**Data Communication**

**(CAECC12)**

***Lab Practicals***

Nupur Sudhakar

2019UCS2048

CSAI

***INDEX***

|  |  |  |
| --- | --- | --- |
| ***S.No.*** | ***Topic*** | ***Page No.*** |
| 1 | Introduction |  |
| 1. To perform matrix computation | 1 – 2 |
| 1. To plot sine wave of frequency 200 Hz. | 3 – 4 |
| 1. To plot a pulse of width 10. | 5 – 6 |
| 1. To plot the spectrum (amplitude and phase) of the pulse generated above. | 7 – 9 |
| 2 | To generate uniform distributed random number and plot its density function. Also find its mean and variance. | 10 – 11 |
| 3 | To generate Gaussian distributed random number and plot its density function. Also find its mean and variance. | 12 – 13 |
| 4 | Compute the signal to quantisation noise ratio of Uniform Quantisation. Plot SNQR versus Quantisation levels. | 14 – 16 |
| 5 | Study of passband digital communication technique BPSK. Calculate the BER of BPSK modulated signal. | 17 – 19 |
| 6 | Given is a linear block code with the generator matrix G :-  1 1 0 0 1 0 1  G = 0 1 1 1 1 0 0  1 1 1 0 0 1 1   1. Calculate the number of valid code words N and the code rate RC. Specify the complete Code set C. 2. Determine the generator matrix G′ of the appropriate systematic (separable) code C’. 3. Determine the syndrome table for single error. | 20 – 24 |

**Practical - 1**

**a)**

**Aim**

To perform matrix computation

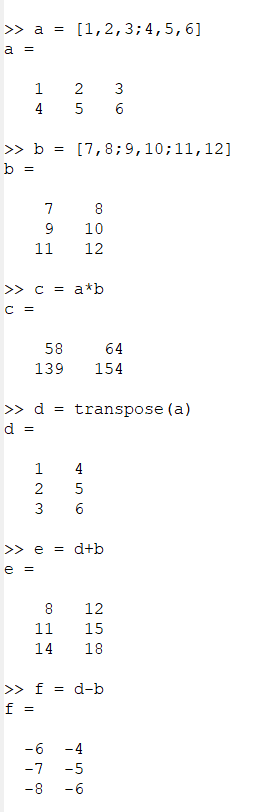
**Theory**

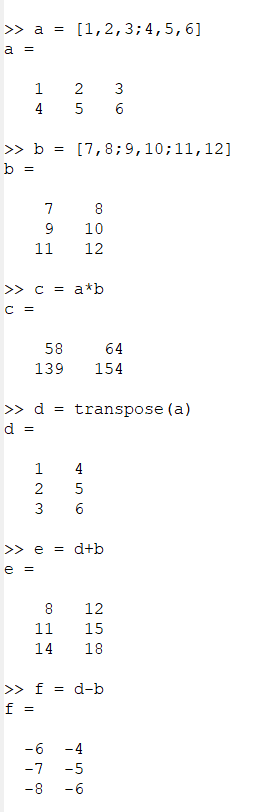
A matrix is a rectangular array or table of numbers arranged in rows and columns. We can perform various operations on matrices like addition, subtraction multiplication, inverse, transpose etc.

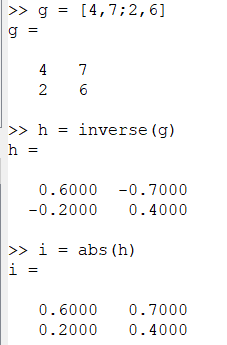
**Software used**

Octave

**Code & Output**







**Result**

We have successfully performed matrix computation.

**b)**

**Aim**

To plot sine wave of frequency 200 Hz.

**Theory**

The sine wave is the most fundamental form of a periodic analog signal. When we visualize it as a simple oscillating curve, its change over the course of a cycle is smooth and consistent, a continuous, rolling flow. Each cycle consists of a single arc above the time axis followed by a single arc below it. A sine wave can be represented by three parameters: the peak amplitude, the frequency and the phase.

