Name: Aravind Reddy V
Roll: IMT 2015 524
Email: aravind.reddy@iiitb.org

Midterm Lab

All codes are written for Octave

Q1.
Objective: tent signal x(t) = (1 - |t|) I[-2, 2]

Plot:
- x(3t - 3/2)
- X(f)
- x(t) * v(t)

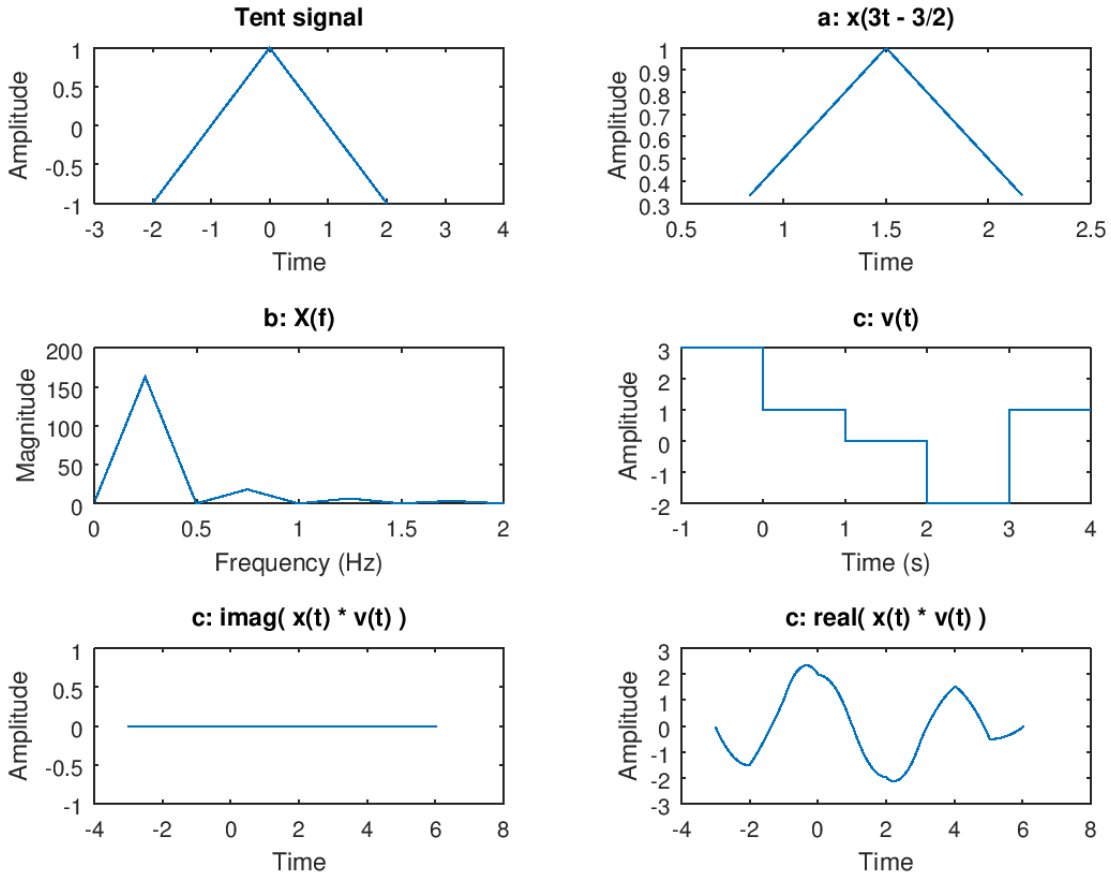
Code:

```
% Octave

rows = 3;
cols = 2;

dt = 0.01;


t = -2:dt:2;
x = 1 - abs(t);

subplot(rows, cols, 1);
plot(t, x)
xlabel("Time");
ylabel("Amplitude");
title("Tent signal");
xlim([-3, 4]);
% ylim([0, 2]);

%%%%%%%%%%%%%%%%%%

% a

ta = -2/3:dt:2/3;
xa = 1 - abs(ta);
ta = ta.+3/2;

subplot(rows, cols, 2);
plot(ta, xa);
xlabel("Time");
ylabel("Amplitude");
title("a: x(3t - 3/2)");
xlim([0.5, 2.5]);
% ylim([0, 2]);
```

```
%%%%%%%%%%%%%%%%%%%%

% b

Fs = 1/dt;

tc = -2:dt:2;
xc = 1 - abs(t);

x = fft(xc);
N = length(xc);

freqHz = (0:1:(length(x) - 1)).*Fs/N;

Frequencies = freqHz(1:length(freqHz)/2);
Amplitudes = abs(x)(1:length(x)/2);

subplot(rows, cols, 3);
plot(Frequencies, Amplitudes);
xlabel("Frequency (Hz)");
ylabel("Magnitude");
title("b: X(f)");
xlim([0, 2]);

%%%%%%%%%%%%%%%%%%%%

% c

t_first = -1:dt:0;
x_first = ones(length(t_first), 1).*3;

t_second = 0:dt:1;
x_second = ones(length(t_second), 1).*1;

t_third = 1:dt:2;
x_third = ones(length(t_third), 1).*0;

t_fourth = 2:dt:3;
x_fourth = ones(length(t_fourth), 1).*-2;

t_fifth = 3:dt:4;
x_fifth = ones(length(t_fifth), 1).*1;

tc = vertcat(t_first', t_second', t_third', t_fourth', t_fifth');
xc = vertcat(x_first, x_second, x_third, x_fourth, x_fifth);

subplot(rows, cols, 4);
plot(tc, xc);
xlabel("Time (s)");
ylabel("Amplitude");
title("c: v(t)");
```

```
% The original signal
t = -2:dt:2;
x = 1 - abs(t);

[T, X] = contconv(x, xc, t(1), tc(1), dt);

subplot(rows, cols, 5);
plot(T, imag(X));
xlabel("Time");
ylabel("Amplitude");
title("c: imag( x(t) * v(t) )");

subplot(rows, cols, 6);
plot(T, real(X));
xlabel("Time");
ylabel("Amplitude");
title("c: real( x(t) * v(t) )");


print -dpng 1.png
```

