distribution this? Normal distribution.



v. Comment on your results (i.e. how well does the histogram conform to a normal distribution. And also, how fast does it converge?).

It converges very fast. It starts to show close to normal distribution from the summation of 2 Xi's.

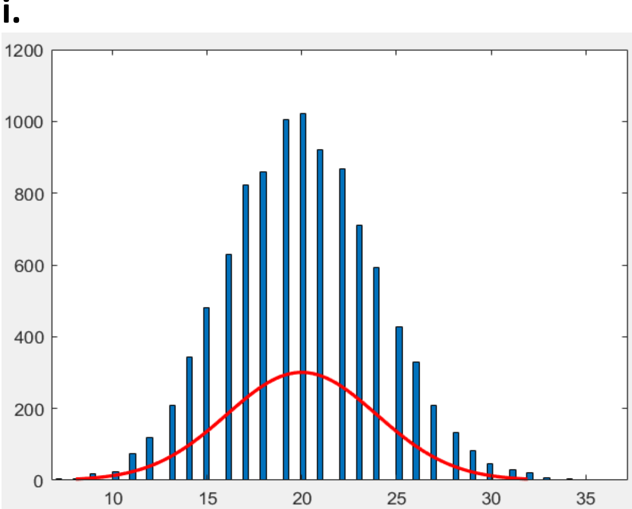
=====< used code>==========================

```
X = rand(1000,10000)-0.5;
Y(1,:) = zeros(1,10000);
for n=1:1000
   Y(1,1:1:end) = Y(1,1:1:end)+X(n,1:1:end);
end
histfit(Y)
```

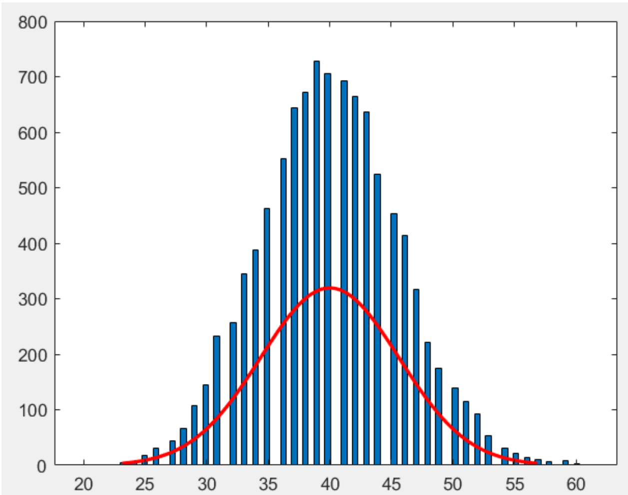
==

b) Repeat part (a) for any random distribution of your choice (i.e. choose from ***binornd(.)***, ***randsrc(.), round(rand()), raylrnd(.), etc..).***

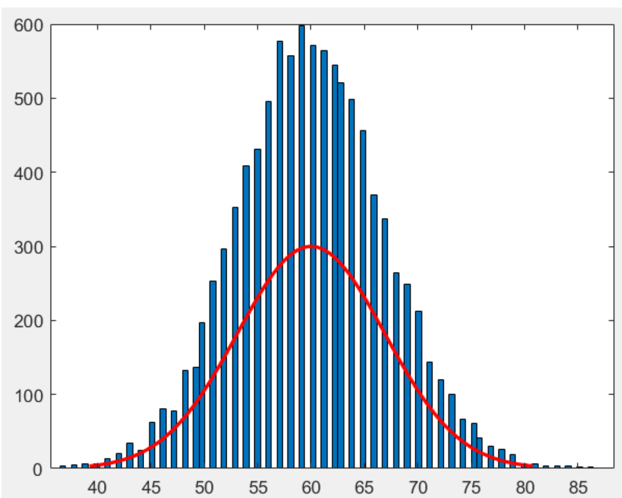
i.



ii.

