DIGITAL COMMUNICATION SYSTEM

DIGITAL ASSIGNEMENT -1

Name : K.Trinath Raja
Reg no: 18BEC1317

A)

SPRING SYSTEM

My topic is spring system, I got this idea while I was driving my car fast on the road, I didn’t notice the speed breakers in front of me. I was expecting a jerk but it didn’t happen, then I thought about the scientific reason behind it and I came to know about the coil spring compressor in the car which helps to absorb bumps in the road. The springs must be designed to oscillate at the right frequency and with the right amplitude. Controlling the amplitude, or how much the spring moves, can change the ride of the car tremendously. Since the spring must compress up to a certain distance to avoid the bump, I can say this system is non-linear. Scientifically speaking, the amount of work done by the spring is non – Linear. We can't just say that work = Force * Displacement here, as the Force is not constant. Hence, we need to calculate the Kinetic energy to calculate the work. It can be calculated by integrating the small forces which will result in the following equation.

$$\boxed{\textbf{Work} = (\tfrac{1}{2})*K*x^2}$$

DIGITAL ASSIGNMENT 2

(B) As we know, the sequence of data required can be generated using the above formula.

For obtaining this particular sequence as data, I have varied the value of x by assuming the value of spring constant to be 4**.**

We will first measure the data when the car was sent to the service centre. Over there they have a special machine called Universal Testing Machine (UTM), which is used to measure various properties of springs such as compression, tension, torsion, and fatigue.

Springs are an indispensable part of almost every mechanical design, but to serve their purpose, the materials need to have the required rigidity and elasticity. So, the data can be measured physically using the machine shown in the below figure, and can be used for further analysis. As my data is uniform, I can use uniform quantization as all the quantization levels are uniformly placed

Fig 1: Spring tesing machines

(C)

