Welcome all to Digital Communication Lab

Presented by

Jayashri Kawale

Comparison of Digital and analog communication System

Digital Communication System	Analog Communication System
Advantages	Disadvantages
1. Inexpensive digital Circuit	1. Expensive analog component L and C
2. Privacy Preserved (Data Encryption)	2. No Privacy
3. Can merge different data(Voice video and data)	3. Cannot merge data from different sources
and transmit over a common digital communication system	
4. Error correction by coding	4. No error correction capability
Disadvantages 1. Larger B.W 2. Synchronization problem is relatively difficult	Advantages 1. Smaller B.W 2. Synchronization problem is relatively easier

Generation of digital message signal by using square function:

X=A*square(2*pi*f*t)

For given Data Bits Waveform Generation:

```
n=[1,1,1,0,1];
for k = 1:length(n)
       if n(k)==1
       amplitude=4;
       else
       amplitude=0;
       end
end

i=1;
t=0:0.01:length(n);
for j=1:length(t)
      if t(j)<=i
       y(j)=n(i);
       else
       y(j)=n(i);
       i=i+1;
       end
end
plot(t,y,'r-');
```

Experiment No- 1

To know the Principles of Sampling and Quantization

Practical Aspects of Sampling

- Sampling Theorem
- Method of Sampling
- Significance of sampling rate
- Anti- aliasing filter
- Applications of Sampling Theorem (PAM, TDM)

Analog to digital is basically a two step process

1. Sampling 2. Quantization

An analog signal is continuous in both it’s amplitude and it’s time period.

So, we have a two task to do now we have to change continuous amplitude signal to a discrete amplitude signal.

In a same way to a continuous time signal to a discrete time signal. So these are two task .

Now to convert a continuous time signal to a discrete time signal we are going for a technique called **Sampling.**

Where all the signals are converted to samples of time per second.

Quantization:

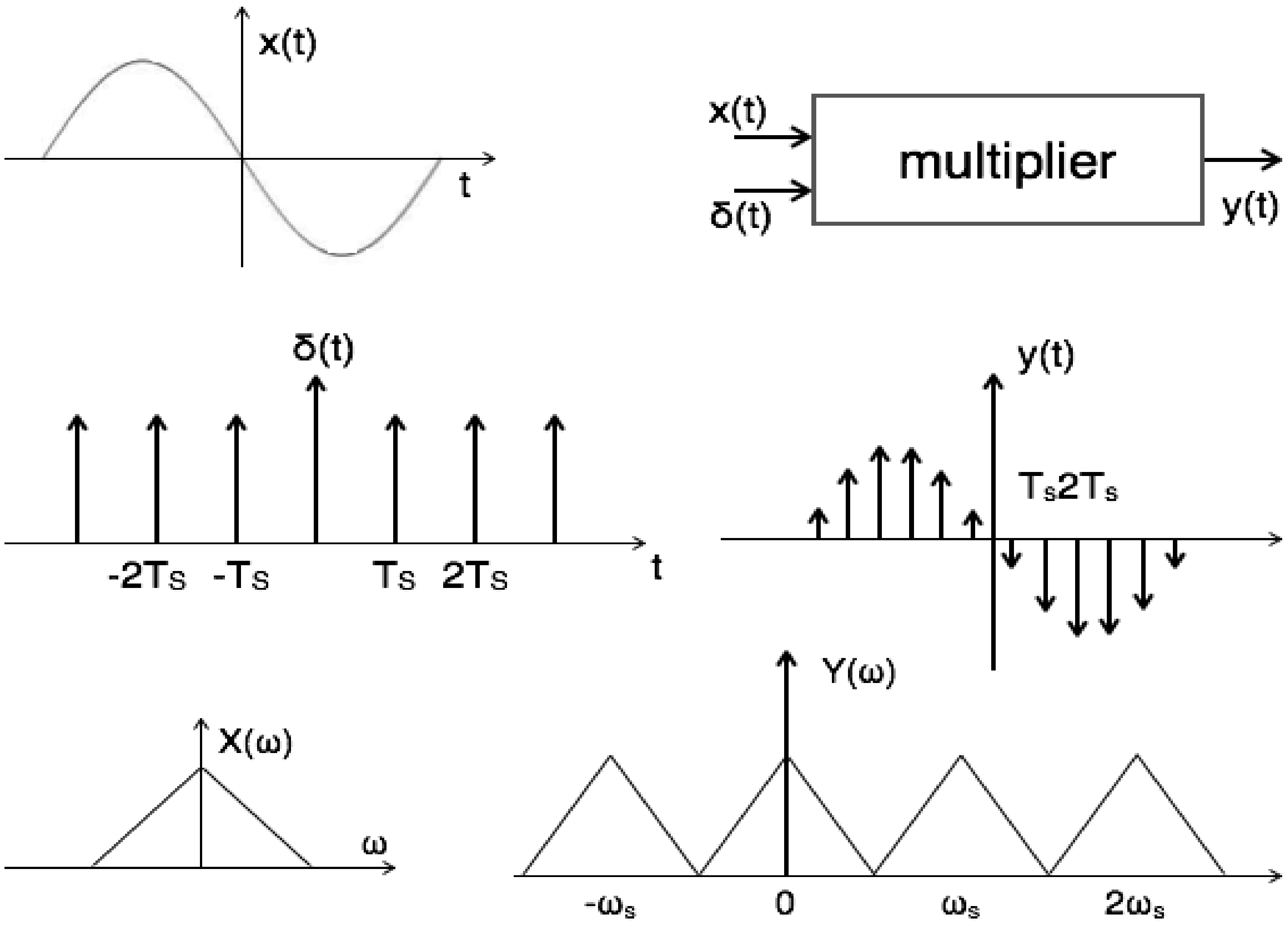
- And if the second task is the continuous amplitude need to be converted to discrete amplitude then we are going for a technique called quantization.
- Where we round off the amplitude value to the nearest value.

So now our signal output will be a digital signal.

Now the question arises why we go for sampling theorem.

- If we consider a periodic signal to a exactly recover a original signal we should adopt a sampling frequency greater than or equal to twice of maximum frequency.($fs \geq 2\ fm$)
- If $fs < 2\ fm$ ---- Periodic signal will be overlap on each other. So we won't get output properly so aliasing occur.
- $Fs > 2fm$ ----- Signal will be reconstructed properly but there will be some guard band present.
- $Fs = 2\ fm$ ------ The samples are recovered or reconstructed properly.

Theory:

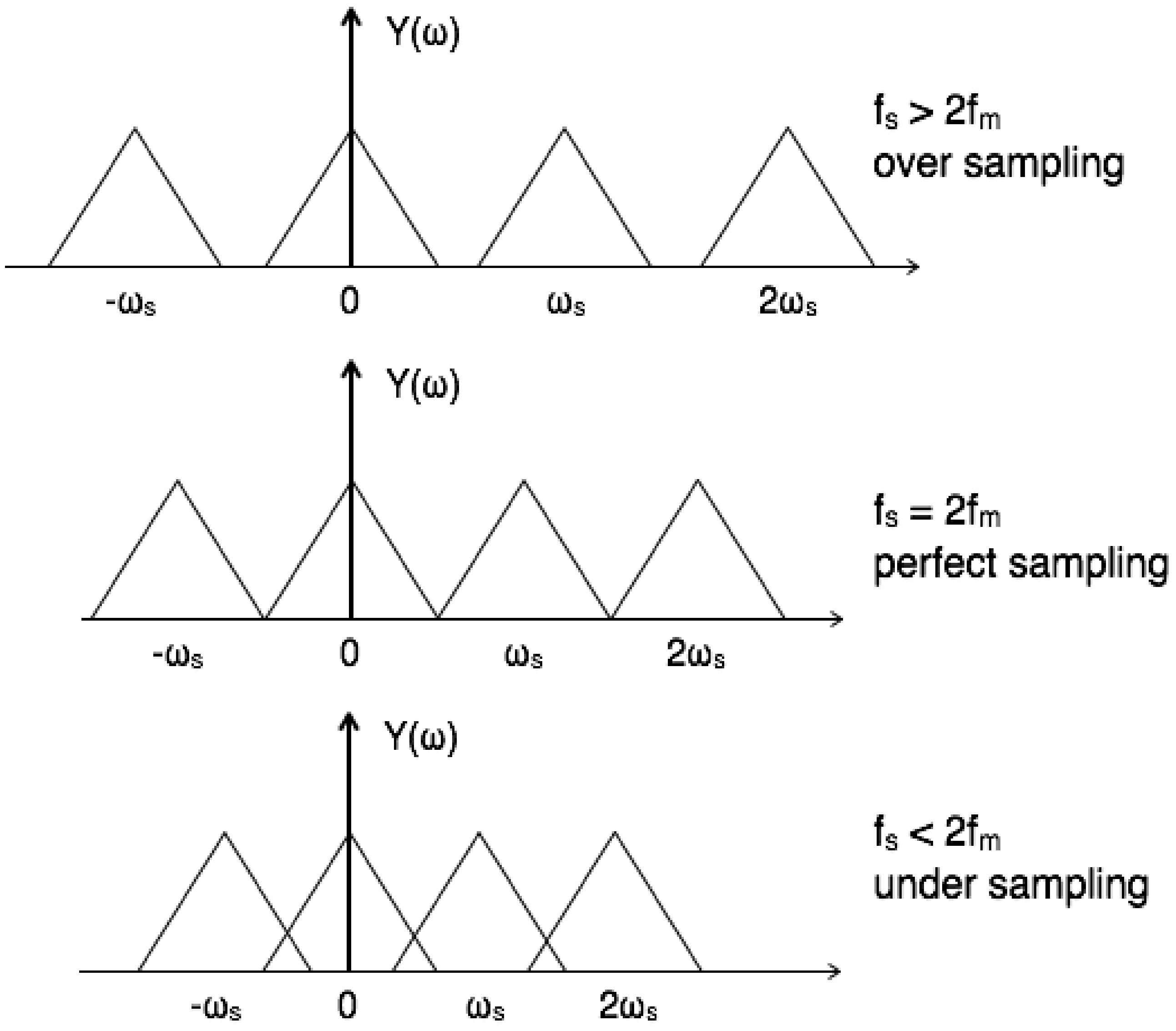

x(t)
t
x(t)
δ(t)
multiplier
y(t)
δ(t)
-2Ts -Ts Ts 2Ts
t
y(t)
Ts2Ts
X(ω)
ω
Y(ω)
-ωs
0
ωs
2ωs

Statement:

- A continuous time signal can be represented in its samples and can be recovered back when sampling frequency fs is greater than or equal to the twice the highest frequency component of message signal. i. e.

- $fs \geq 2fm$.

Theory:



Y(ω)
-ωs
0
ωs
2ωs
fs > 2fm
over sampling
Y(ω)
-ωs
0
ωs
2ωs
fs = 2fm
perfect sampling
Y(ω)
-ωs
0
ωs
2ωs
fs < 2fm
under sampling

MATLAB Code:

```
%sampling
close all;
clc;
t=0:0.01:1; % Time Vector
fm=10; % Message (Input) signal amplitude
Am=1; % (Input) signal amplitude
```

MATLAB Code:

```
x=Am*sin(2*pi*fm*t); % Message signal
subplot(2,2,1);
plot(t,x,'linewidth',2);
xlabel('time');
ylabel('amplitude');
grid;
title('Message input signal');
```