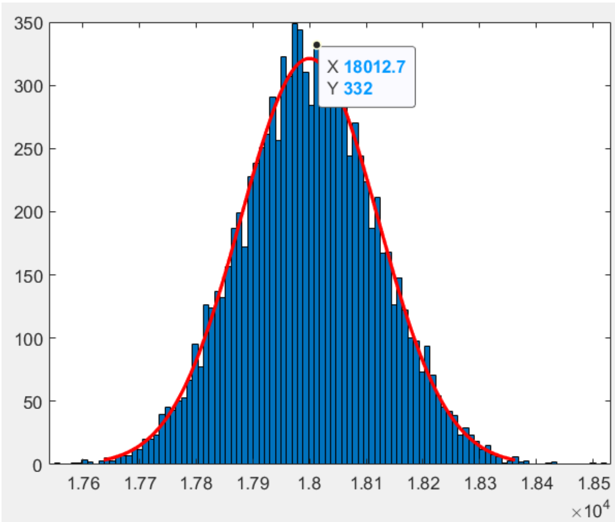


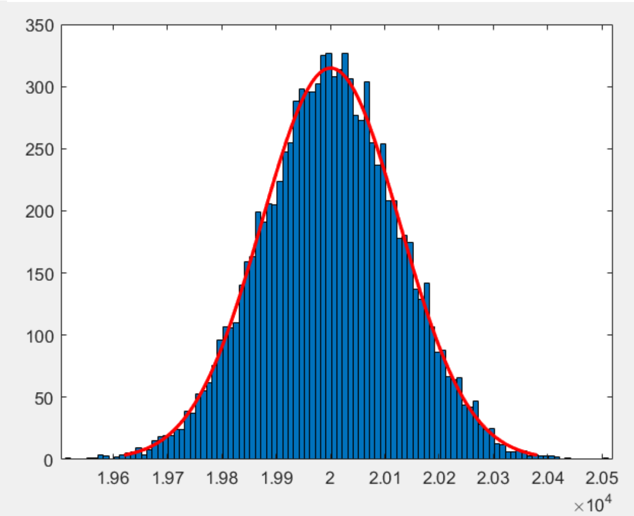
iii.



iv.

< summation of 900 Xi's > **< summation of 1000 Xi's >**



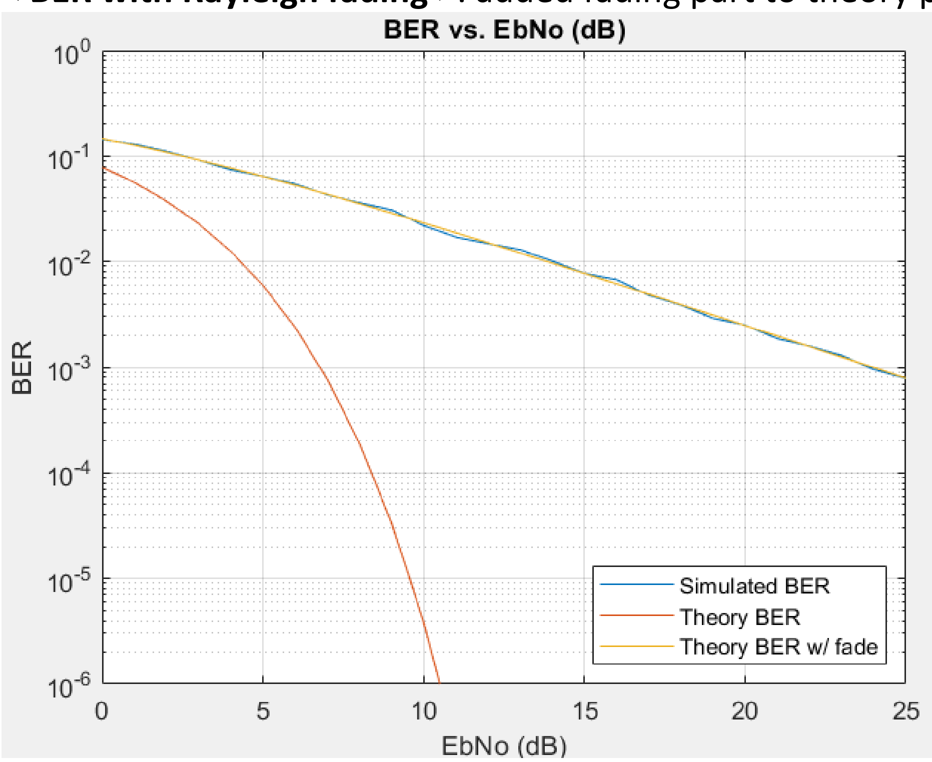


v.

It converges very slowly. It`s not close to normal distribution even with summation of 900 Xi`s.