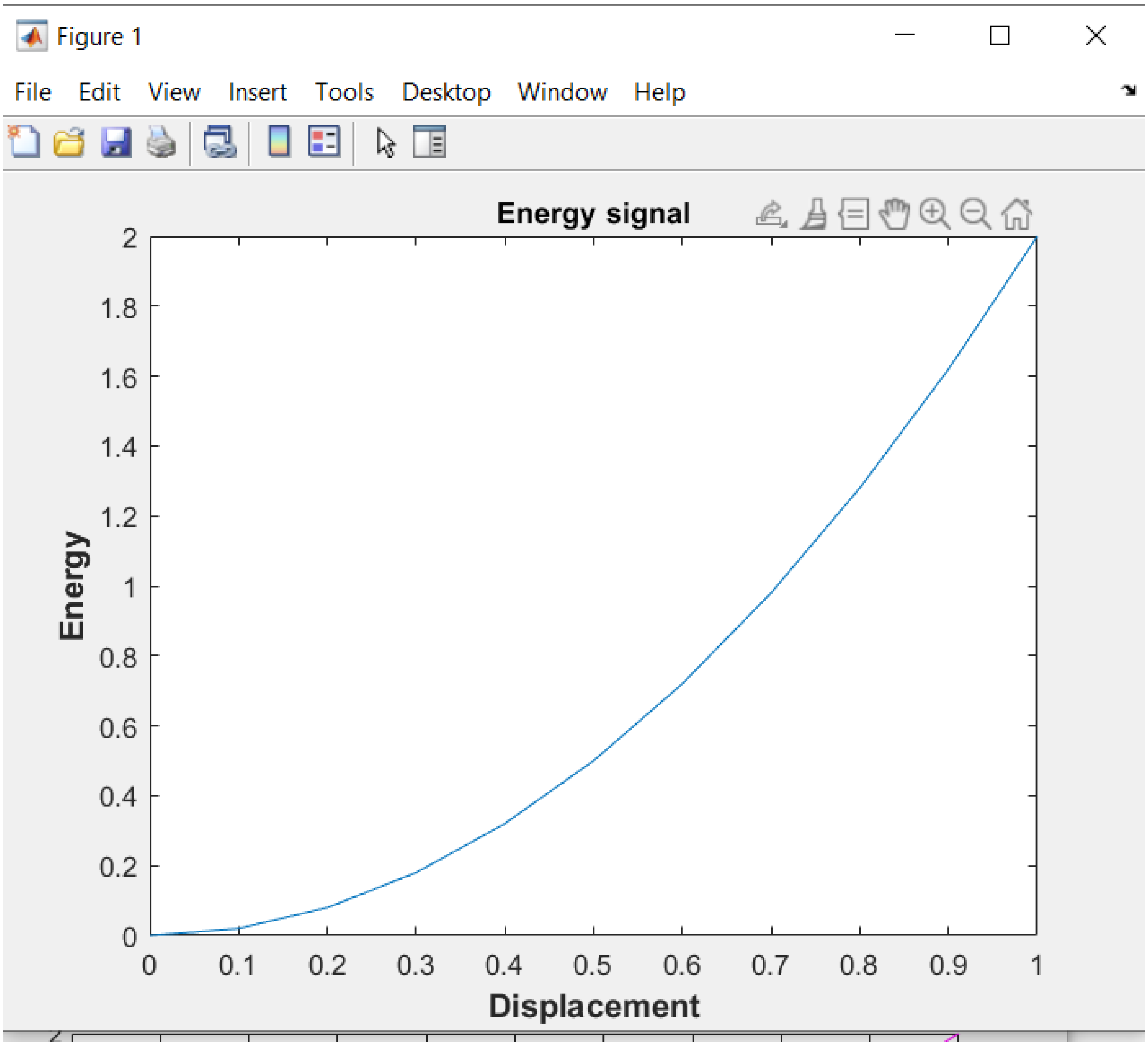
Matlab Code:

```
clc
clear all
k=4
x = 0:0.1:1 % generating few values as x
E = 1/2*k*x.^2; % using energy formula to clauclated KE
in spring
figure(1)
%plot(t,y,'bo-',t,v,'r+-',t,E,'g');
plot(x,E) %plotting origina signals
title('Energy signal');
xlabel('Displacement','FontSize',12,'FontWeight',"bold");
ylabel('Energy','FontSize',12,'FontWeight',"bold");
figure(2)
stem(x,E); % plot of sampled signal
title('SAMPLED SIGNAL');
xlabel('Displacement');
ylabel('Energy');

n1=4;%number of bits per sample
L=2^n1;%no of levels of quantization
xmax=2;
xmin=-2;
del=(xmax-xmin)/L; %defining del
partition=xmin:del:2 % definition of decision lines
codebook=xmin-(del/2):del:xmax+del/2; % definition of
representation levels
[index,quants]=quantiz(E,partition,codebook);%quantiz is
inbuilt function which is used to quantize the signal
% gives rounded off values of the samples
figure(3)
set(gca,'Fontsize',12,'Fontweight','bold')
stem(quants,"color",'r');%plotting of quantized signal
title('QUANTIZED SIGNAL(18BEC1317)');
xlabel('Displacement','FontSize',12,'FontWeight',"bold");
ylabel('Energy','FontSize',12,'FontWeight',"bold");


% NORMALIZATION
l1=length(index); % to convert 1 to n as 0 to n-1
indicies
```

```
for i=1:l1
    if (index(i)~=0)
        index(i)=index(i)-1;
    end
end
l2=length(quants);
for i=1:l2 % to convert the end representation levels
within the range.
    if(quants(i)==xmin-(del/2))
        quants(i)=xmin+(del/2);
    end
    if(quants(i)==xmax+(del/2))
        quants(i)=xmax-(del/2);
    end
end
% ENCODING
code=de2bi(index,'left-msb'); % DECIMAL TO BINANRY CONV
OF INDICIES
k=1;
for i=1:l1 % to convert column vector to row vector
    for j=1:n1
        coded(k)=code(i,j);
        j=j+1;
        k=k+1;
    end
    i=i+1;
end
figure(4);
stairs(coded);
axis([0 190 -2 2])
%plot of digital signal
title('DIGITAL SIGNAL(18BEC1317)');
set(gca,'Fontsize',12,'Fontweight','bold')
xlabel('Displacement','FontSize',12,'FontWeight',"bold");
ylabel('Energy','FontSize',12,'FontWeight',"bold");
%Demodulation
code1=reshape(coded,n1,(length(coded)/n1));
index1=bi2de(code1,'left-msb'); %converting from decimal
to binary
resignal=del*index+xmin+(del/2);
figure(5);%plot of demodulated signal compared to
original signl
subplot(2,1,1)%plot of demodulated signal
plot(resignal,"color",'k');
title('DEMODULATAED SIGNAL(18BEC1317)');
xlabel('Displacement','FontSize',12,'FontWeight',"bold");
ylabel('Energy','FontSize',12,'FontWeight',"bold");
```

```
subplot(2,1,2)
plot(x,E,"color",'m');%plot of original signal
title('ORIGINALSIGNAL(18BEC1317)');
xlabel('Displacement','FontSize',12,'FontWeight',"bold");
ylabel('Energy','FontSize',12,'FontWeight',"bold");
```

