Amit Kumar Yadav

194107/ ECE-A

Comm. Sys. LAB

## Experiment-3

**Aim**- Develop a matlab code to sample and then reconstruct an analog signal from sample.

**Software**- MATLAB R2021b

**Theory**- Sampling is defined as the reductions of continous time signal (analog signal) to discrete time signal. We take a message signal as input (which is an analog signal)

$$m(t) = A_m \sin(2\pi f_m t)$$

$2\pi f_m = \omega_m \rightarrow$ maximum freq. component of $m(t)$.

$A_m \rightarrow$ amplitude of message signal.

In order to do sampling for the analog signal, we need a sampling frequency $f_s$.

**Sampling Theorm**⇒

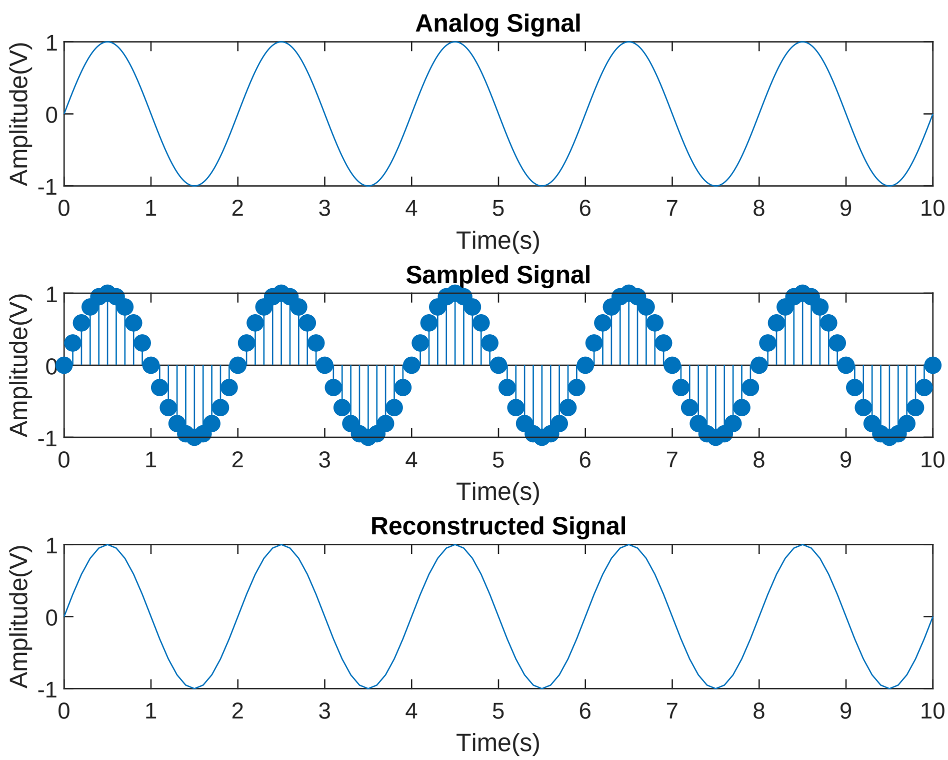
Sampling theorm states that a signal can be represented in its samples and can be recovered back only when sampling frequency is greater than or equal to twice of the maximum frequency component present in the signal i.e. $\boxed{f_s \geq 2f_m}$.

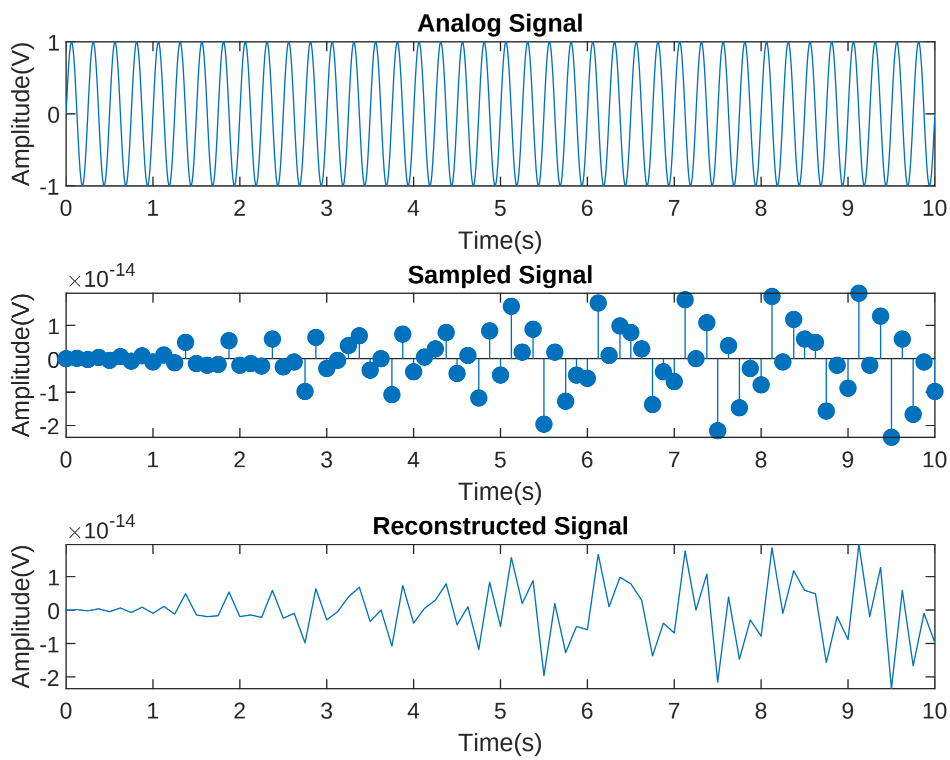
⇒ If $f_s < 2f_m$, we can't obtain the continous time signal from the sampled signal accurately.

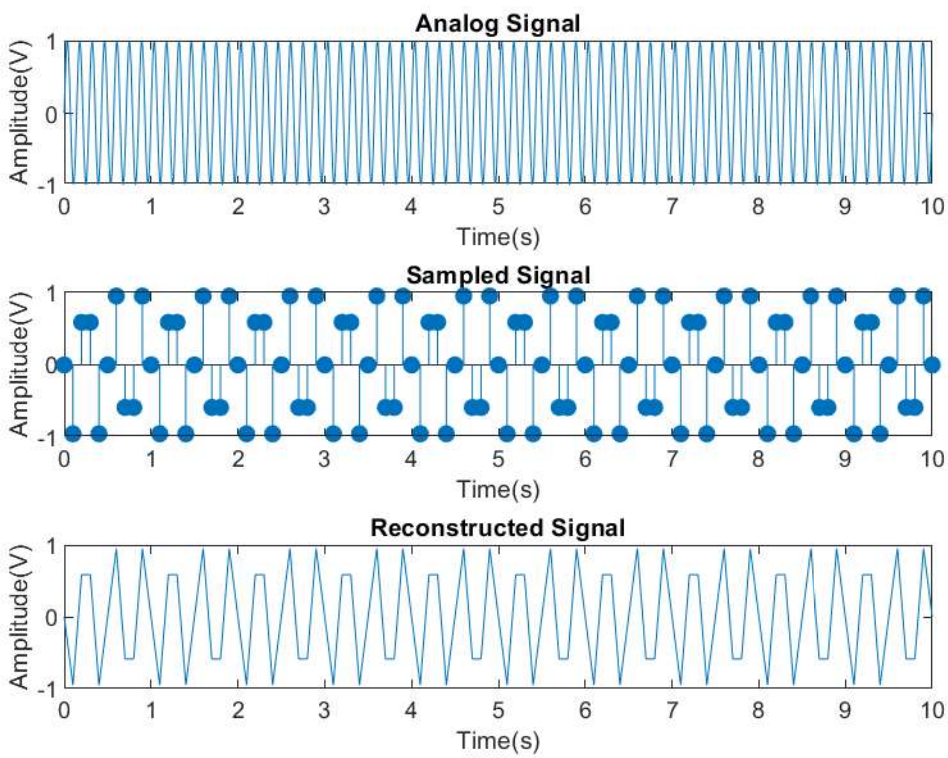
```
v%%%%%%%%%%%%%
% Author: Amit Kumar Yadav
% Roll: 194107 (ECE-A)
% Lab Date: 31-01-2022
%%%%%%%%%%%%%

close all
clc

% CASE-1: fs>2fm
% for plotting original analog signal
% very large frequency to get accurate plot
fs=100; %Sampling frequency
Ts=1/fs;
t=0:Ts:10;

fm=0.5; %Frequency of message signal
Am=1; %Amplitude of message signal
xt=Am.*sin(2*pi*fm.*t); %Message signal
figure(1)
subplot(3,1,1)
plot(t,xt)
title('Analog Signal')
xlabel('Time(s)')
ylabel('Amplitude(V)')

% taking fs>2fm
fs=10; % 10>2*0.5
Ts=1/fs;
t=0:Ts:10;
xt=Am.*sin(2*pi*fm.*t);
subplot(3,1,2)
stem(t,xt,'filled') %converts the given analog signal to digital signal
title('Sampled Signal')
xlabel('Time(s)')
ylabel('Amplitude(V)')

%reconstructing the analog signal from the digital samples
subplot(3,1,3)
plot(t,xt)
title('Reconstructed Signal')
xlabel('Time(s)')
ylabel('Amplitude(V)')

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

% CASE-2: fs=2fm
fs=100;
Ts=1/fs;
```

```
t=0:Ts:10;
fm=4;
Am=1;
xt=Am.*sin(2*pi*fm.*t);
figure(2)
subplot(3,1,1)
plot(t,xt)
title('Analog Signal')
xlabel('Time(s)')
ylabel('Amplitude(V)')

% taking fs=2fm
fs=8; %8=2*4
Ts=1/fs;
t=0:Ts:10;
xt=Am.*sin(2*pi*fm.*t);
subplot(3,1,2)
stem(t,xt,'filled')
title('Sampled Signal')
xlabel('Time(s)')
ylabel('Amplitude(V)')

%reconstructing from the digital samples
subplot(3,1,3)
plot(t,xt)
title('Reconstructed Signal')
xlabel('Time(s)')
ylabel('Amplitude(V)')

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

% CASE-3: fs<2fm
fs=100;
Ts=1/fs;
t=0:Ts:10;
fm=7;
Am=1;
xt=Am.*sin(2*pi*fm.*t);
figure(3)
subplot(3,1,1)
plot(t,xt)
title('Analog Signal')
xlabel('Time(s)')
ylabel('Amplitude(V)')

% taking fs<2fm
fs=10; %10<2*7
Ts=1/fs;
t=0:Ts:10;
```

```
xt=Am.*sin(2*pi*fm.*t);
subplot(3,1,2)
stem(t,xt,'filled')
title('Sampled Signal')
xlabel('Time(s)')
ylabel('Amplitude(V)')

%reconstructing from the digital samples
subplot(3,1,3)
plot(t,xt)
title('Reconstructed Signal')
xlabel('Time(s)')
ylabel('Amplitude(V)')
```


Analog Signal
Amplitude(V)
1
0
-1
0
1
2
3
4
5
6
7
8
9
10
Time(s)
Sampled Signal
Amplitude(V)
1
0
-1
0
1
2
3
4
5
6
7
8
9
10
Time(s)
Reconstructed Signal
Amplitude(V)
1
0
-1
0
1
2
3
4
5
6
7
8
9
10
Time(s)


Analog Signal
Amplitude(V)
1
0
-1
0 1 2 3 4 5 6 7 8 9 10
Time(s)
Sampled Signal
×10^-14
Amplitude(V)
1
0
-1
-2
0 1 2 3 4 5 6 7 8 9 10
Time(s)
Reconstructed Signal
×10^-14
Amplitude(V)
1
0
-1
-2
0 1 2 3 4 5 6 7 8 9 10
Time(s)



Analog Signal
Amplitude(V)
1
0
-1
0
1
2
3
4
5
6
7
8
9
10
Time(s)
Sampled Signal
Amplitude(V)
1
0
-1
0
1
2
3
4
5
6
7
8
9
10
Time(s)
Reconstructed Signal
Amplitude(V)
1
0
-1
0
1
2
3
4
5
6
7
8
9
10
Time(s)

## Explainations-

→ Here, we take a message signal m(t) and convert that analog signal into digital signal and then again convert or reconstrult the digital signal into analog signal.

→ We take some value of amplitude of message signal, and frequency of message signal and change the sampling frequency according to the value of frequency of message signal.

→ Stem command is used to make an analog signal to digital signal.

figure 1, graphs for case $f_s > 2f_m$
figure 2, graph for case $f_s = 2f_m$
figure 3, graph for case $f_s < 2f_m$.

## Result-

from this, we can say that if $f_s < 2f_m$, we can't get the analog signal correctly. When $f_s >= 2f_m$, we get the reconstructed original signal somewhat accurately.