2. *Fading simulation*

< BER with Rayleigh fading >: added fading part to theory problem 3 code



3. *Least Squares Channel Estimation*

In this matlab exercise, you will be estimating the channel impulse response and performing channel equalization on a vector of noisy received samples.

a) First, load the data file "p5_msg.mat". Parameters of interest are as follows:

$f_s = 20\text{MHz}$

$T_s = 1/\ f_s = 50\text{ns}$

$T_g = 1.6\mu\text{s}$

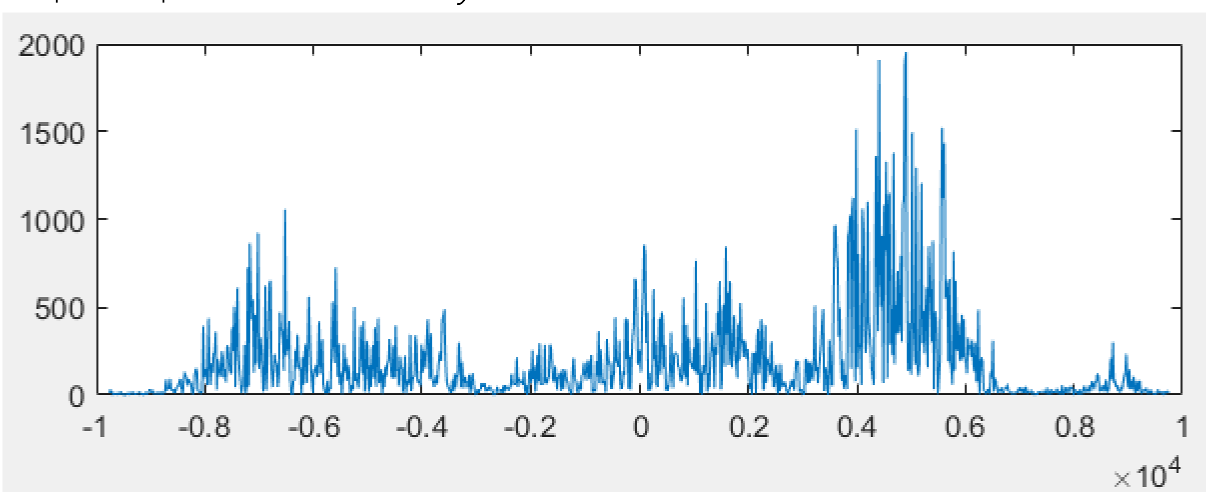
$T_t = 6.35\mu\text{s}$

The message contained in variable y consists of a data packet sampled at $f_s = 20\text{MHz}$ that has been corrupted by a typical indoor fading channel (i.e. such as the WiFi channel). The packet consists of the following: a BPSK pseudo-noise preamble of length T_t followed by a guard interval of duration T_g and the modulated payload of length T_p.

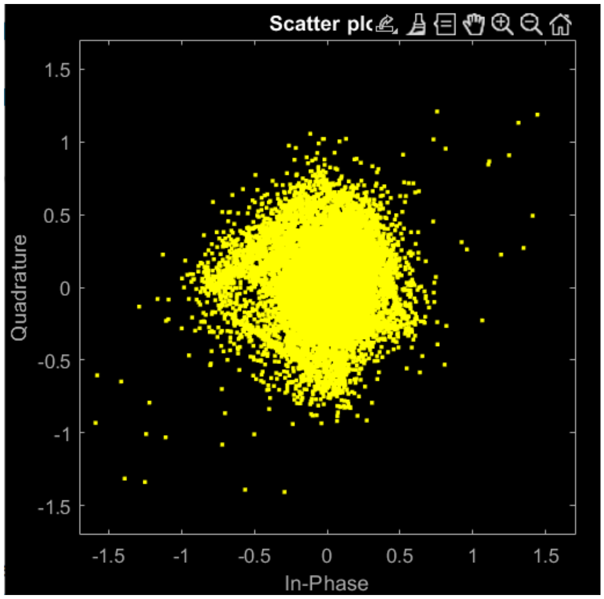
The order $M = 7$, PN sequence was generated using ***pngen.m,*** which can be found in the labs folder.

b) Since the data symbols have been corrupted by a fading channel, it is almost impossible to detect the symbols without applying any sort of equalization. Verify the effect of the channel response by plotting:

i. The power spectrum of the vector y.



ii. The constellation of vector y.



c) Using the linear model as discussed in ***Lecture 4B*** and given below:

$\boldsymbol{r} = \mathbf{T}\boldsymbol{h} + \boldsymbol{n}$

$$\mathbb{T} = \begin{bmatrix} t_0 & 0 & \cdots & 0 \\ t_1 & t_0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ t_{2J} & t_{2J-1} & \cdots & t_0 \\ \vdots & & & \vdots \\ t_{K-1} & t_{K-2} & \cdots & t_{K-L} \end{bmatrix}.$$

$\mathbf{T}_{ij} = t_{i-j}$ *for* $i = 0, 1, \cdot \cdot \cdot, K-1; j = 0, 1, \cdot \cdot \cdot, L-1.$

find the least-squares estimate of the channel coefficients given by $\hat{\boldsymbol{h}}$:

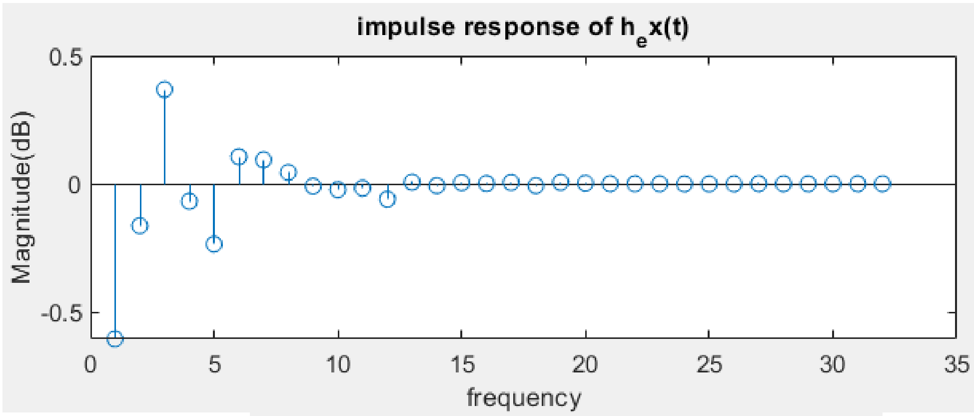
$$\hat{\mathrm{h}} = \left(\mathbb{T}^{\dagger}\mathbb{T}\right)^{-1}\mathbb{T}^{\dagger}\mathbf{r}.$$

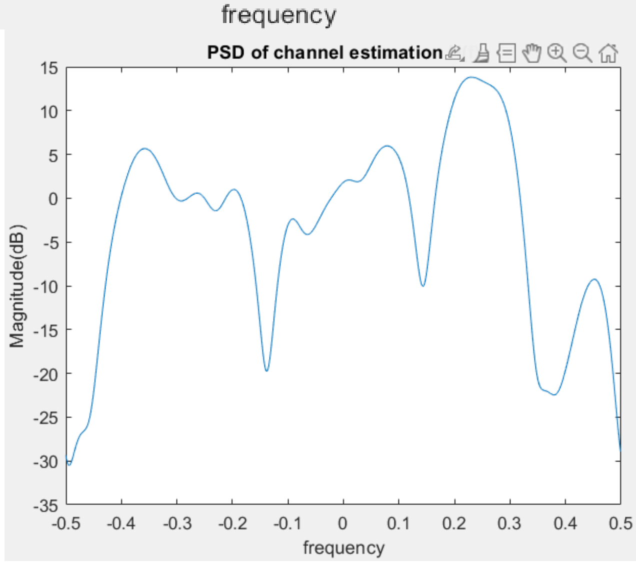
$$or$$

$$\hat{h} = pinv\,(\mathbb{T})\,r$$

where *pinv*(.) denotes the pseudo-inverse. Note that, if $\mathbf{T}$ is invertible, then *pinv* ($\mathbf{T}$) = $\mathbf{T}^{-1}$. You may assume that the guard interval T_g is equal to the length of the channel response.

i. Plot the time series $\hat{h}(n)$ and the power spectrum $\hat{H}(e^{j\omega})$.





d) While the previous approach gives us the least-squares estimate of the channel coefficients $h(n)$ it does not directly give us the coefficients of the equalization filter $g(n)$. One approach to finding the equalization filter coefficients $g(n)$ is by forming another linear equation which describes the convolution of the received samples $\boldsymbol{r}$ and the equalization filter $g(n)$ as given below in matrix notation:

$\mathbf{R}\boldsymbol{g} = \mathbf{t} + \boldsymbol{n}$

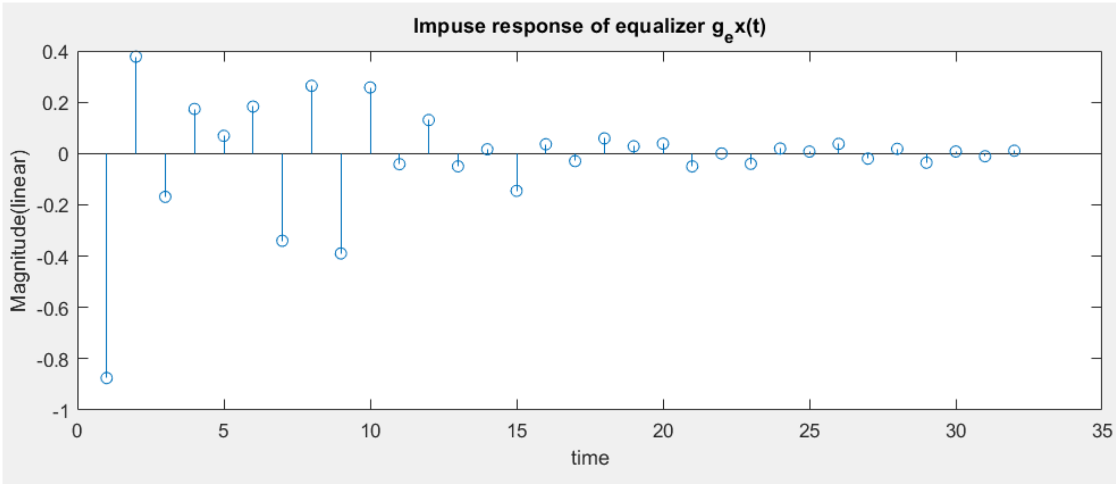
where $\boldsymbol{g}$ is the equalization filter, $\mathbf{R}$ is the matrix representation of the received signal vector $\boldsymbol{r}$ and $\boldsymbol{t}$ is training sequence vector.

i. Find the least-squares estimate $\hat{\boldsymbol{g}}$, which is given by:

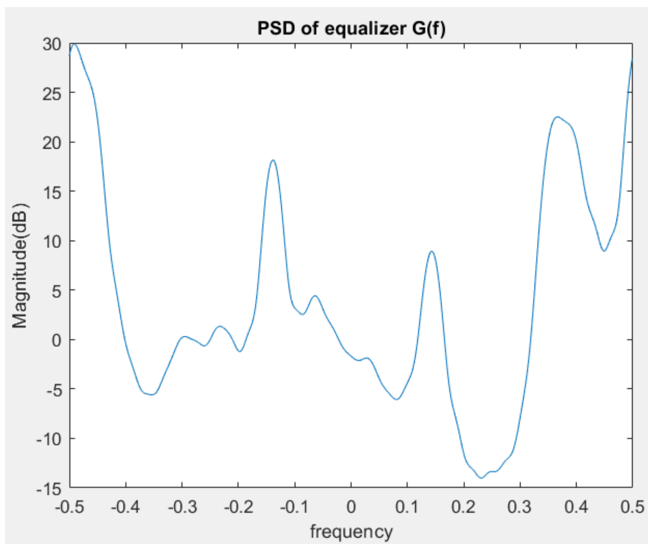
$$\hat{g} = \left(\mathbb{R}^{\dagger}\mathbb{R}\right)^{-1}\mathbb{R}^{\dagger}t$$

$$or$$

$$\hat{g} = pinv\left(\mathbb{R}\right)t$$



ii. Plot the time series $\hat{g}(n)$ and the power spectrum $\hat{G}(e^{j\omega})$.

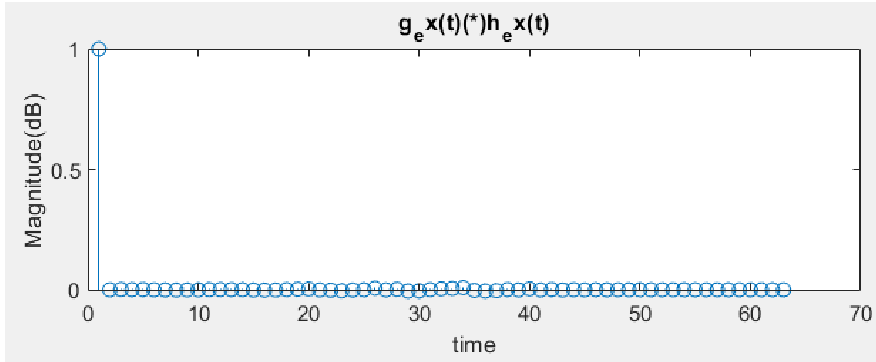


e) Recall that given a fading channel with impulse response $h(n)$ and frequency response $H(e^{j\omega})$, the goal of the equalization filter $g(n)$ is to remove the unwanted distortion caused by the channel $h(n)$. Therefore, one requirement for the ideal equalization filter is that:

$$h(n) * g(n) = \delta(n)$$

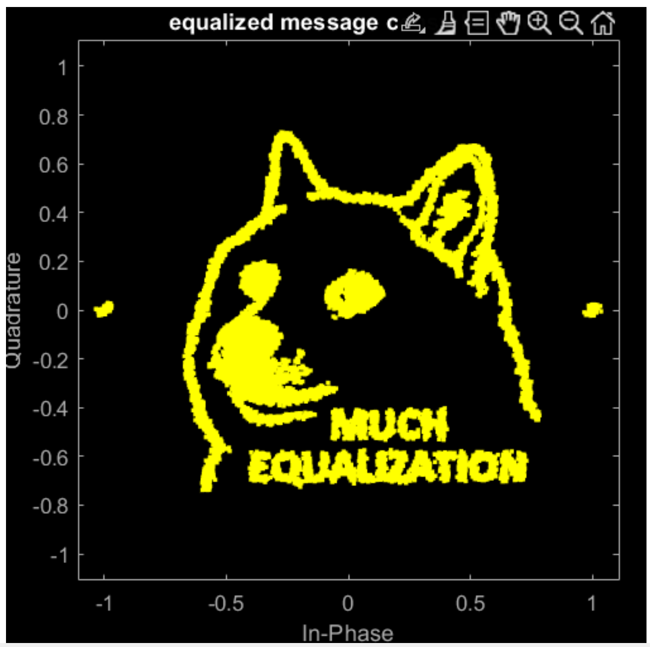
the convolution of $h(n)$ and $g(n)$ should give us a delta function $\delta(n)$.

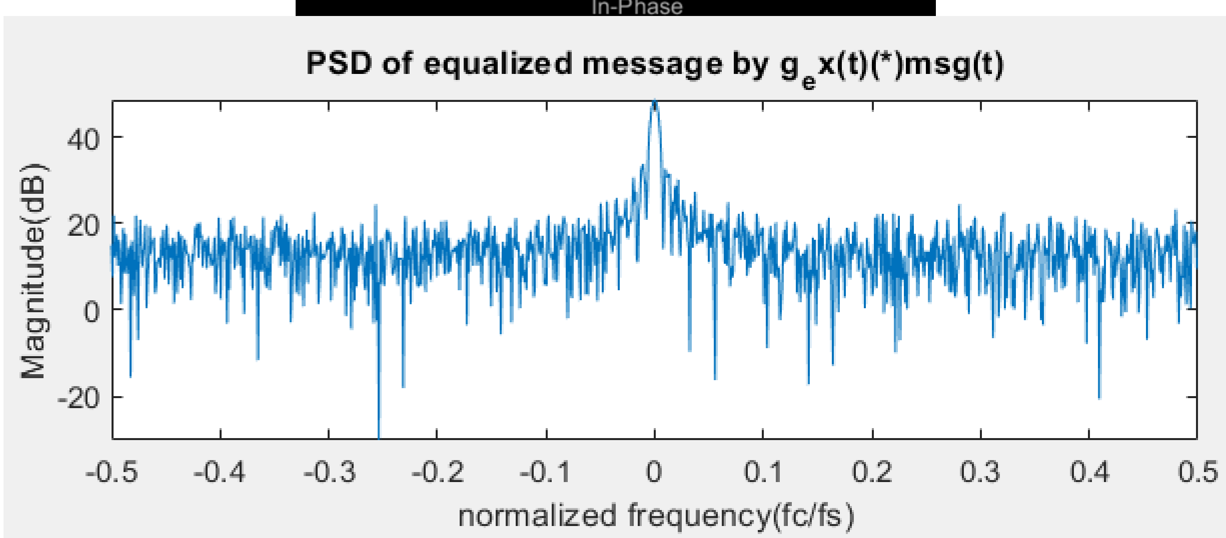
i. Plot the time-series $\hat{h}(n) * \hat{g}(n)$.



f) Finally, apply the equalization filter ˆ$g(n)$ to the packet and verify that the equalization. process has successfully removed the channel response $h(n)$.

i. Plot the equalized constellation $y_{eq}(n)$ and the corresponding power spectrum $Y_{eq}(e^{j\omega})$.





ii.

ii. Comment on your results (i.e. what type of modulation is used for the payload? etc.).