MATLAB Code:

```
n1=-5:1:5;
fs1=1.6*fm;
fs2=2*fm;
fs3=8*fm;

x1=Am*cos(2*pi*fm/fs1*n1);
subplot(2,2,2);
stem(n1,x1,'linewidth',3);
xlabel('number of samples');
ylabel('amplitude');
grid on;
title('under sampling');
```

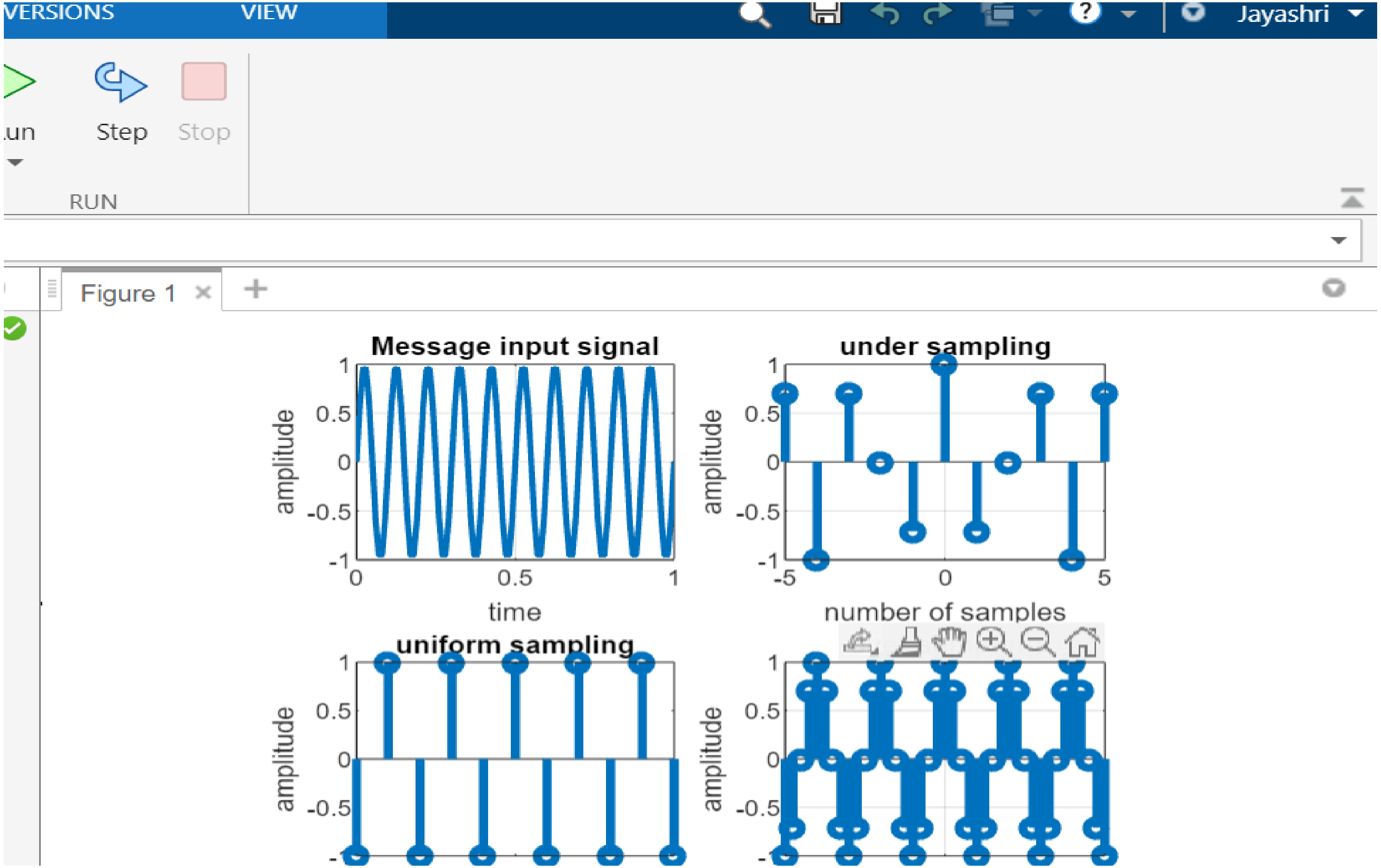
MATLAB Code:

```
n2=-5:1:5;
x2=cos(2*pi*fm/fs2*n2);
subplot(2,2,3);
stem(n2,x2,'linewidth',3);
xlabel('number of samples’);
ylabel('amplitude');
hold on;
title('uniform sampling');
```

MATLAB Code:

```
n3=-20:1:20;
x3=cos(2*pi*fm/fs3*n3);
subplot(2,2,4);stem(n3,x3,'linewidth',3);
hold on;
xlabel('number of samples');
ylabel('amplitude');
grid;
title('over sampling');
```

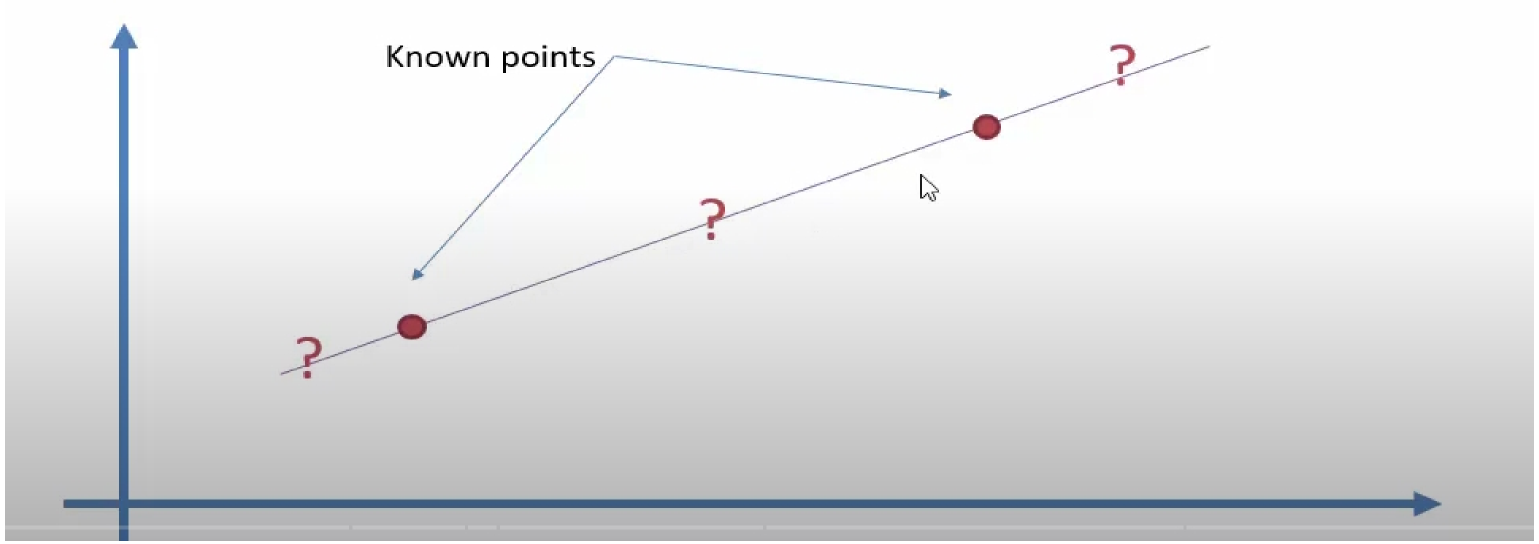
Output:



VERSIONS
VIEW
Jayashri
un
Step
Stop
RUN
Figure 1
Message input signal
amplitude
time
under sampling
amplitude
number of samples
uniform sampling
amplitude
amplitude

Interpolation of Curve Fitting:

- Interpretation is used to estimate the data points between two known points.
- The most common Interpretation technique is Linear Interpretation.



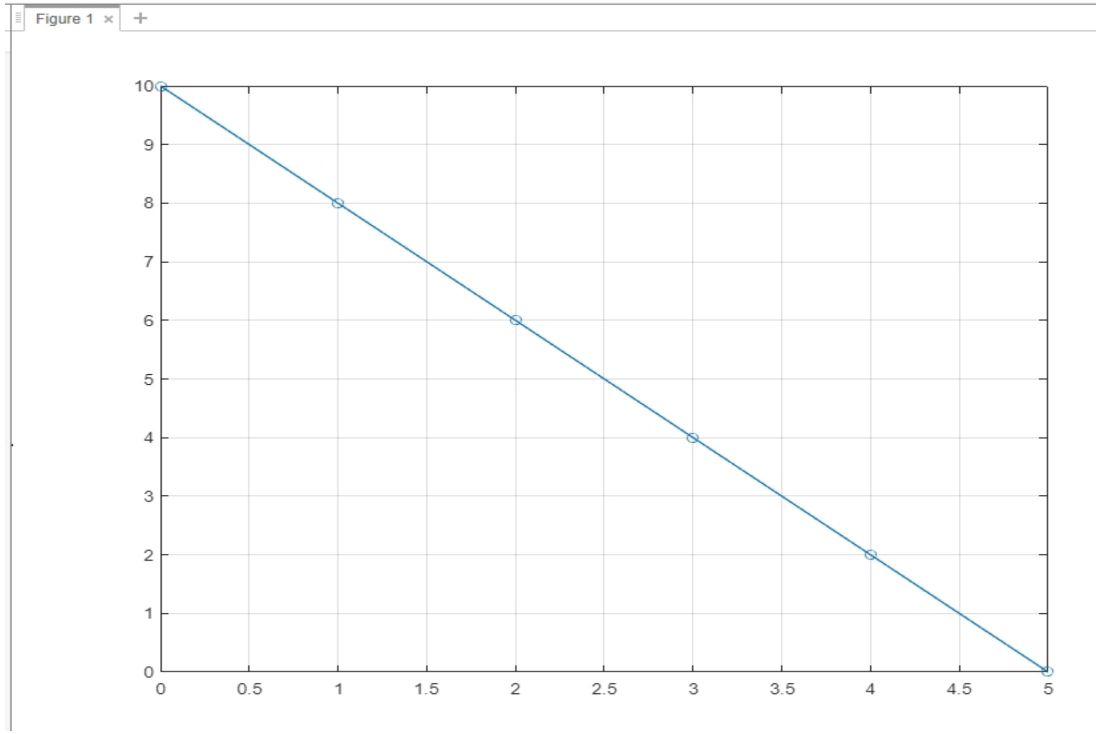
Interpolation

- Interpolation is used to estimate data points between two known points. The most common Interpolation technique is Linear Interpolation.
- In MATLAB we can use the interp 1() function.
- The default is linear Interpolation, but there are other types available, such as:
- Linear
- Nearest
- Spline
- Cubic
- Type “help interp1” in order to read more about the different options.

Problem 1: We want to find the Interpolated Value for example x= 2.5

Data:

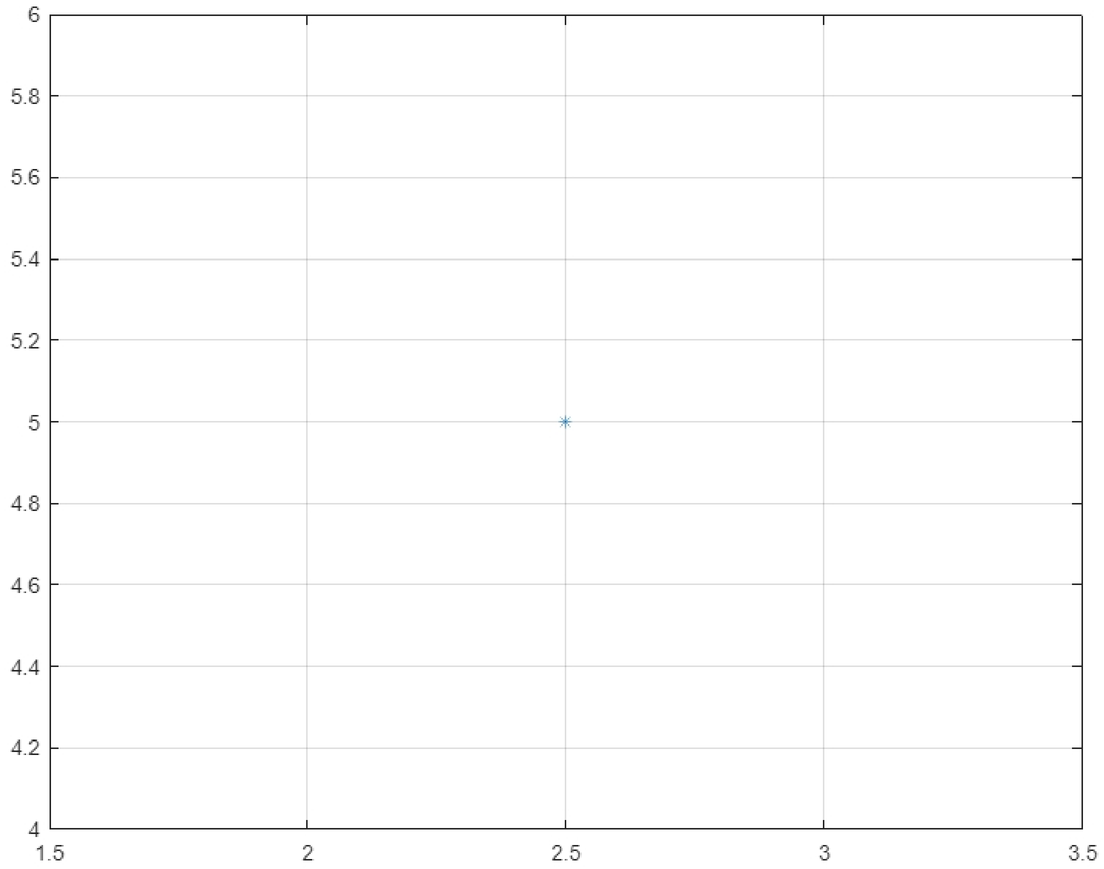
X	Y
0	10
1	8
2	6
3	4
4	2
5	0



Program:

```
x=0:5;
y=[10,8,6,4,2,0];
plot(x,y,'-o')
grid
new_x= 2.5;
new_y= interp1(x,y,new_x);
plot(new_x,new_y,'*')
grid
```

Output:



Q1. What is Trapezoidal Rule?

Trapezoidal Rule is an integration rule, in Calculus, that evaluates the area under the curves by dividing the total area into smaller trapezoids rather than using rectangles.

Q2. Why the rule is named after trapezoid?

The name trapezoidal is because when the area under the curve is evaluated, then the total area is divided into small trapezoids instead of rectangles. Then we find the area of these small trapezoids in a definite interval.

Q3. What is the difference between Trapezoidal rule and Riemann Sums rule?

In trapezoidal rule, we use trapezoids to approximate the area under the curve whereas in Riemann sums we use rectangles to find area under the curve, in case of integration.

Random Number:

a) Write a MATLAB code for generation of Random number

b) Write a MATLAB code to Create Probability Density Function

```
x= randn(1,1000);
x
figure;
hist(x)

[f,a] = hist(x,50) % 50 is the number of bin a is the location of each bin f is the total number in 'x' within each pin f

figure;
bar(a,f/trapz(a,f))
```

Output:

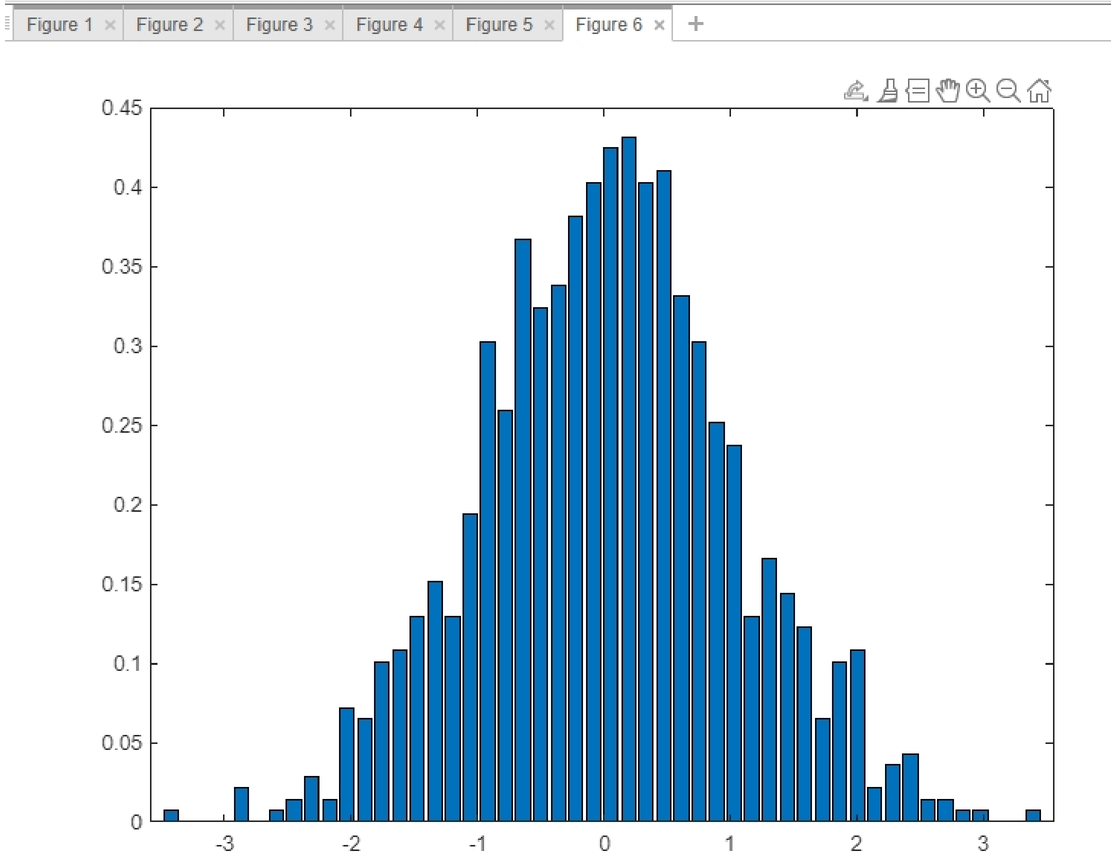


Figure 1
Figure 2
Figure 3
Figure 4
Figure 5
Figure 6
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
-3
-2
-1
0
1
2
3

References:

[1]https://www.tutorialspoint.com/signals_and_systems/signals_sampling_theorem.htm

[2] https://www.youtube.com/watch?v=km7VQfWrrw0

[3] https://www.geeksforgeeks.org/trapezoidal-numerical-integration-in-matlab/

[4] https://www.youtube.com/watch?v=LgbixtH9cC0 - Interpretation of Curve fitting

[5] https://www.youtube.com/watch?v=mv61vEPgksc Estimate the count, probability, PDF, CDF, using MATLAB

[6] CDF: https://www.youtube.com/watch?v=gSo0apQdVTA