**Software used**

Octave

**Code**

t=[0:0.0001:0.005];

f=200;

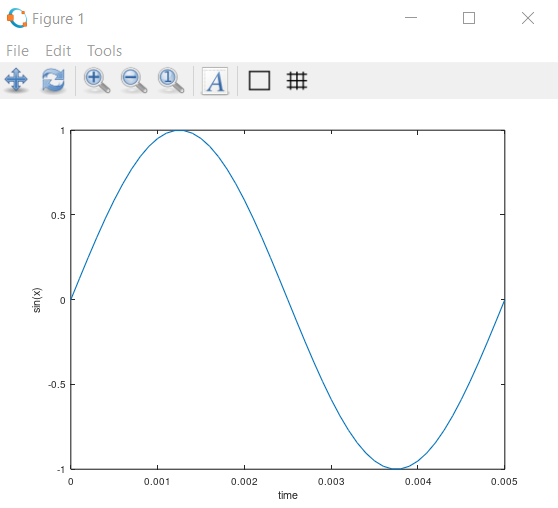
x=sin(2\*pi\*f\*t);

plot(t,x);

xlabel("time");

ylabel("sin(x)");

**Output**



**Result**

A sine wave of frequency 200 Hz has been successfully plotted.

**c)**

**Aim**

To plot a pulse of width 10.

**Theory**

A pulse in signal processing is a rapid, transient change in the amplitude of a signal from a baseline value to a higher or lower value, followed by a rapid return to the baseline value. The pulse width is a measure of the elapsed time between the leading and trailing edges of a single pulse of energy.

**Software used**

Octave

**Code**

x=[-10:0.005:10];

y=sign(x+5)-sign(x-5);

z=y\*0.5;

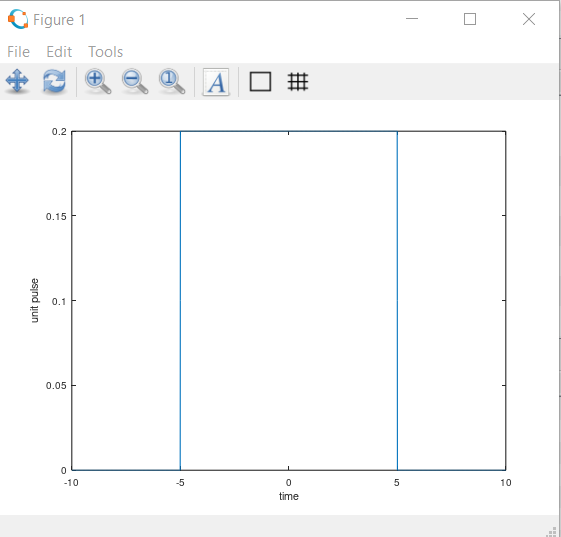
z=z\*0.2;

plot(x,z);

xlabel("time");

ylabel("unit pulse");

**Output**



**Result**

A pulse of width 10 has been successfully plotted.

**d)**

**Aim**

To plot the spectrum (amplitude and phase) of the pulse generated above.

**Theory**

The amplitude spectrum simply gives amplitude at each frequency. The phase spectrum simply gives the phase at each frequency.

**Software used**

Octave

**Code**

* ***Amplitude spectrum***

x=[-10:0.005:10];

y=sign(x+5)-sign(x-5);

z=y\*0.5;

z=z\*0.2;

k=fft(z);

c=fftshift(k);

plot(x,c);

axis([-0.15,0.15]);

xlabel("frequency");

ylabel("amplitude spectrum");

* ***Phase spectrum***

x=[-10:0.005:10];

y=sign(x+5)-sign(x-5);

z=y\*0.5;

z=z\*0.2;

k=fft(z);

c=fftshift(k);

g=angle(c);

plot(x,g);

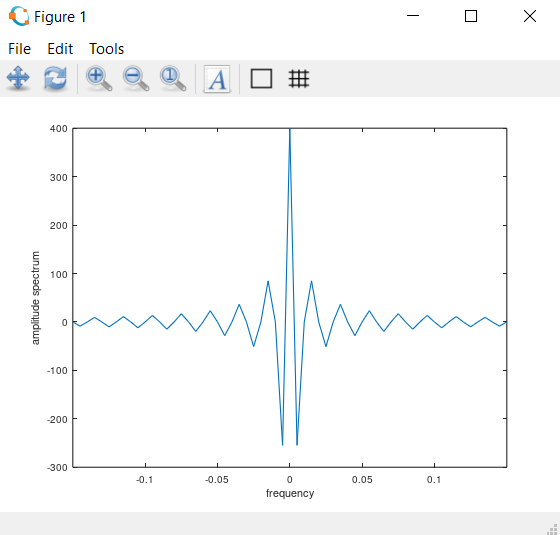
axis([-0.15,0.15]);

xlabel("frequency");

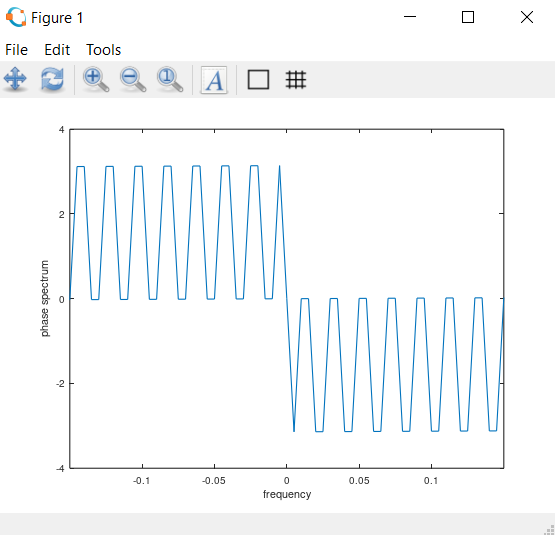
ylabel("phase spectrum");

**Output**

* ***Amplitude spectrum***



* ***Phase spectrum***



**Result**

Amplitude and phase spectrums of the pulse have been successfully plotted.

**Practical - 2**

**Aim**

To generate uniform distributed random number and plot its density function. Also find its mean and variance.

**Theory**

Auniform distribution, also called a rectangular distribution, is a probability distribution that has constant probability.

The [expected value](https://www.statisticshowto.com/probability-and-statistics/expected-value/) (i.e. the [mean](https://www.statisticshowto.com/probability-and-statistics/statistics-definitions/mean-median-mode/#mean)) of a uniform random variable X is:

E(X) = (1/2).(a + b)

The [variance](https://www.statisticshowto.com/probability-and-statistics/descriptive-statistics/sample-variance/) of a uniform random variable is:

Var(x) = (1/12).(b-a)2

**Software used**

Octave

**Code**

N = 100000

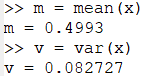
x = rand(1,N);

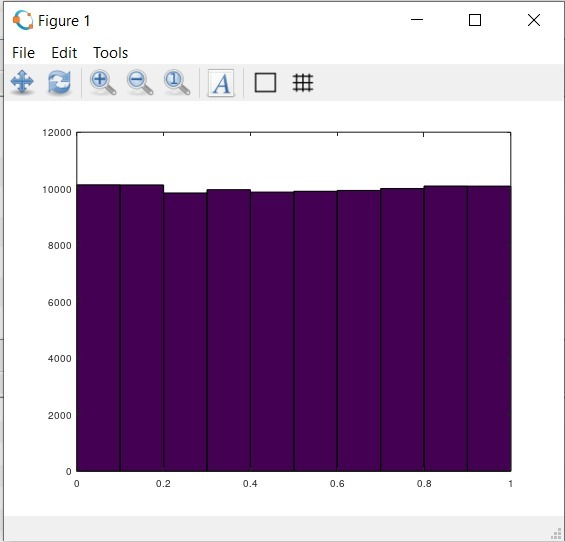
hist(x)

m = mean(x)

v = var(x)

**Output**





**Result**

Uniform distributed random number has been generated and its density function has been successfully plotted.

**Practical - 3**

**Aim**

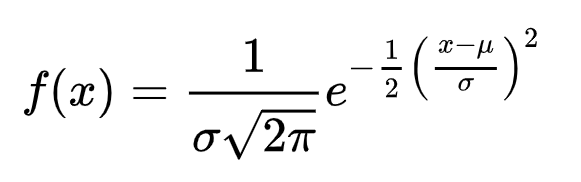
To generate Gaussian distributed random number and plot its density function. Also find its mean and variance.

**Theory**

A Gaussian (or normal) distribution is an arrangement of a data set in which most values cluster in the middle of the range and the rest taper off symmetrically toward either extreme.

A graphical representation of a normal distribution is sometimes called a bell curve because of its flared shape. The precise shape can vary according to the distribution of the population but the peak is always in the middle and the curve is always symmetrical. In a Gaussian distribution, the mean, mode and median are all the same.

The general form of its [probability density function](https://en.wikipedia.org/wiki/Probability_density) is –

{\displaystyle f(x)={\frac {1}{\sigma {\sqrt {2\pi }}}}e^{-{\frac {1}{2}}\left({\frac {x-\mu }{\sigma }}\right)^{2}}} 

The parameter {\displaystyle \mu }μ is the [mean](https://en.wikipedia.org/wiki/Mean) or [expectation](https://en.wikipedia.org/wiki/Expected_value) of the distribution while the parameter {\displaystyle \sigma }σ is its [standard deviation](https://en.wikipedia.org/wiki/Standard_deviation). The [variance](https://en.wikipedia.org/wiki/Variance) of the distribution is σ2.