: it`s a dog or bear. As the constellation has big degree of freedom, it seems like big number of QAM is used.

6

<used code for theory problem 5>

```
fftlen = 256;
A = 0.2;
h_n = [ 1 0 0 0 0 0 0 0 -A];
[H_f,W]=freqz(h_n,1,fftlen,'whole');

figure(1)
zplane(h_n,1)

figure(2)
plot(-0.5:1/fftlen:0.5-1/fftlen,20*log(abs(H_f)))
title('PDF of H(f)')
ylabel('Magnitude')
xlabel('frequency')

figure(3)
zplane(1,h_n)

figure(4)
[G_f,W]=freqz(1, h_n, fftlen,'whole');
plot(-0.5:1/fftlen:0.5-1/fftlen, 20*log(abs(G_f)))
title('PDF of G(f)')
ylabel('Magnitude')
xlabel('frequency')

figure(5)
stem(g_n)
zplane(ifft(G_f,fftlen),1)

%%
fftlen = 1024;
n = 0:1:500;
fc = 1;
fs = fftlen/4;

figure(6)
[G_f,W]=freqz(1, h_n, fftlen, "whole");
g_n = ifft(G_f)
stem(g_n)
```

```
title('Impulse response of g(n)')
ylabel('Magnitude')
xlabel('time')

figure(7)
check_n = conv(h_n,g_n);
stem(check_n)
figure(8)
G_f = fftshift(20*log(abs(fft(g_n,fftlen))));
plot(-0.5:1/fftlen:0.5-1/fftlen, G_f)
```

<Used code for Matlab problem 1>

```
X = binornd(100,0.2,1000,10000
Y(1,:) = zeros(1,10000);
for n=1:900
   Y(1,1:1:end) = Y(1,1:1:end)+X(n,1:1:end);
end
histfit(Y)
```

<Used code for Matlab problem 2>

(: reused the prelab2 solution code with unnecessary codes removal)

```
Ls = 26;                    % Number of 1db Ebno steps to sweep
numSymbols = 10000;         % data vector size
ebno_db = 0:Ls-1;           % generate ebno
ebno = 10 .^ (ebno_db/10);  % generate ebno in linear scale
BER = zeros(1,Ls);          % pre-allocate BER vector

fading = 1;                 % apply fading
threshold = 0;              % decision threshold
clock_offset = 0;           % clock offset in samples
N = 1;                      % Samples Per Symbol
h_ps = ones(1,N);           % Pulse Shaping Filter
h_mf = h_ps/N;              % Matched Filter

% pre-allocate vectors
bits = zeros(1, numSymbols);  % bits
rBits = zeros(1,numSymbols);  % received bits
n = zeros(1,numSymbols*N);    % noise vector
x = zeros(1,numSymbols*N);    % data vector
e = zeros(1,numSymbols*N);    % error vector
y = zeros(1,numSymbols*N);    % received vector
r = y;                        % output of matched filter

% ensure clock_offset value does not go out of index
clock_offset = mod(abs(clock_offset),N);

%sweep ebno
for i=1:Ls

    numErrors = 0;
    numBits = 0;
```

```
    % gen noise, note: Eb is N, sigma = No/2
    sigma = sqrt(N/(2*ebno(i)));

    % generate rayleigh sigma
    sig_r = sqrt(N/2);

    while( numErrors < 500 )
        % gen data bits
        bits = sign(randn(1,numSymbols));

        x = upsample(bits,N);       % upsample to N samples per symbol
        x = filter(h_ps,1,x);       % pulse shape

        % gen noise
        n = sigma*randn(1,N*numSymbols);

        % apply fading / noise
        if( fading )
            r1 = sig_r*randn(1,N*numSymbols);
            r2 = sig_r*randn(1,N*numSymbols);
            R = sqrt(r1 .^ 2 + r2 .^ 2);

            y = x .* R + n;
        else
            y = x + n;
        end

        % apply matched filter

        r = filter(h_mf,1,y);

        % clock offset
        rBits = r( N*(1:numSymbols) - clock_offset );

        % threshold and compare bits
        e = abs( (rBits > threshold) - (bits > 0) );

        % accumulate errors / bits
        numBits = numBits + numSymbols;
        numErrors = numErrors + sum(e);
    end

    BER(i) = numErrors/numBits;
end

%compute theory curves
theory_bpsk = 0.5*erfc(sqrt(ebno));
theory_bpsk_fade = 0.5*(1 - sqrt( ebno ./ (ebno + 1 ) ) );

theory_bpsk_offset = 0.5*0.5*erfc(sqrt(ebno)*(1-2*clock_offset/N))+0.5*0.5*erfc(sqrt(ebno));


% plots
figure(1);
semilogy( ebno_db,BER,ebno_db, theory_bpsk, ebno_db, theory_bpsk_fade);
```

```
grid on;
xlabel('EbNo (dB)');
ylabel('BER');
title('BER vs. EbNo (dB)');
legend('Simulated BER','Theory BER','Theory BER w/ fade','location','SouthEast');
axis([0 Ls-1 1e-6 1]);
```

