GEBZE TECHNICAL UNIVERSITY

ELEC361 ANALOG COMMUNICATION SYSTEMS

PROJECT 2 HOMEWORK

AHMET ALİ TİLKİCİOĞLU
210102002163

MATLAB SIMULATION:

The handling code was added before the signals given by the project were written. The m(t) function is defined with the rectpuls function. The fourier transform of the message sign was taken with MATLAB code and plotted.The message sign and its spectrum are drawn in figures 1 and 2.

```
% Ahmet Ali Tilkicioğlu / ELEC361 Project - 2 / FM Project / 210102002163

clc
clear all

t0 = 4;                                   % 5 second variable
ts = 0.0001;                              % sample time variable (second)
t = 0:ts:t0-ts;                           % 0 to 5 sec. sampled time
fs = 1/ts;                                % sample frequency

w = 2;                                    % rectangular signal tao value
m = rectpuls(t-1,w) - rectpuls(t-3,w);    % m(t) signal

M = fft(m);                               % m(t) amplitude spectrum
shifted_M = abs(fftshift(M)/length(M));   % M(f) shift frequency val.

f = fs/length(M)*(-length(M)/2:length(M)/2-1); % sampled all frequency
values


% PLOT m(t), M(f)
figure(1)

subplot(2,1,1)
plot(t,m(1:length(t)))                    % plotting m(t)
xlabel('time (s)')
ylabel('m(t)')
title('m signal')

subplot(2,1,2)
plot(f,shifted_M)                         % plotting M(f)
xlabel('frequency (hz)')
ylabel('M(f)')
title('M spectral')
```

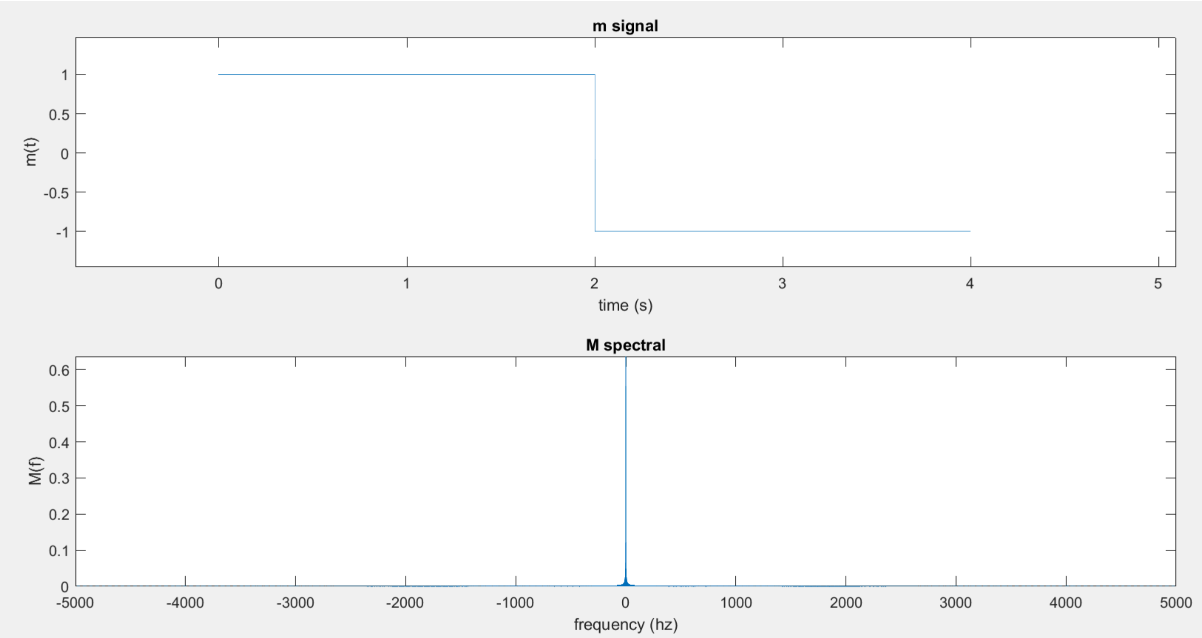

Figure 1: m(t) Signal and Magnitude Spectrum

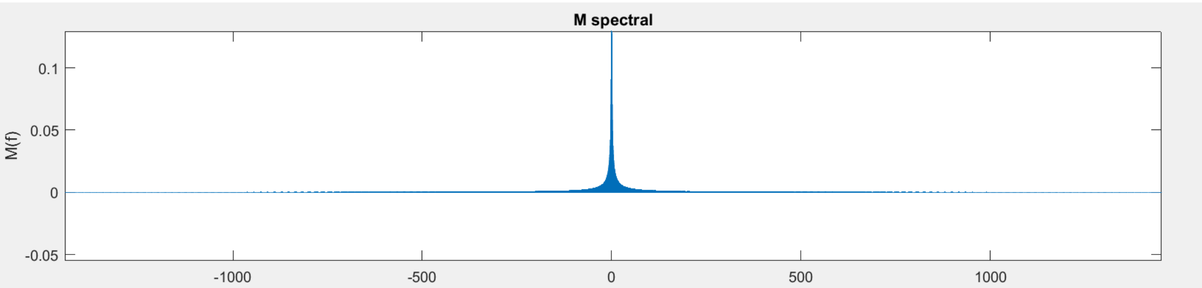

Figure 2: m(t) Magnitude Spectrum Zoom Out (sinc signal)

The message signal was integrated to find the phase sign. The desired phase deviation constant is defined in the homework. Phase signal and spectrum are in figure 3.

```
kf = 50;                                    % frequency deviation constant
o = 2.*pi.*kf.*ts.*cumtrapz(m);             % o(t) phase signal
O = fft(o);                                 % o(t) amplitude spectrum
shifted_O = abs(fftshift(O)/length(O));     % O(f) shift frequency val.

% PLOT o(t), O(f)
figure(2)
subplot(2,1,1)
plot(t,o(1:length(t)))                      % plotting o(t)
xlabel('time (s)')
ylabel('o(t)')
title('o signal')

subplot(2,1,2)
plot(f,shifted_O)                           % plotting O(f)
xlabel('frequency (hz)')
ylabel('O(f)')
title('O spectral')
```

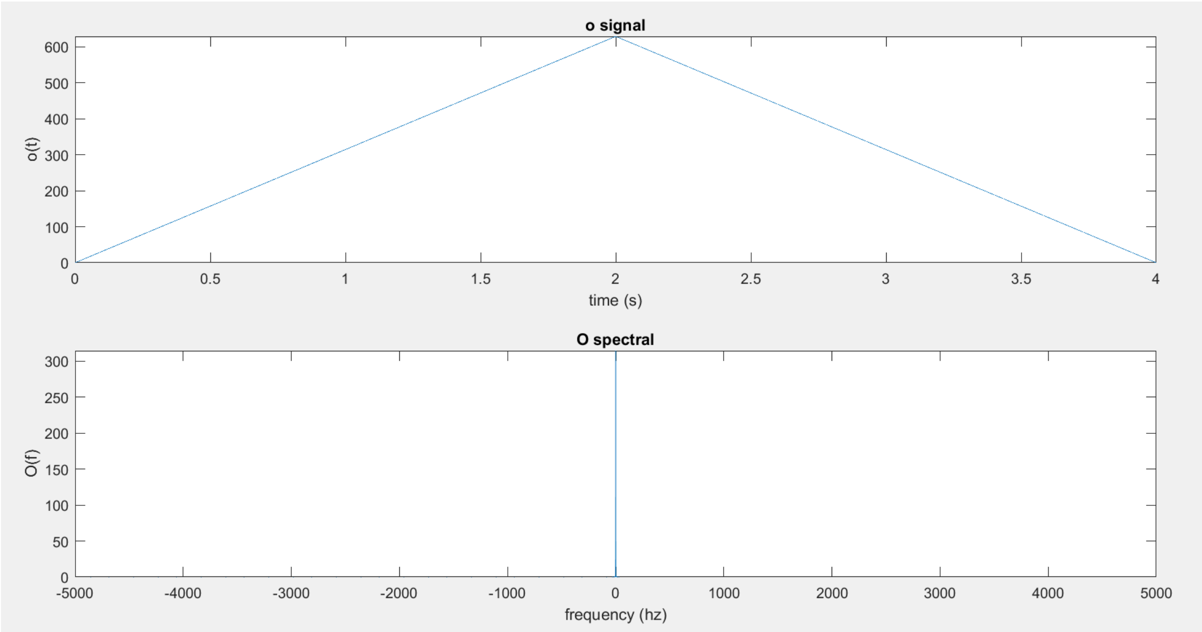

Figure 3: o(t) phase signal and magnitude spectrum

It is defined within the y(t) function with the phase signal for which we defined the frequency modulation. In Figure 4, y(t) signal and spectrum were obtained**.**

```
y = 5*cos(500*pi.*t + o);                        % y(t) signal
Y = fft(y);                                      % y(t) amplitude spectrum
shifted_Y = abs(fftshift(Y)/length(Y));          % Y(f) shift real frequency
value

% PLOT y(t), Y(f)
figure(3)

subplot(2,1,1)
plot(t,y(1:length(t)))                       % plotting y(t)
xlabel('time (s)')
ylabel('y(t)')
title('y signal')

subplot(2,1,2)
plot(f,shifted_Y)                            % plotting Y(f)
xlabel('frequency (hz)')
ylabel('Y(f)')
title('Y spectral')
```

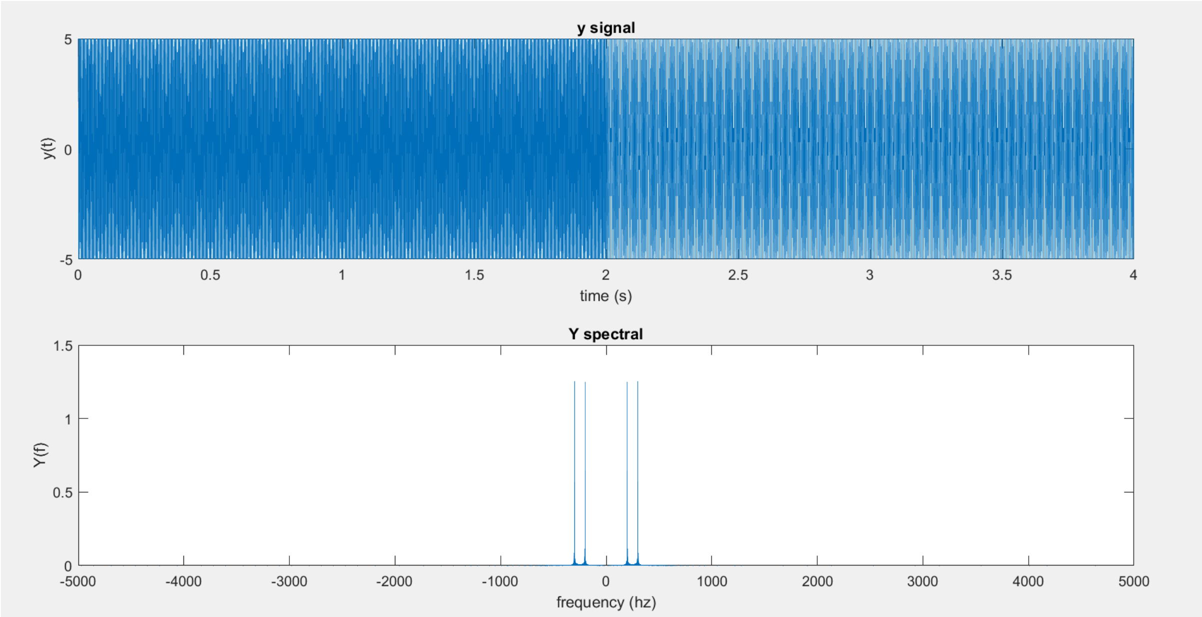

Figure 4: y(t) Signal and Magnitude Spectrum

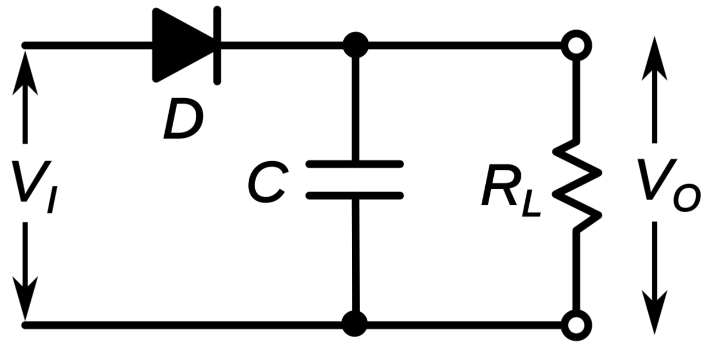

Figure 5: Envelope Detector Circuit

In order to demodulate the y(t) function we obtained, a differential was first taken. The circuit in figure 5 is simulated as MATLAB code. The circuit works as an envelope detector. In the circuit, the diode is used to receive the upper half wave. In MATLAB, the abs function was used to get the upper half wave. Then, the region of the message signal with higher power density was taken with a low-pass filter. Since the value taken at the output is in the frequency domain, it is taken as inverse Fourier. Since there is a DC value at the output of the low-pass filter, this DC value is blocked with the ideal capacitor. For this process, the mean function was used and the demodulated signal was obtained. The described process and code output are shown in figure6-10, respectively.

```
% Demodulator Diagram
%                    z(t)          u(t)        l(t)
% y(t) ---> |d/dt| ---> |Diode| ---> |LPF| ---> |DC Blocking| ---> d_m(t)
%

z = gradient(y,t);                                   % y(t) differantial
```

```
u = abs(z);                                         % diode output (half wave)

Z = fft(z);                                         % z(t) amplitude spectrum
U = fft(u);                                         % u(t) amplitude spectrum

df = 0.25;                                          % sampled frequency values
LPF_BW = 200;                                       % LPF bandwidth
Max_freq = length(m);                               % Vector list lenght
% Low Pass Filter
L = zeros(1,length(m));
L(1:LPF_BW/df+1) = U(1:LPF_BW/df+1);
L((Max_freq-LPF_BW/df)+1:Max_freq) = U((Max_freq-LPF_BW/df)+1:Max_freq);

l = ifft(L);             % inverse fourier transform

d_m = l - mean(l);       % DC bloking
D_M = fft(d_m);          % D_M(f) amplitude spectrum (Demodulated Signal)

shifted_Z = abs(fftshift(Z)/length(Z));         % Z(f) shift real frequency
value
shifted_U = abs(fftshift(U)/length(U));         % U(f) shift real frequency
value
shifted_L = abs(fftshift(L)/length(L));         % L(f) shift real frequency
value
shifted_D_M = abs(fftshift(D_M)/length(D_M));   % D_M(f) shift real
frequency value

% PLOT z(t), Z(f)
figure(4)

subplot(2,1,1)
plot(t,z)                                       % plotting z(t)
xlabel('time (s)')
ylabel('z(t)')
title('z signal')

subplot(2,1,2)
plot(f,shifted_Z)                               % plotting Z(f)
xlabel('frequency (hz)')
ylabel('Z(f)')
title('Z spectral')

% PLOT u(t), U(f)
figure(5)

subplot(2,1,1)
plot(t,u(1:length(t)))                          % plotting z(t)
xlabel('time (s)')
ylabel('u(t)')
title('u signal')

subplot(2,1,2)
plot(f,shifted_U)                               % plotting Y(f)
xlabel('frequency (hz)')
ylabel('U(f)')
title('U spectral')

% PLOT l(t), L(f)
figure(6)
```

```
subplot(2,1,1)
plot(t,l(1:length(t)))                              % plotting z(t)
xlabel('time (s)')
ylabel('l(t)')
title('l signal')

subplot(2,1,2)
plot(f,shifted_L)                                   % plotting Y(f)
xlabel('frequency (hz)')
ylabel('L(f)')
title('L spectral')

% PLOT d_m(t), D_M(f)
figure(7)
subplot(2,1,1)
plot(t,d_m(1:length(t)))                              % plotting z(t)
xlabel('time (s)')
ylabel('d_m(t)')
title('d_m signal')

subplot(2,1,2)
plot(f,shifted_D_M)                                   % plotting Y(f)
xlabel('frequency (hz)')
ylabel('D_M(f)')
title('D_M spectral')

% PLOT comparison m(t) D_M(t)
figure(8)
subplot(2,1,1)
plot(t,m(1:length(t)))                              % plotting m(t)
xlabel('time (s)')
ylabel('m(t)')
title('m signal')

subplot(2,1,2)
plot(t,d_m(1:length(t)))                            % plotting z(t)
xlabel('time (s)')
ylabel('d_m(t)')
title('d_m signal')
```

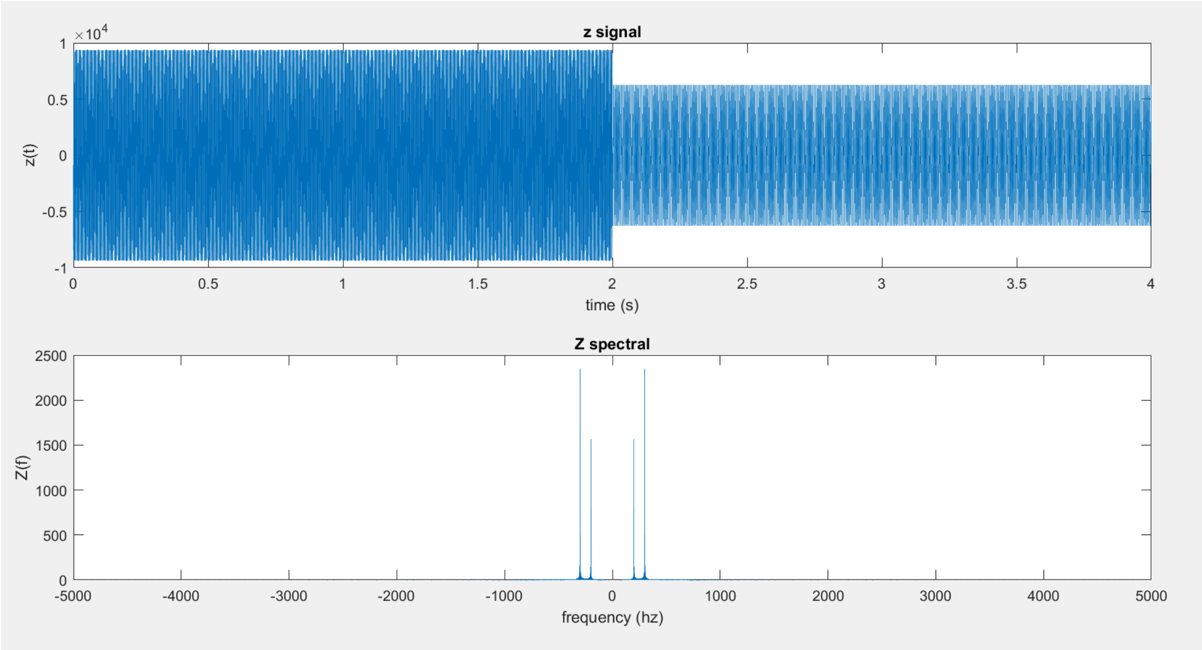

Figure 6: z(t) Differentiator Output and Spectrum

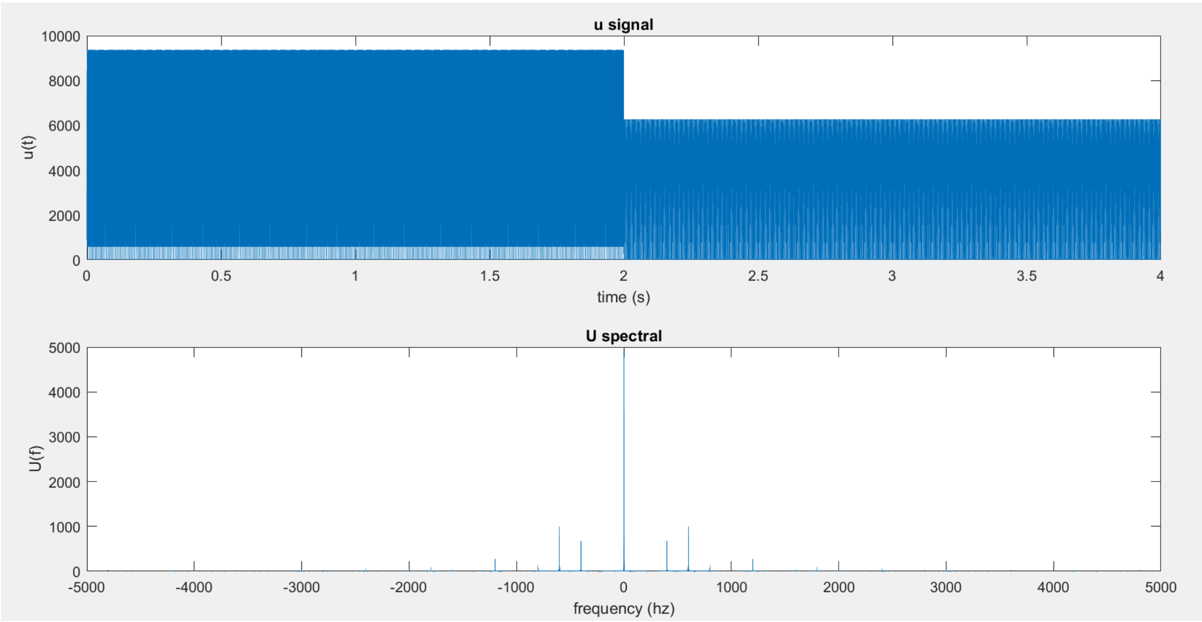

Figure 7: u(t) Diode Output (Half wave) and Spectrum

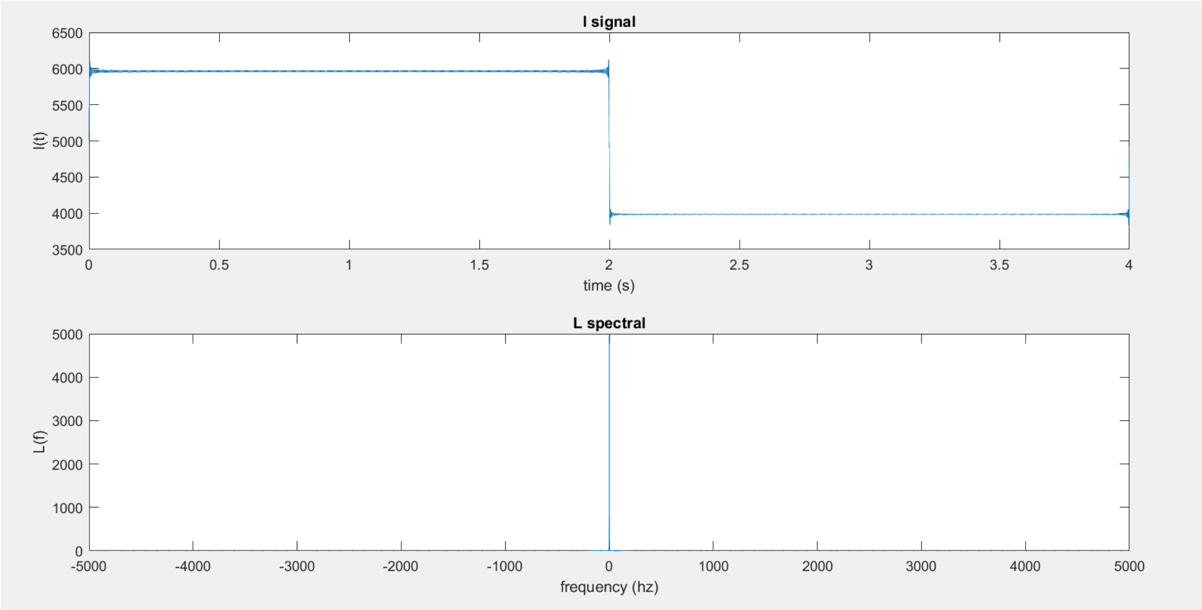

Figure 8: l(t) Low Pass Filter Output and Spectrum

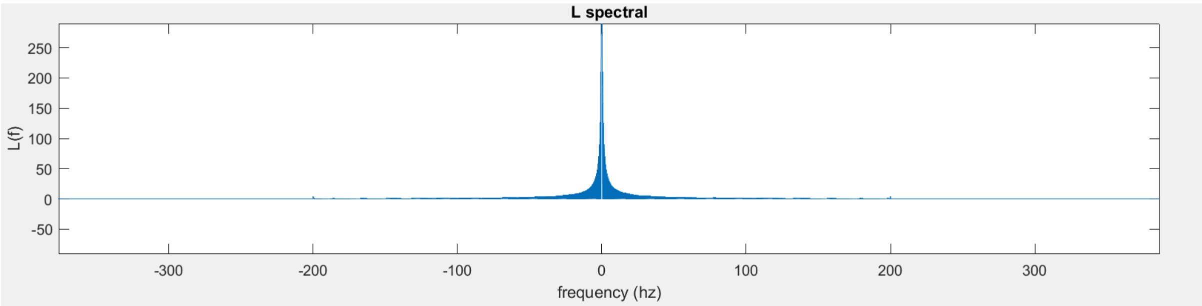

Figure 9: Low Pass Filter Spectrum Zoom Out

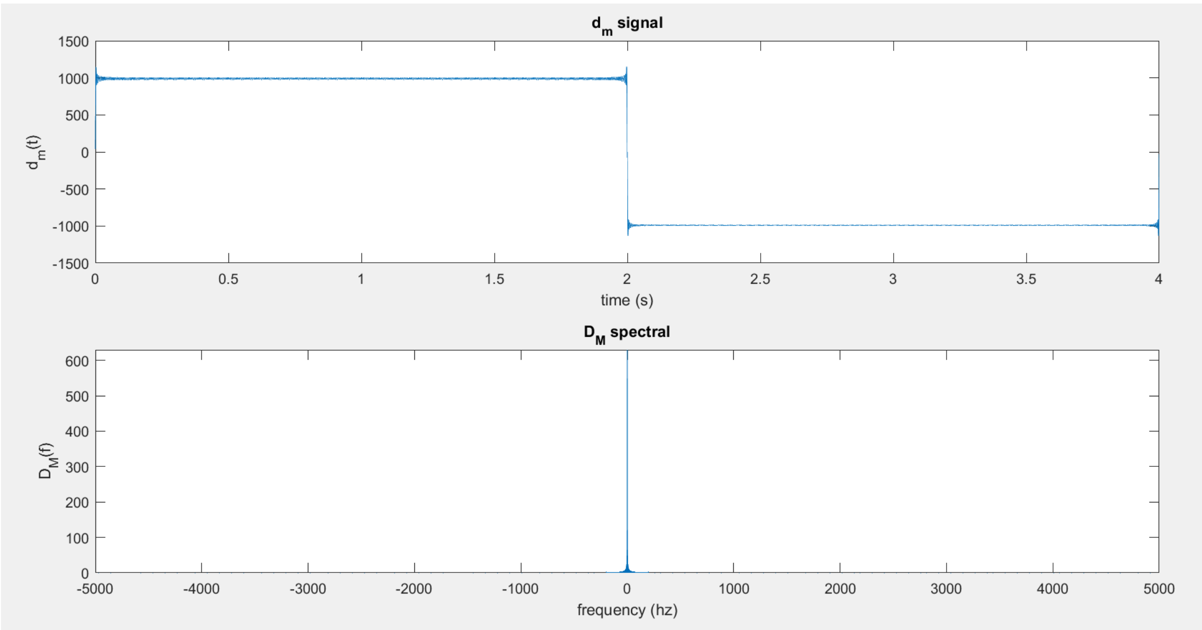

Figure 10: d_m(t) Demodulated Message Signal and Spectrum

A comparison of message sign and demodulated message sign is made in Figure 11. (Since the demodule sign is larger, it cannot be shown on the same table.)

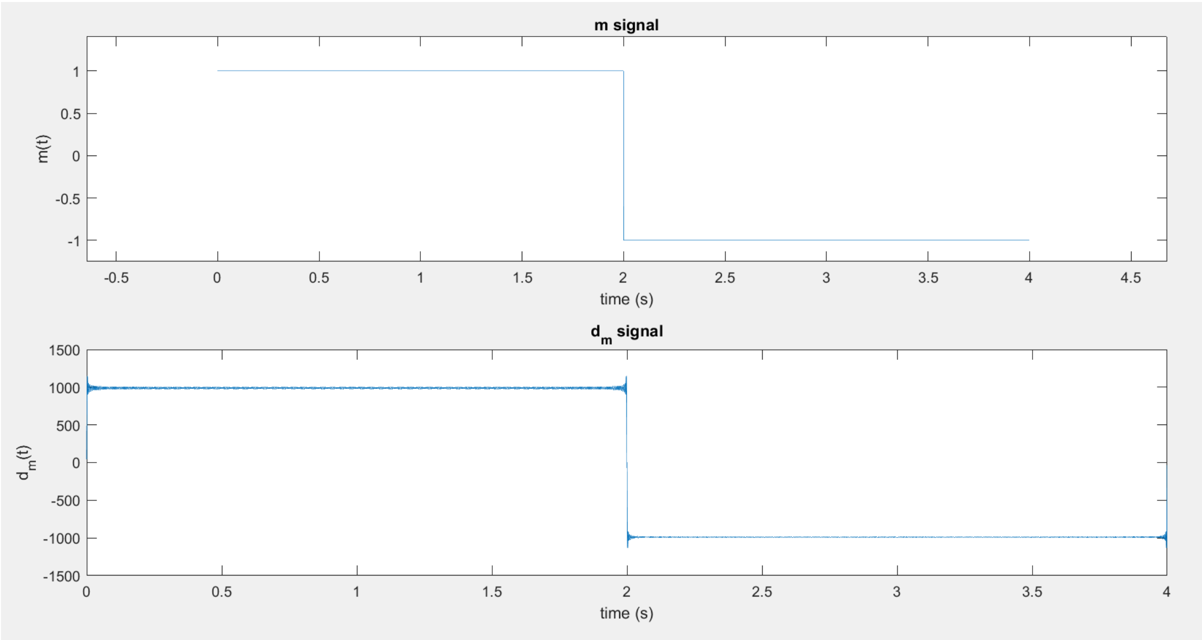

Figure 11: Comparison Table Demodulated Signal and Message Signal