Plot:



Q.2

Objective:

Message signal m(t) has frequency spectrum M(f) = 2tent(-1, 1), plot frequency spectrum of DSB signal uDSB = m(t)cos(2pi*fc*t), fc = 25

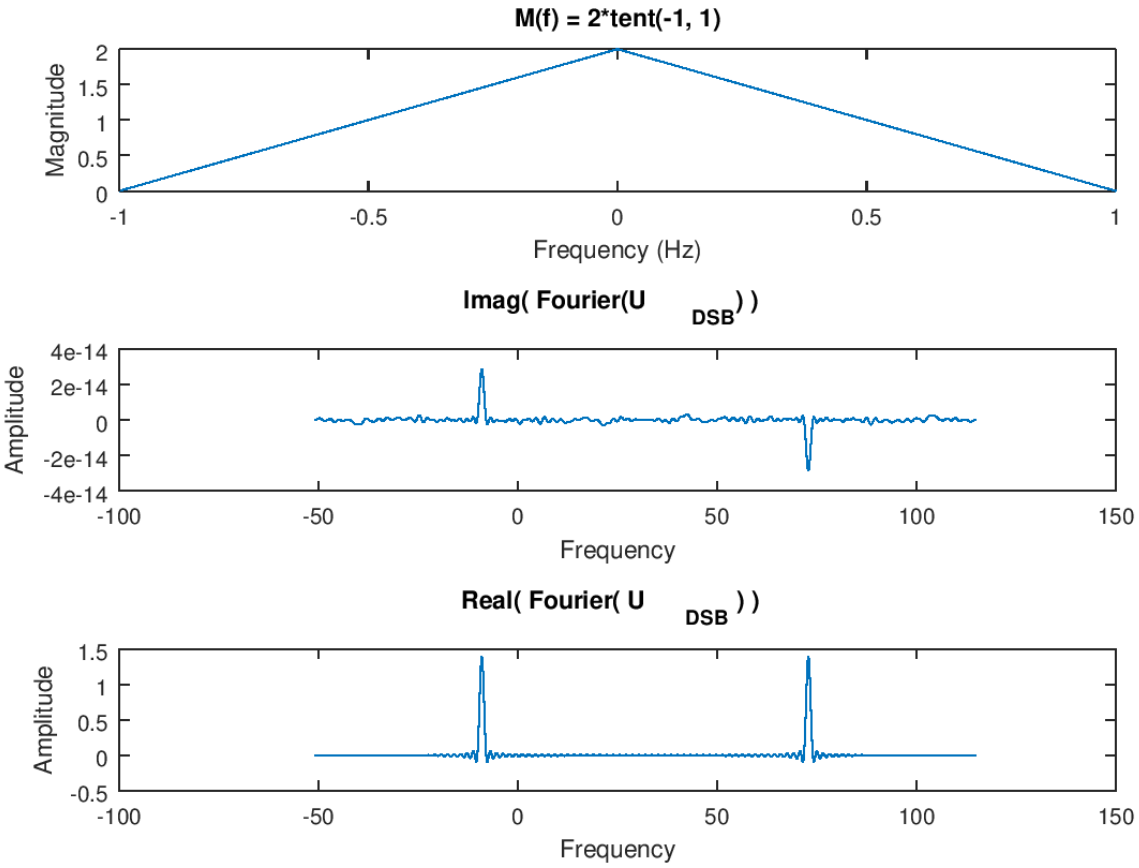
Approach: Fourier transform of uDSB = UDSB
UDSB = M(f) * F(cos(s*pi*fc*t))