<Used code for Matlabg problem 4>

```
clear all;close all;clc;

fftlen = 1024;
fs = 20*10^6; %20MHz
Ts =1/fs ;%50nsec
Tg = 1.6*10^-6  ;%1.63us
Tt = 6.35*10^-6 ;%6.35us
msg = load("p5_msg_revised.mat");
MSG = fft(msg.y,fftlen);
P_MSG = fftshift(abs(MSG).^2);

figure(1)
subplot(211)
plot((-0.5:1/fftlen:0.5-1/fftlen)*fs/fftlen,P_MSG)
title('massage spectrum')
ylabel('Magnitude')
xlabel('frequency')
subplot(212)
stem(1:1:127,msg.y(1:1:127));
axis([1 127 -2 2])
title('Massage preambgle in time domain')
ylabel('Magnitude')
%figure(2)
scatterplot(msg.y,2)
title('Massage constellation')
%%==============================================================
%%< T matrix >
T = zeros(127,32);
dm = pngen(7,127).*2-1;
T= toeplitz(dm(1:1:127),[dm(1) zeros(1,31)]);
%%==================================================================
% <h_ex>
msg_127= msg.y(1:1:127)
h_ex = pinv(T)*msg_127.'
H_ex = fft(h_ex,fftlen);
figure(3)
plot(-0.5:1/fftlen:0.5-1/fftlen,fftshift(10*log(abs(H_ex))))
title('H(f)')
ylabel('Magnitude(dB)')
xlabel('frequency')
%%==================================================================
% <check h_ex>
msg_127_check = T*h_ex;
figure(4)
```

```
subplot(211)
stem(msg_127)
title('original message preamble')
subplot(212)
stem(msg_127_check)
title('preambgle recovered by H(f)')
%%==============================================================
%%< R matrix >
R= toeplitz(msg_127(1:1:127),[msg_127(1) zeros(1,31)]);
%%==================================================================
% <g_ex>
dm = pngen(7,127)*2-1;
g_ex = pinv(R)*dm.'
G_ex = fft(g_ex,fftlen);
figure(5)
plot(-0.5:1/fftlen:0.5-1/fftlen,fftshift(10*log(abs(G_ex))))
title('G(f)')
ylabel('Magnitude(dB)')
xlabel('frequency')
%%====================q============================================
% < h_ex*g_ex >
eq_check = conv(h_ex,g_ex);
EQ_check = fft(eq_check,fftlen);
figure(6)
subplot(211)
stem(eq_check)
title('g_ex(t)(*)h_ex(t)')
ylabel('Magnitude(dB)')
xlabel('time')
subplot(212)
plot(-0.5:1/fftlen:0.5-1/fftlen,fftshift(10*log(abs(EQ_check))))
title('G_ex(f)*H_ex(f)')
ylabel('Magnitude(dB)')
xlabel('frequency')
%%====================q============================================
% < g_ex(*)msg.y >
msg_check = conv(g_ex,msg.y);
MSG_check = fft(msg_check,fftlen);
figure(7)
subplot(211)
stem(msg_check)
title('equalized message by g_ex(t)(*)msg(t)')
ylabel('Magnitude(dB)')
xlabel('time')
subplot(212)
plot(-0.5:1/fftlen:0.5-1/fftlen,fftshift(10*log(abs(EQ_check))))
title('equalized message spectrum by G_ex(f)*H_ex(f)')
ylabel('Magnitude(dB)')
xlabel('frequency')
%%====================q============================================
% < equalized messagbe constellation >]
scatterplot(msg_check)
```