Outputs:



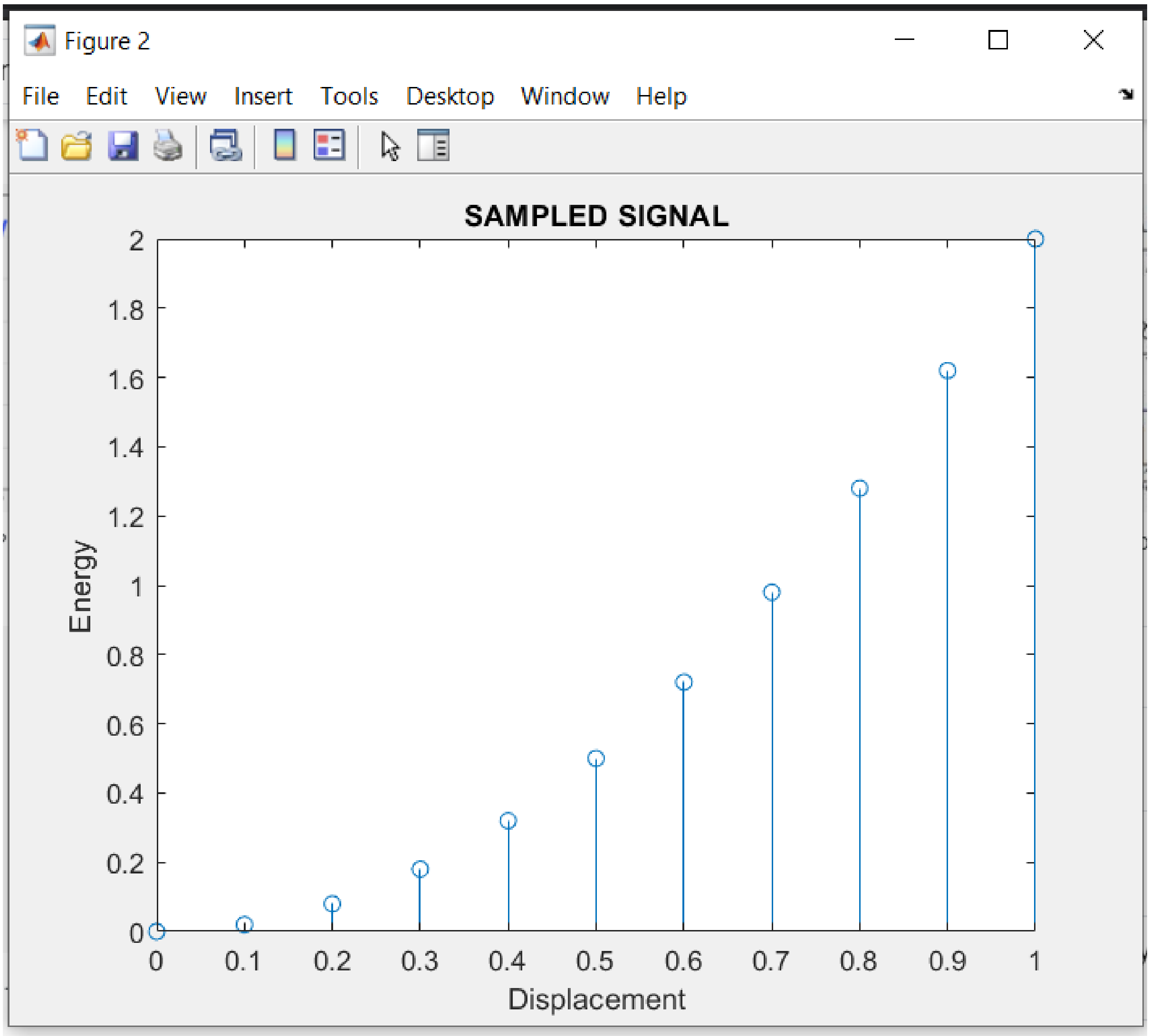


Figure 2
File Edit View Insert Tools Desktop Window Help
SAMPLED SIGNAL
Energy
Displacement
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
0.1
0.3
0.5
0.7
0.9

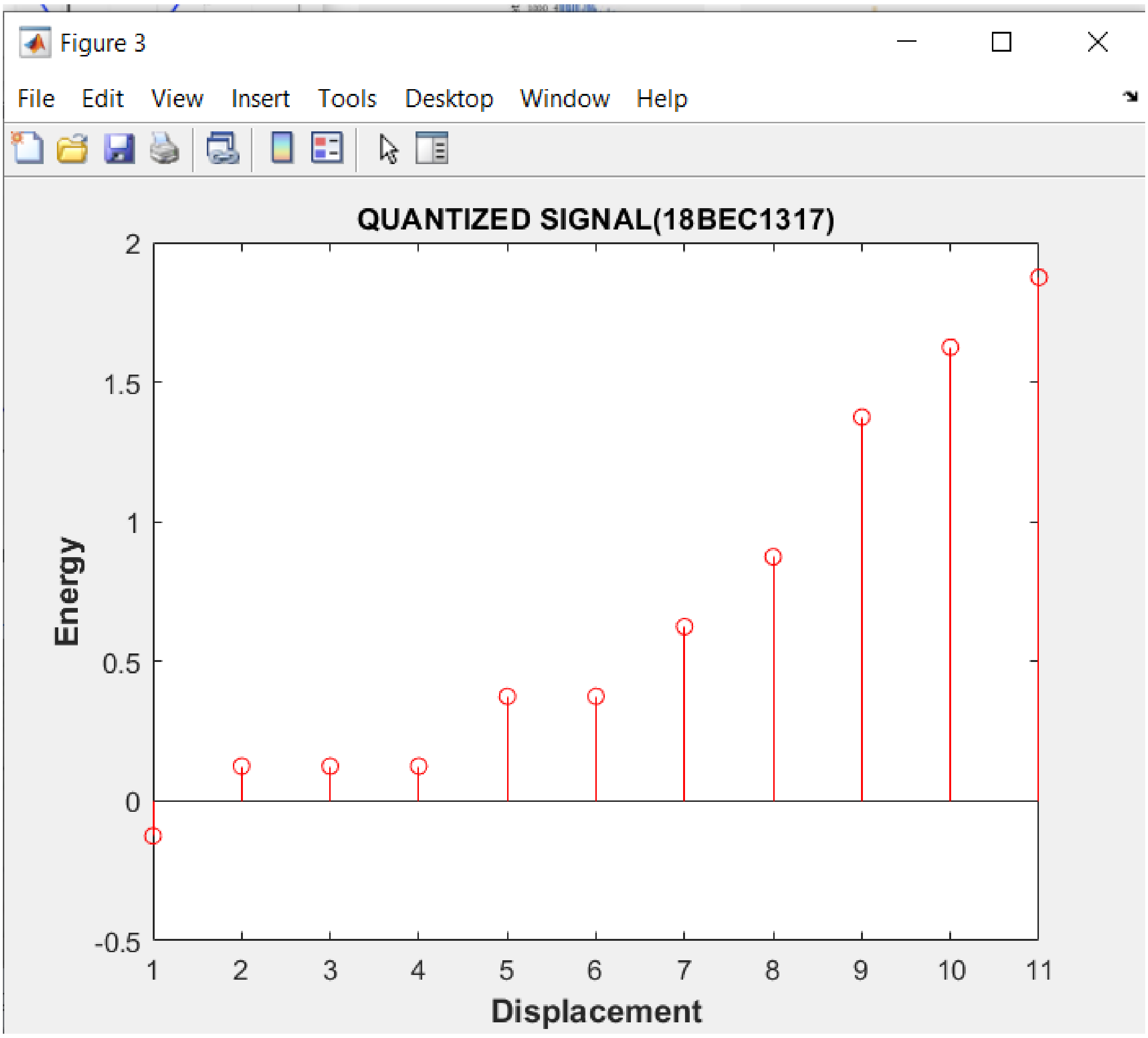
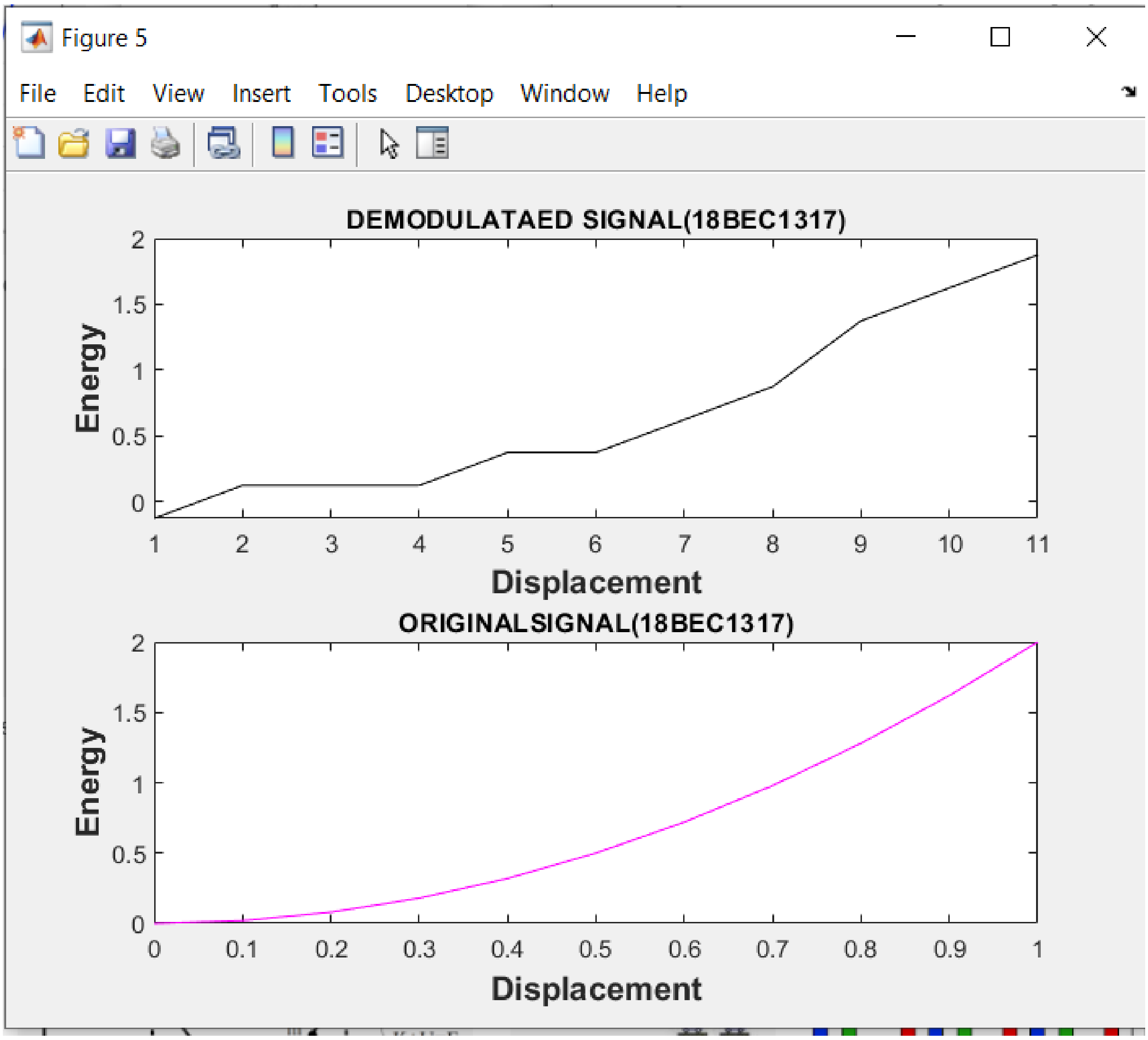


Figure 3
File Edit View Insert Tools Desktop Window Help
QUANTIZED SIGNAL(18BEC1317)
Energy
Displacement
2
1.5
1
0.5
0
-0.5
1 2 3 4 5 6 7 8 9 10 11



Result: We have successfully generated the data and sampled it, the we have quantized and reconstructed the original signal using demodulation

GITHUB link

https://github.com/KopparapuTrinathraja/18bec1317