PROJECT MATLAB

Communication systems
Submitted to Dr. Fatma Newagy

Moataz Mokhtar Said Abdelghani
1701460@eng.asu.edu.eg

Ahmed Ayman Ahmed Hassan
1700037@eng.asu.edu.eg

JUNE 10, 2021

FACULTY OF ENGINEERING AIN SHAMS UNIVERSITY
CAIRO EGYPT

Part 1:

a) *For the input signal:*

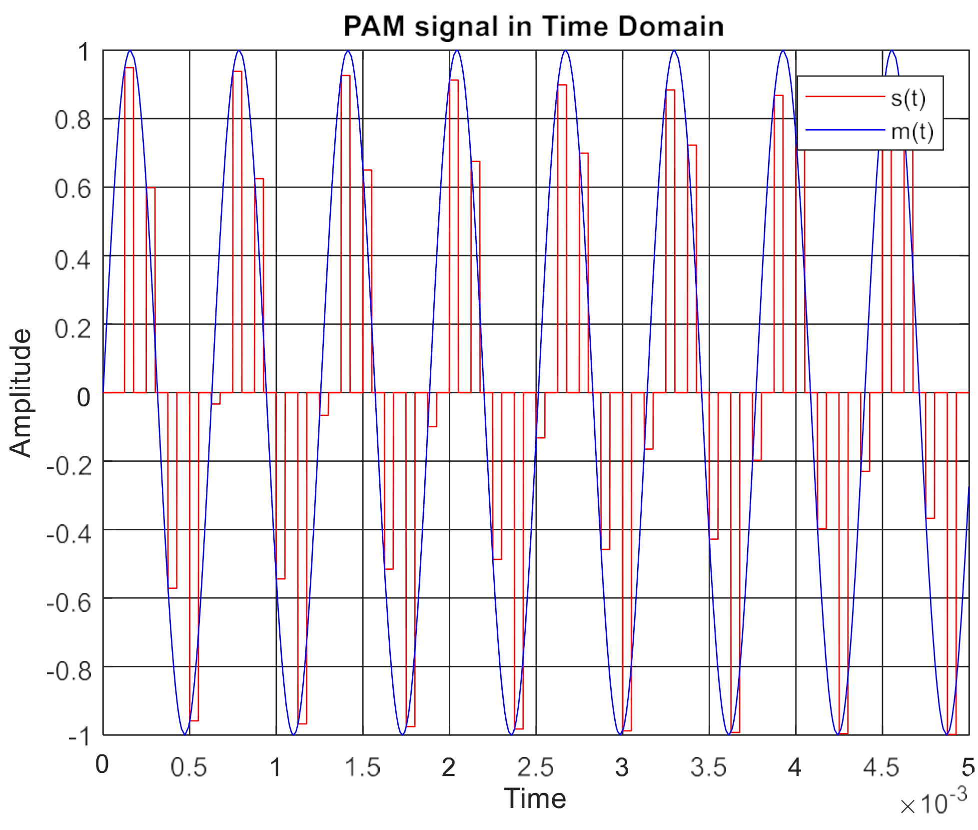
$$\mathbf{m(t) = sin(2\pi f_o t)}$$

- $\mathbf{f \ = \frac{10^4}{2\pi}\ Hz}$
- $Sampling\ frequency\ fs = 8\ kHz$
- $Sampling\ interval\ t_s\ = \frac{1}{8k} = 0.25\ msec$
- $T_{Pulse} = 50\mu sec$
- $Sample\ and\ hold\ to\ obtain\ flat\ topped\ PAM\ signal\ s(t).$
- $n = number\ of\ samples.$

The PAM signal:

$$\mathrm{s(t)} = \sum_{-\infty}^{\infty} \mathrm{m(n}t_s) * h(t - nts)$$

*Plotting the signal **in time domain**:*



b)

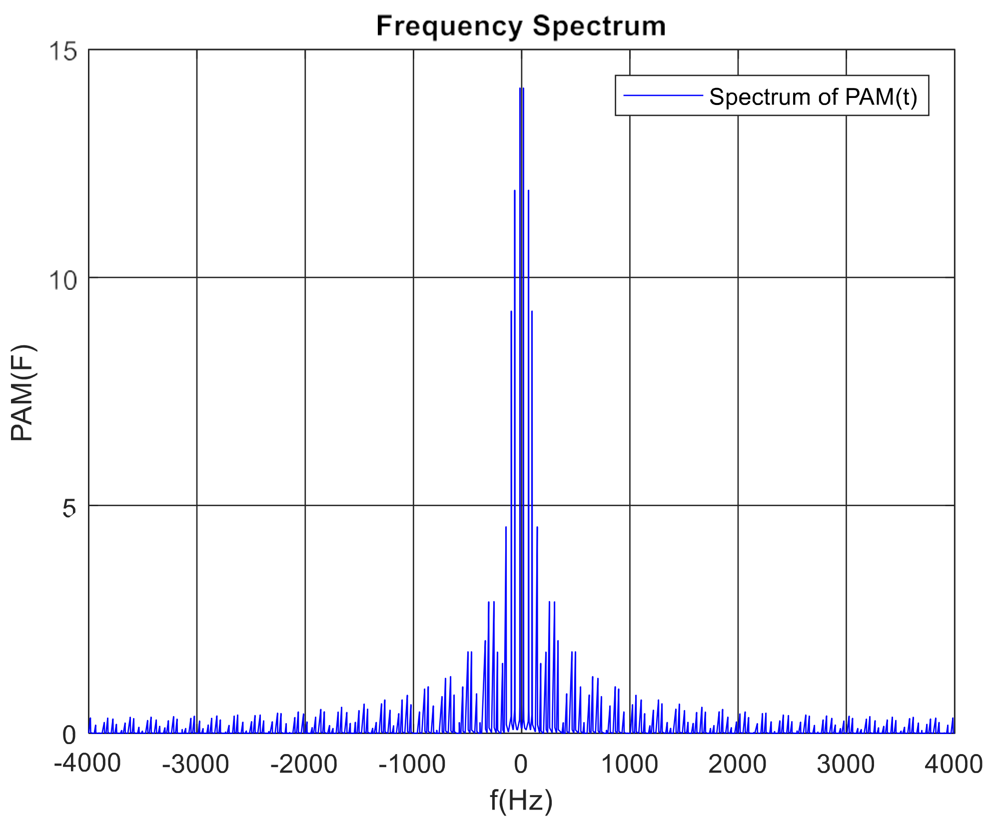
$$s(t) = \sum_{-\infty}^{\infty} m(nt_s) * \mathrm{h(t-nts)} = m(t) * h\ (t)$$

$in\ the\ frequency\ domain\ the\ two\ signal\ the\ input\ and\ the\ train\ of\ pulses\ will\ be\ multiplicated$

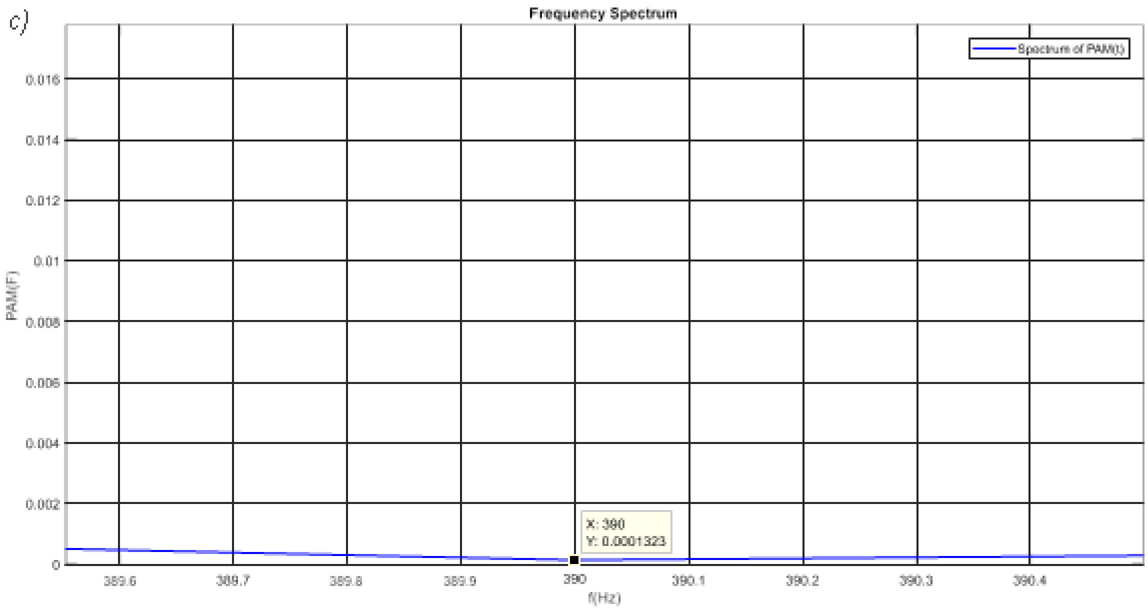
$as\ they\ are\ convulted\ in\ the\ time\ domain$

$$S(F) = M(F)\ x\ H\ (F)$$

*Plotting the signal in **frequency domain:***



c)

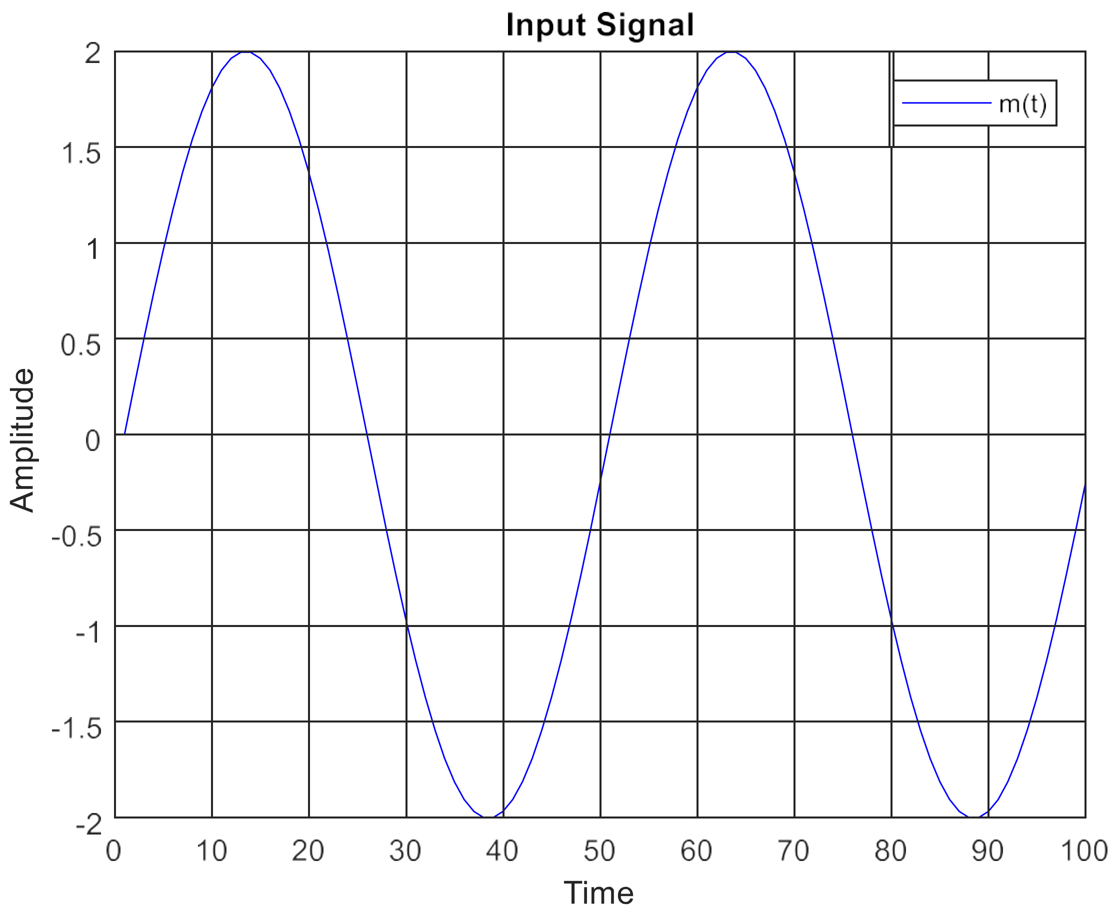


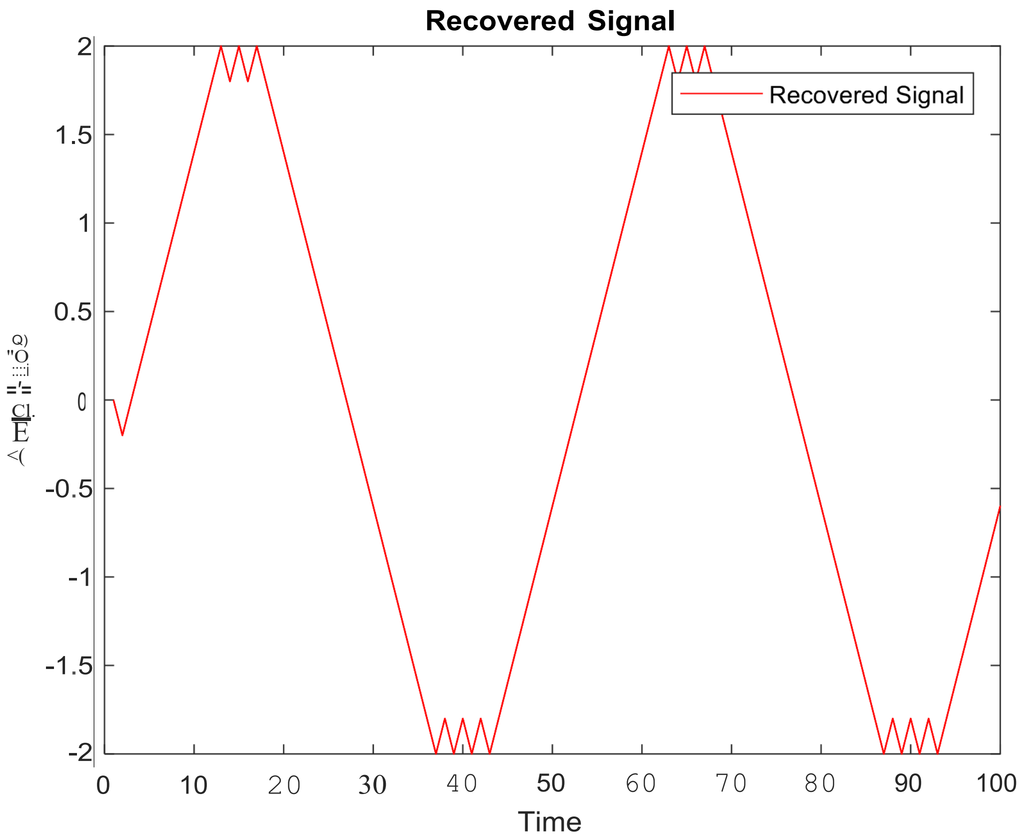
Comment: as shown from previous figure that $|s(f)|$ goes first to zero at frequency $f_o = 400Hz$, but the actual value of this frequency from hand analysis supposed to be $f_o = 2000Hz$, so there is a difference between MATLAB results and calculated results which we haven't settled.

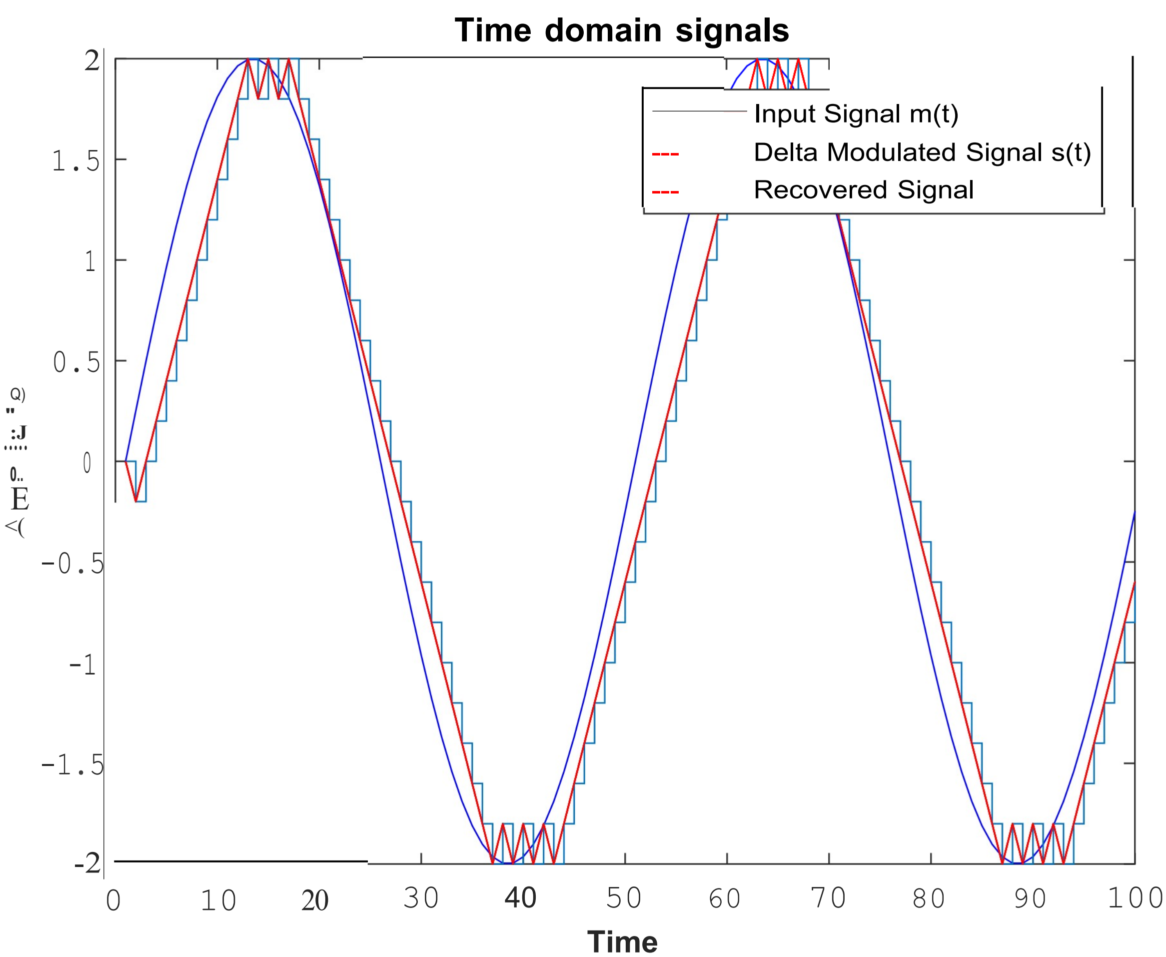
Part 2 Delta Modulation DM:

- $m(t) = 2\,sin(2\pi t)$
- $max\,|\frac{dm(t)}{dt}| = 4\pi$
- $step\ size\ \Delta = 0.2$
- $sampling\ frequency\ f_s = 50Hz$
- $so\ \frac{\Delta}{T_s} = \Delta * f_s = 10$

From previous analysis we notice that $max\,|\frac{dm(t)}{dt}| > \frac{\Delta}{T_s}$, *therefore the recovered signal will suffer from slope overload distortion effect which we will ensure using MATLAB.*




Recovered Signal
Recovered Signal
Time


Time domain signals
Input Signal m(t)
Delta Modulated Signal s(t)
Recovered Signal
Time

Comment: from the previous MATLAB graphs it's shown that the recovered signal suffers from slope overload distortion as mentioned before, and we see also the effect of granular noise which looks like kind quantization error in PCM.

***Bitstream** of the first cycle from MATLAB*

	1	2	3	4	5	6
1	0	1	1	1	1	1
	7	8	9	10	11	12
1	1	1	1	1	1	1
	13	14	15	16	17	18
1	0	1	0	1	0	0
	19	20	21	22	23	24
1	0	0	0	0	0	0
	25	26	27	28	29	30
1	0	0	0	0	0	0
	31	32	33	34	35	36
1	0	0	0	0	0	0
	37	38	39	40	41	42
1	1	0	1	0	1	0
	43	44	45	46	47	48
1	1	1	1	1	1	1
	49	50	51	52	53	54
1	1	1	1	1	1	1

MATLAB Code

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%PART 1%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fo=(1e4)/(2*pi);    %freq of the signal
fs=8e3;             %sampling rate
ts=1/fs;            %sampling interval
T=50e-6;            %Pulse width interval
RFACT = 25e-4;      %Range factor to control the range

t = 0:1/(100*fs):(2*RFACT-(1/(100*fs)));

mt=sin(2*pi*fo*t);                 %input signal
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%
%                              PAM SIGNAL IN TIME DOMAIN
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%
numberofpulses = 2*RFACT*fs;       %number of pulses
duty = 40;                         %duty cycle of the train of the pulses

v = length(t);   % vector of time to manipulate the error of the matrix 
dimensions

samples = v/numberofpulses;              %number of samples
%number samples = 5e-4/ts;
pulseindex = [1:samples:v];
PWM_HIGH =  ceil(samples * duty/100);    % ceil the values of the falt topp 
rectangular pulse
PAMsignal = zeros(1,v);
for i = 1 : length(pulseindex)           %for loop to draw the PAM signal
    PAMsignal(pulseindex(i):pulseindex(i) + PWM_HIGH) = mt(pulseindex(i));
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%
%                            PLOTTING PAM SIGNAL IN TIME DOMAIN
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%
figure(1);
plot(t,PAMsignal,'r')
hold on;
plot(t,mt,'b');
hold on;
ylabel('Amplitude')
xlabel('Time')
title('PAM signal in Time Domain')
legend('s(t)','m(t)')
grid
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%
%                            PLOTTING PAM SIGNAL IN FREQEUNCY DOMAIN
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%

%fourier transform and frequency domain
N=length(PAMsignal);  %define Vector of the same Length of N
```

```
%To plot the frequency specterum the frequency and time should be in the
%same length
%f=0:1:N-1 -(N-1)/2:1/N:(N-1)/2  -fs*(N-1)/2:fs/N:fs*(N-1)/2
f=(-fs/2):(fs/N):((fs/2)-(fs/N)); %Define frequency axis to draw the spectrum on
it shifting it
%-fs/N to remove one iteration which resemples the zero value to get the
%same length of the array of the fourier transform
SF=fftshift(fft(PAMsignal));
%plot frequency spectrum
figure(2)
plot(f,((10/N*7))*abs(SF),'b') % ((10/N*7)) to normalize the signal
title('Frequency Spectrum');
xlabel('f(Hz)');
ylabel('PAM(F)');
legend('Spectrum of PAM(t)');
grid

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%PART 2%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%                               Decleration
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
fs=50;
Ts=1/fs;
step=0.2;
t=0:Ts:2;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%                                 I/P Signal
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
inputSignal=2*sin(2*pi*t);
%plotting the input signal
figure(3)
plot(inputSignal,'b');
% title('Input Signal')
ylabel('Amplitude')
xlabel('Time')
xlim([0 100]); %2 cycles of the sin
grid
hold on
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%                            Delta Modulated Signal
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
Bitstream = zeros(1,101);
%intialization of bit stream to store the value of DM PCM code
Stair_Case_approx= zeros(1,101);
%intialization of the strais array to hold the value to be compared with
%value of the incoming input signal

length_ip_signal=length(inputSignal);
%vector array to hold the length the input signal depending on the time
```

```
%range stated at the Decleration top then this length is looped on through
%for loop that loops on all the input signal value and hold value of
%bitstream and stair case depending on the relative postion between each
%two signals at the iteration of the for loop
for i=1:length_ip_signal
    %if the signal value is higher than the stair case hold 1
    if inputSignal(i)>Stair_Case_approx(i)
        Bitstream(i)=1;
        Stair_Case_approx(i+1)=Stair_Case_approx(i)+step;
    else
    %if the signal value is higher than the stair case hold 0
        Bitstream(i)=0;
        Stair_Case_approx(i+1)=Stair_Case_approx(i)-step;
    end
end

%plotting the value of Stair_Case_approx
stairs(Stair_Case_approx);
grid
hold on

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%                              Recovered signal
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%

%for loop that loops on all the bitstream values and compare it to the
%value of the stair case to recover the signal
for i=1:Bitstream
    if Bitstream(i)>Stair_Case_approx(i)
        Bitstream(i)=0;
        Stair_Case_approx(i+1)=Stair_Case_approx(i)-step;
    else
        Bitstream(i)=1;
        Stair_Case_approx(i+1)=Stair_Case_approx(i)+step;
    end
end
%plotting the recovered signal
 plot(Stair_Case_approx,'r');
legend ('Input Signal m(t)','Delta Modulated Signal s(t)','Recovered Signal');
title('Time domain signals')
```