**Software used**

Octave

**Code**

N = 100000

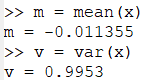
x = randn(1,N);

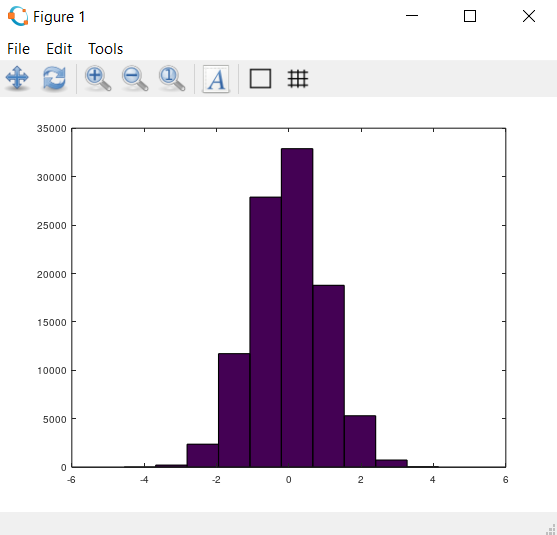
hist(x)

m = mean(x)

v = var(x)

**Output**

****



**Result**

Gaussian distributed random number has been generated and its density function has been successfully plotted.

**Practical - 4**

**Aim**

Compute the signal to quantisation noise ratio of Uniform Quantisation. Plot SNQR versus Quantisation levels.

**Theory**

The digitization of analog signals involves the rounding off of the values which are approximately equal to the analog values. The method of sampling chooses a few points on the analog signal and then these points are joined to round off the value to a near stabilized value. Such a process is called as Quantization.

The type of quantization in which the quantization levels are uniformly spaced is termed as a Uniform Quantization.

SQNR, short for signal to [quantization](https://www.sweetwater.com/insync/quantization/) noise ratio, is a measure of the quality of the quantisation and or digital conversion of an analog signal.

**Software used**

Octave

**Code**

levels = 1:1000;

amplitude = 5;

frequency = 1;

time = 0:0.001:1;

x = amplitude\*sin(2\*pi\*frequency\*time);

mx = max(x);

mn = min(x);

step = (mx-mn)./levels;

for i=levels

in = floor((x-mn)/step(i));

xq = mn+ in\*step(i);

noise = xq - x;

rmsofnoise = var(noise);

power = amplitude\*amplitude/2;

sqnr(i) = power/rmsofnoise;

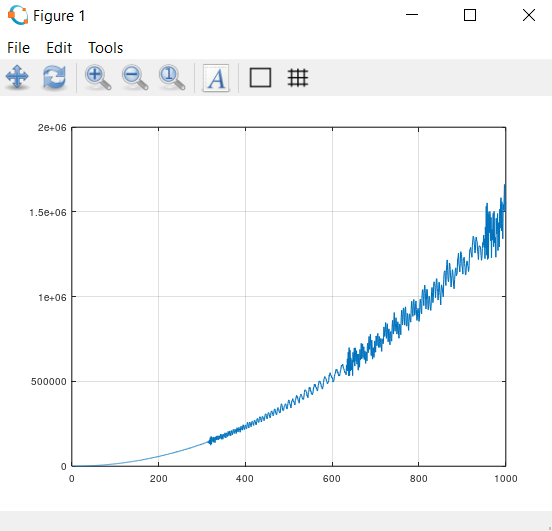
end

sqnrPractical = sqnr;

plot(levels, sqnrPractical);

grid on;

**Output**



**Result**

The signal to quantisation noise ratio of Uniform Quantisation has been computed and SNQR versus Quantisation levels have been plotted.

**Practical - 5**

**Aim**

Study of passband digital communication technique BPSK. Calculate the BER of BPSK modulated signal.

**Theory**

In any phase modulation scheme the information is expressed in terms of phase of the carrier. Phase of the carrier signal is shifted according to the input binary data. BPSK (Binary Phase Shift Keying) is the simplest form of phase shift keying (PSK). In BPSK, individual data bits are used to control the phase of the carrier. During each bit interval, the modulator shifts the carrier to one of two possible phases, which are 180 degrees or π radians apart. Each bit of the digital signal produces a transmit symbol with duration Ts, which is equal to the bit duration Tb.

Bit error rate (BER) of a communication system is defined as the ratio of number of error bits and total number of bits transmitted during a specific period. It is the likelihood that a single error bit will occur within received bits, independent of rate of transmission.

**Software used**

Octave

**Code**

x = randi(2, 1, 1000);

x(x == 2) = -1;

snr =[-40:4:40];

ber = zeros(1, length(snr));

for i = 1:length(snr)

y = awgn(x, snr(i));

for j = 1:length(x)

if x(j) > 0 && y(j) <= 0

ber(i) = ber(i) + 1;

endif

if x(j) <= 0 && y(j) > 0

ber(i) = ber(i) + 1;

endif

endfor

endfor

disp(ber);

figure;

plot(snr, ber);

xlabel('SNR');

ylabel('Bit error rate');

title('BER Simulation for BPSK');

**Output**



**Result**

The BER of BPSK modulated signal has been calculated.

**Practical - 6**

**Aim**

Given is a linear block code with the generator matrix G :-

1 1 0 0 1 0 1

G = 0 1 1 1 1 0 0

1 1 1 0 0 1 1

1. Calculate the number of valid code words N and the code rate RC. Specify the complete Code set C.
2. Determine the generator matrix G′ of the appropriate systematic (separable) code C’.
3. Determine the syndrome table for single error.

**Theory**

In [coding theory](https://en.wikipedia.org/wiki/Coding_theory), a generator matrix is a [matrix](https://en.wikipedia.org/wiki/Matrix_(mathematics)) whose rows form a [basis](https://en.wikipedia.org/wiki/Basis_(linear_algebra)) for a [linear code](https://en.wikipedia.org/wiki/Linear_code). The codewords are all of the [linear combinations](https://en.wikipedia.org/wiki/Linear_combination) of the rows of this matrix, that is, the linear code is the [row space](https://en.wikipedia.org/wiki/Row_space) of its generator matrix.

**Software used**

Octave

**Code**

**a)**

% Generator Matrix

G = [

1 1 0 0 1 0 1;

0 1 1 1 1 0 0;

1 1 1 0 0 1 1;

];

N = 7; % total bits

K = 3; % no of message bits

fprintf("Code rate is %d\n", K/N);

possibleCodes = [

0 0 0;

0 0 1;

0 1 0;

0 1 1;

1 0 0;

1 0 1;

1 1 0;

1 1 1;

];

codeSet = [];

for i=1:8

code = possibleCodes(i,:);

codeSet = [codeSet;mod(code\*G,2)];

endfor

fprintf("\nCode set : \n");

disp(codeSet)

**b)**

G = [ 1 1 0 0 1 0 1;

0 1 1 1 1 0 0;

1 1 1 0 0 1 1;

];

K=3;

N=7;

% Generating appropriate systematic code

G(3,:) = mod(G(3,:) + G(1,:),2);

G(2,:) = mod(G(3,:) + G(2,:),2);

G(1,:) = mod(G(2,:) + G(1,:),2);

fprintf("Generator matrix :\n");

disp(G)

possibleCodes = [

0 0 0;

0 0 1;

0 1 0;

0 1 1;

1 0 0;

1 0 1;

1 1 0;

1 1 1;

];

codeSet = [];

for i=1:8

code = possibleCodes(i,:);

codeSet = [codeSet;mod(code\*G,2)];

endfor

fprintf("\nCode set :\n");

disp(codeSet)

**c)**

% Generator Matrix

G = [

1 1 0 0 1 0 1;

0 1 1 1 1 0 0;

1 1 1 0 0 1 1;

];

N = 7;

K = 3;

H = G';

r = [0 0 0 1 0 1 0];

p = mod(r\*H,2);

num=0;

pos=-1;

p

for i=1:7

if isequal(p,H(i,:))

pos=i;

break;

endif

endfor

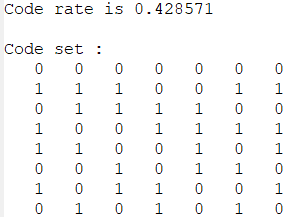
fprintf("Error in %d\n",pos);

r(pos) = mod(r(pos)+1,2);

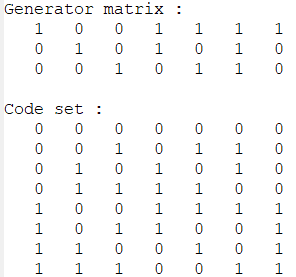
disp(r)

**Output**

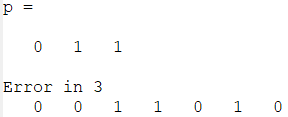
**a)**



**b)**



**c)**



**Result**

All the operations have been performed and the calculated values are same as expected values.