Code:

```
% Octave

% Note: contFT and contconv omitted

% u_DSB = m(t)cos(2pi*fc*t)
% U_DSB = M(f) * F(cos(2pi*fc*t));

rows = 3;
cols = 1;

fc = 25;

dt = 0.01;

df = 0.01;
df_desired = 1;

% The tent freq plot
```

```
Freq = -1:df:1;
M = (1 - abs(Freq)).*2;


subplot(rows, cols, 1);

plot(Freq, M);
title("M(f) = 2*tent(-1, 1)");
xlabel("Frequency (Hz)");
ylabel("Magnitude");

%%%%%%%

subplot(rows, cols, 2);
x_M = ifft(M);
t_M = 0:0.01:length(x_M)*dt - dt;

t_cos = -1*length(x_M)*dt/2:dt:length(x_M)*dt/2;
% t_cos = -1:dt:1;
x_cos = cos(2*pi*fc*t_cos);

% plot(t_cos, x_cos);

[X_cos, F_cos, df_cos] = contFT(x_cos, t_cos(1), dt, df,
df_desired);

% plot(F_cos, X_cos);

[T_U, X_U] = contconv(M, X_cos, Freq(1), F_cos(1), dt);


plot(T_U, imag(X_U));
title("Imag( Fourier(U_D_S_B) )");
xlabel("Frequency");
ylabel("Amplitude")


%%

subplot(rows, cols, 3);

plot(T_U, real(X_U));
title("Real( Fourier( U_D_S_B ) )");
xlabel("Frequency");
ylabel("Amplitude");


print -dpng 2.png;
```

Plot:


M(f) = 2*tent(-1, 1)
Magnitude
Frequency (Hz)
Imag(Fourier(U DSB))
Amplitude
Frequency
Real(Fourier(U DSB))
Amplitude
Frequency

Q.4

Objective:

Bitstream b[n] = [100101]

a. Perform binary modulation using sinc(4(t-0.5)) I[0, 1]
b. Generate AM wave with modulation index = 0.5
c. Demodulate the signal

Code:

```
message = [1, 0, 0, 1, 0, 1];
dt = 0.01;

% message = [1, -1, -1, 1, -1, 1];

rows = 3;
cols = 1;

%% getRef: generates reference sinc function
function [time, signal] = getRef(bit, time_offset, dt)
     time = time_offset:dt:(time_offset + 1 - dt);
     signal = sinc(time.-0.5.*4).*bit;
end

%% getMessage: generate message signa;
function [time, signal] = getMessage(message, dt)
     count = 0;
     time = [];
     signal =[];

     for t = message
          [t_time, t_signal]= getRef(t, count, dt);
          time = [time, t_time];
          signal = [signal, t_signal];
          count = count + 1;
     end
end

[t_m, x_m] = getMessage(message, dt);

size(t_m)
size(x_m)

subplot(rows, cols, 1);

plot(t_m, x_m);
title("a: Binary modulation");
xlabel("Time (s)");
ylabel("Amplitude");
```

```
%%%%%%%%%%%%%%%

amp = 0.4;
fc = 10;

%% getIcarrier: get cos
function [time, signal] = getIcarrier(amp, fc, length, dt)
     time = 0:dt:length - dt;
     signal = cos(2*pi*fc*time).*amp;
end

function [time, signal] = getQcarrier(amp, fc, length, dt)
     time = 0:dt:length - dt;
     signal = sin(2*pi*fc*time).*amp;
end

[t_up, x_up] = getIcarrier(amp, fc, length(message), dt);

x_up = x_up.*x_m;


subplot(rows, cols, 2);

plot(t_up, x_up);

title("b: AM wave");
xlabel("time (s)");
ylabel("Amplitude");



%%%%%%%%%%%%%%%

%% diodeFilter: makes all values less than zero 0
function [result] = diodeFilter(vector)
     vector(vector < 0) = 0;
     result = vector;
end


%% RCFilter:
function [time_f, signal_f] = RCfilter(time, signal, RC)
     % t_response = 0:ns/length(time):ns;

     % 1/fc < RC < 1/b
     % b = 1.5 KHz

     t_response = time;

     dt = 1/40;
```

```
     u_response = ones(length(signal), 1);


     temp_t = t_response./RC;
     temp_t = temp_t.*-1;

     temp_t_exp = arrayfun( @(x) exp(x), temp_t);

     u_response = u_response.*temp_t_exp;

     u_response = u_response(1,:);

     % size(t_response)
     % size(u_response)

     [time_f, signal_f]  = contconv(signal, u_response, time(1),
t_response(1), dt);
     time_f = time_f.*2./5;
end

%% DBblock: function description
function [time_dcblock, signal_dcblock] = DCblock(time, signal)

     meanSignal = mean(signal)

     time_dcblock = time;
     signal_dcblock = signal.-meanSignal;

end



T_in = t_up;
X_in = x_up;

X_in = diodeFilter(X_in); % diode filter

RC = 0.6;
% RC = 5;

[T_out, X_out] = RCfilter(T_in, X_in, RC);

% [T_out, X_out] = DCblock(T_out, X_out);


subplot(rows, cols, 3);
plot(T_out, X_out);
title("c: demodulated");
xlabel("time (s)");
ylabel("Amplitude");
xlim([0, 6]);
```

```
print -dpng 4.png
```

Plot: