Department of Electrical Engineering
Indian Institute of Technology Kharagpur

Digital Signal Processing Laboratory (EE39203)
Autumn, 2022-23

Experiment 1 **Discrete and Continuous Time Signals**

Slot: **X** Date: **09/08/2023**

Student Name: **P. Manoj Kumar** Roll No.: **21IE10027**

Grading Rubric

	Tick the best applicable per row			Points
	Below Expectation	Lacking in Some	Meets all Expectation	
Completeness of the report				
Organization of the report (5 pts) *With cover sheet, answers are in the same order as questions in the lab, copies of the questions are included in report, prepared in LaTeX*				
Quality of figures (5 pts) *Correctly labelled with title, x-axis, y-axis, and name(s)*				
Understanding of continuous and discrete-time signals (15 pts) *Matlab figures, questions*				
Ability to compute integral manually and in Matlab (30 pts) *Manual computation, Matlab figures, Matlab codes, questions*				
Ability to define and display functions (1D and 2D) (30 pts) *Matlab figures, Matlab codes, questions*				
Understanding of sampling (15 pts) *Matlab figures, questions*				
			TOTAL (100 pts)	

Total Points (100): TA Name: TA Initials:

Digital Signal Processing Laboratory (EE39203)

P Manoj Kumar (21IE10027)

Experiment 1 - Discrete and Continuous Time Signals

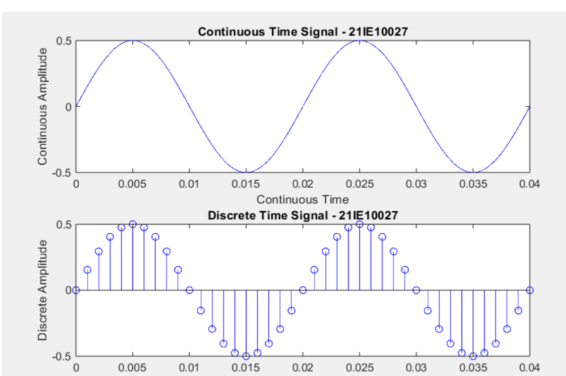
1 Continuous-Time Vs. Discrete-Time

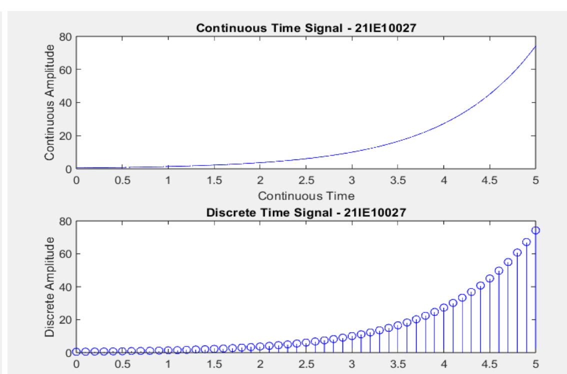
1.1 Displaying Continuous-Time Vs. Discrete-Time

The continuous and discrete time signals are plotted below for both the functions.
First signal is a sinusoidal signal with a frequency of 50Hz and amplitude of 0.5, sampled with a sampling frequency of 1KHz.
Second signal is a real exponential signal having coefficient of 0.5 is sampled with a sampling frequency of 10Hz.

```
A = 0.5; % Amplitude
f = 50;  % Frequency
N = 2;   % Cycles
t = linspace(0,N/f,10000);
x = A*sin(2*pi*f*t);
subplot(2,1,1);
plot(t,x,'b'); % Continuous-Time Signal Plotting
xlabel("Continuous Time");
ylabel("Continuous Amplitude");
title("Continuous Time Signal - 21IE10027");
fs = 20*f; % Sampling Frequency
n = 0:1/fs:N/f;
X = A*sin(2*pi*f*n);
subplot(2,1,2);
stem(n,X,'b'); % Discrete-Time Signal Plotting
xlabel("Discrete Time");
ylabel("Discrete Amplitude");
title("Discrete Time Signal - 21IE10027");
```

```
A = 0.5; % Amplitude
t = linspace(0,5,10000);
x = A*exp(t);
subplot(2,1,1);
plot(t,x); % Continuous Signal Plotting
xlabel("Continuous Time");
ylabel("Continuous Amplitude");
title("Continuous Time Signal - 21IE10027");
fs = 10; % Sampling Frequency
n = 0:1/fs:5;
X = A*exp(n);
subplot(2,1,2);
stem(n,X); % Discrete-Time Signal Plotting
xlabel("Discrete Time");
ylabel("Discrete Amplitude");
title("Discrete Time Signal - 21IE10027");
```

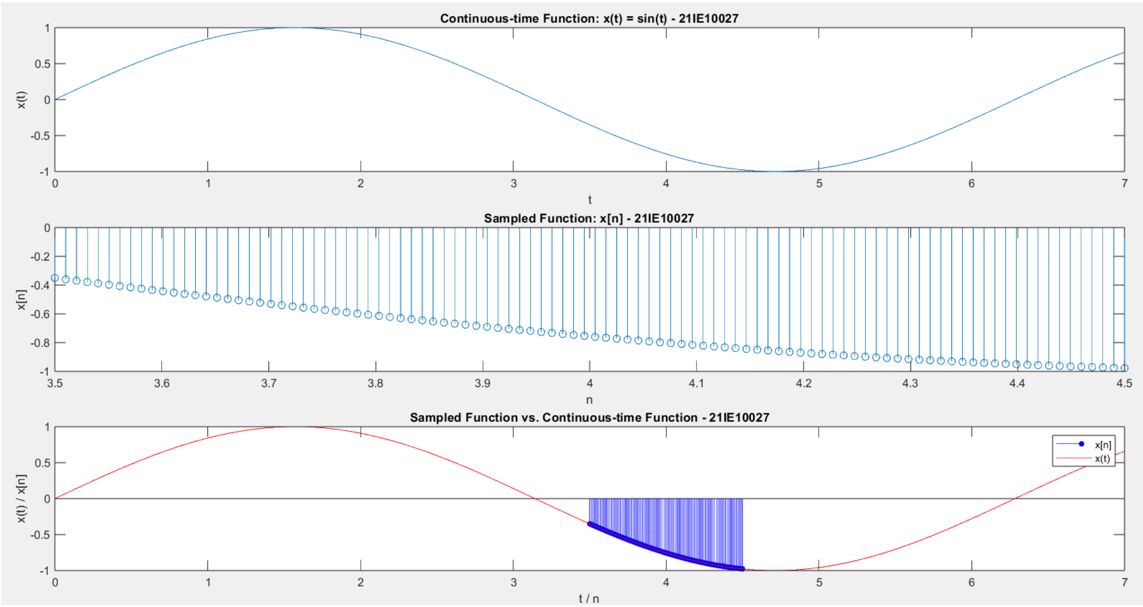




1.2 Vector Index vs. Time

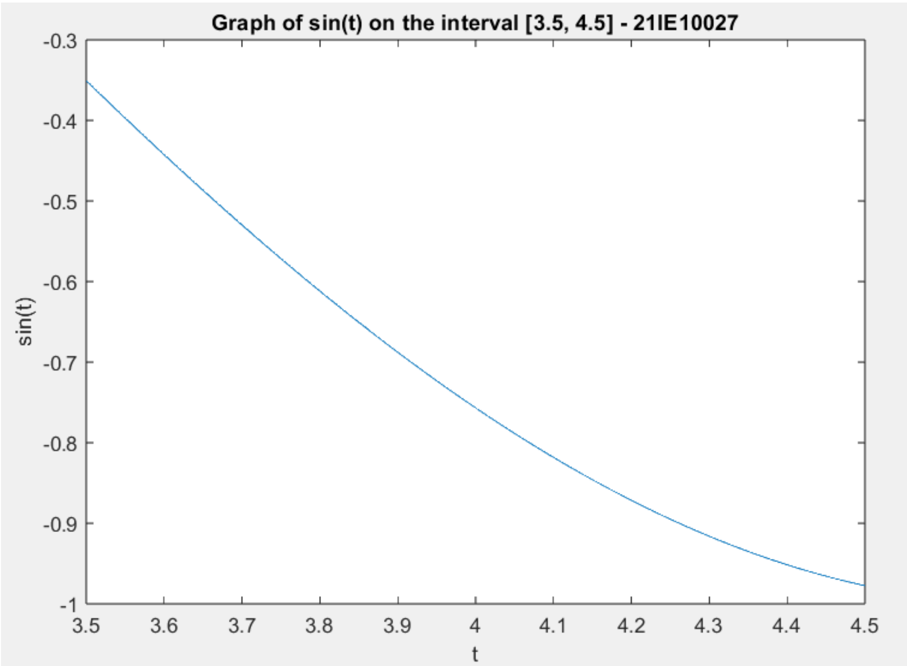
The continuous-time function $x(t) = sin(t)$ and $x[n] = sin[n]$ is plotted in the respective intervals and both the plots were compared .

```
t_cont = linspace(0, 7, 1000);  % Continuous time vector from 0 to 10
t_samp = linspace(3.5, 4.5, 100);   % Sampled time vector from 3.5 to 4.5
x_cont = sin(t_cont);% Continuous-time function
x_samp = sin(t_samp);% Sampled function
subplot(3,1,1);
plot(t_cont, x_cont);% Plot the continuous-time function
title('Continuous-time Function: x(t) = sin(t) - 21IE10027');
xlabel('t');
ylabel('x(t)');
subplot(3,1,2);
stem(t_samp, x_samp);% Plot the sampled function
title('Sampled Function: x[n] - 21IE10027');
xlabel('n');
ylabel('x[n]');
% Plot the sampled function with continuous-time plot
subplot(3,1,3);
stem(t_samp, x_samp, ...
    'b', 'Marker', 'o', 'MarkerSize', 4, 'MarkerFaceColor', 'b');
hold on;
plot(t_cont, x_cont, 'r');
hold off;
title('Sampled Function vs. Continuous-time Function - 21IE10027');
xlabel('t / n');
ylabel('x(t) / x[n]');
legend('x[n]', 'x(t)');
```



The function is $x(t) = sin(t)$ for the values of t on the interval [3.5, 4.5] .

```
% Print the graph of sin(t) for the values of t on the interval [3.5, 4.5]
t = linspace(3.5, 4.5, 1000);
x = sin(t);
figure; % Create a new figure
plot(t, x);
title('Graph of sin(t) on the interval [3.5, 4.5] - 21IE10027');
xlabel('t');
ylabel('sin(t)');
```



1.3 Analytical Calculation

1. $\int_0^1 e^t \, dt = e^t\big|_0^1 = e^1 - e^0 = e - 1 = 1.71828$

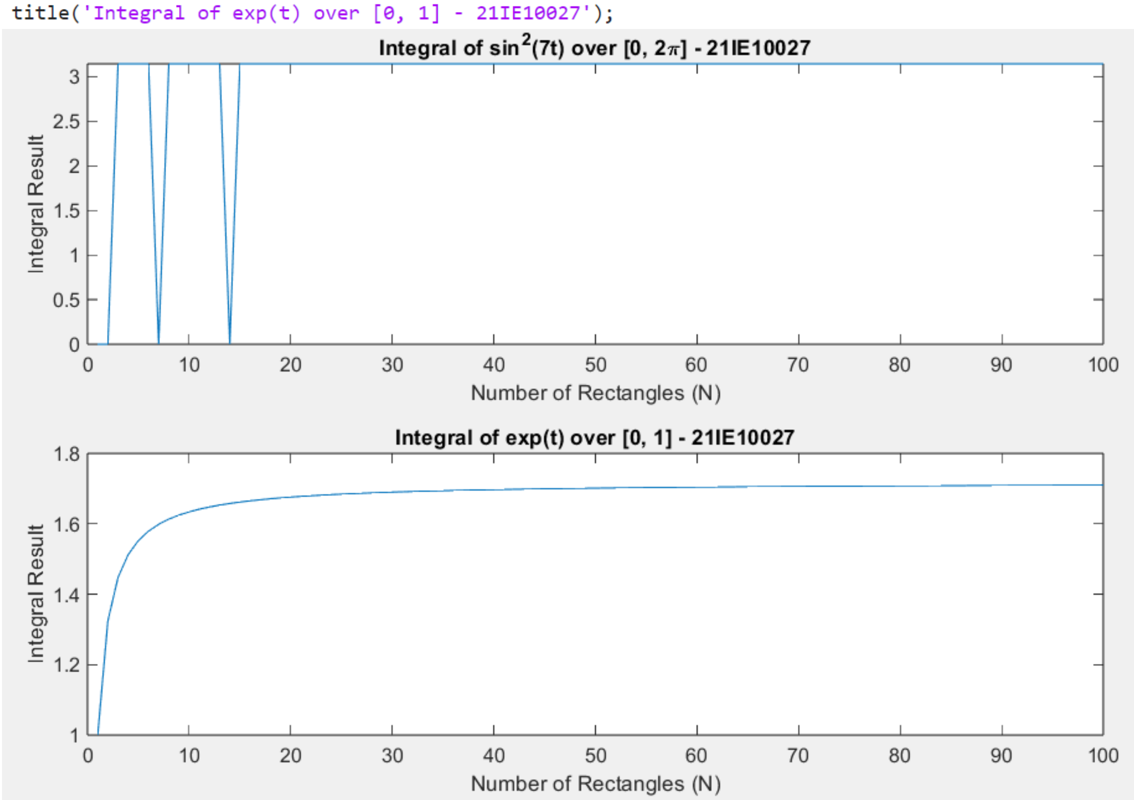
2. $\int_0^{2\pi} \sin^2(7t) \, dt = \frac{1}{2} \int_0^{2\pi} (1 - \cos(14t)) \, dt$
$= \frac{1}{2} \left[t - \frac{1}{14} \sin(14t)\right]_0^{2\pi}$
$= \frac{1}{2} \left(2\pi - \frac{1}{14} \sin(28\pi) - 0 + \frac{1}{14} \sin(0)\right)$
$= \frac{1}{2} (2\pi - 0)$
$= \pi = 3.141592$

1.4 Numerical Computation of Continuous-Time Signals

I(N) is Matlab function for numerically computing the integral of the function $sin^2(7t)$ over the interval [0, 2] where I is the result and N is the number of rectangles used to approximate the integral.
J(N) is Matlab function for numerically computing the integral of the function $exp(t)$ over the interval [0, 1] where I is the result and N is the number of rectangles used to approximate the integral.
The MATLAB code and plots for both I(N) and J(N) were attached below.

```
function I = integ1(N)
    t = linspace(0, 2*pi, N+1);
    % Divide the interval [0, 2*pi] into N equal parts
    dt = t(2) - t(1);
    % Width of each rectangle

    % Compute the heights of the rectangles using sin^2(7t)
    heights = sin(7*t).^2;

    % Numerical approximation of the integral
    % using the sum of areas of rectangles
    I = sum(heights(1:end-1)) * dt;
end
function J = integ2(N)
    t = linspace(0, 1, N+1);
    % Divide the interval [0, 1] into N equal parts
    dt = t(2) - t(1);
    % Width of each rectangle

    % Compute the heights of the rectangles using exp(t)
    heights = exp(t);

    % Numerical approximation of the integral
    % using the sum of areas of rectangles
    J = sum(heights(1:end-1)) * dt;
end
```

```
% Initialize an empty vector to store the results
results1 = zeros(1, 100);
results2 = zeros(1, 100);
% Evaluate the integrals for different values of N
for N = 1:100
    results1(N) = integ1(N);
    results2(N) = integ2(N);
end
subplot(2,1,1);
plot(1:100, results1); % Plot the results for integ1
xlabel('Number of Rectangles (N)');
ylabel('Integral Result');
title('Integral of sin^2(7t) over [0, 2\pi] - 21IE10027');
subplot(2,1,2);
plot(1:100, results2); % Plot the results for integ2
xlabel('Number of Rectangles (N)');
ylabel('Integral Result');
title('Integral of exp(t) over [0, 1] - 21IE10027');
```



The value of both the integrals are equal to the value of I(N) and J(N) respectively when N i.e. the number of rectangles are tending to infinity.

The value of I(7) is zero because this corresponds to using 7 rectangles (N = 7) to approximate the integral. Given that the $sin^2(7t)$ is periodic with a period of $\pi/7$, using 7 rectangles to cover the interval [0, 2] results in each rectangle aligning with the troughs of the sine wave. Since the value of the $sin^2(7t)$ is 0 at these points, the sum of their areas will be 0. similarly for I(14).

2 Special Functions

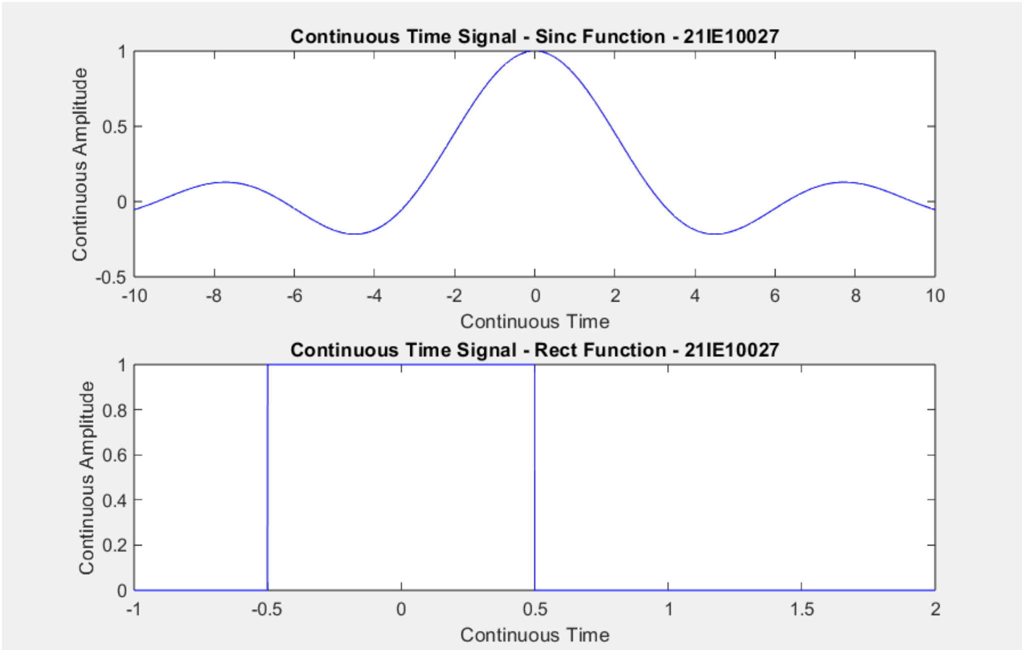
2.1 Sinc Function and Rect Function

$$x_3(t) = \begin{cases} \dfrac{\sin \pi t}{\pi t}, t \neq 0 \\ 1, t = 0 \end{cases} \; for\ t \in [-10,10]$$

$$x_4(t) = rect(t)\ for\ t \in [-1,2]$$

The above two continuous time functions are plotted below over the specified intervals.

```
t3 = -10:0.1:10;
x3 = sin(t3)./t3; % Sinc Function
subplot(2,1,1);
plot(t3,x3,'b'); % Continuous Signal Plotting
xlabel("Continuous Time");
ylabel("Continuous Amplitude");
title("Continuous Time Signal - Sinc Function - 21IE10027");
t4 = -1:0.001:2;
x4 = (t4>=-0.5)-(t4>=0.5); % Rect function
subplot(2,1,2);
plot(t4,x4,'b'); % Continuous Signal Plotting
xlabel("Continuous Time");
ylabel("Continuous Amplitude");
title("Continuous Time Signal - Rect Function - 21IE10027");
```

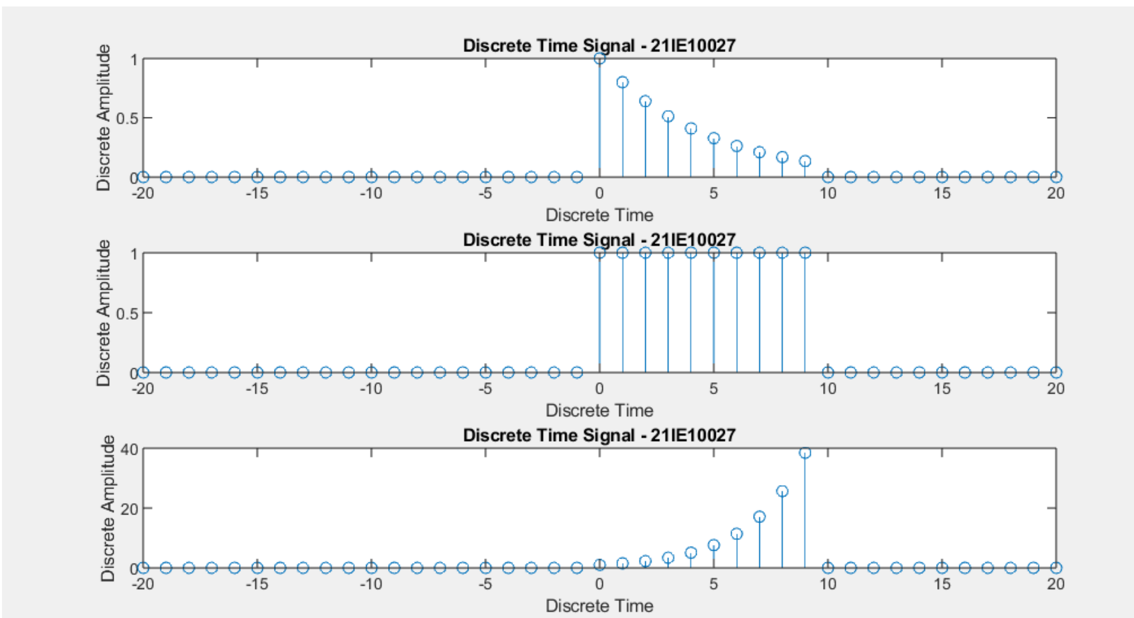


2.2 Bounded Exponential Signals

x[n] $= a^n(u[n] - u[n-10]), \quad n \in [-20, 20]$

The discrete time function x[n] is plotted below for three different values of a such that a=0.8, a=1.0 and a=1.5. The MATLAB code also given for respective plots.

```
n = -20:20;
a1 = 0.8; a2 = 1.0; a3 = 1.5; % Three different values of a
x1 = (a1.^n).*((n>=0)-(n>=10));
x2 = (a2.^n).*((n>=0)-(n>=10));
x3 = (a3.^n).*((n>=0)-(n>=10));
subplot(3,1,1); stem(n,x1); % Plotting function for a = 0.8
xlabel("Discrete Time");
ylabel("Discrete Amplitude");
title("Discrete Time Signal - 21IE10027");
subplot(3,1,2); stem(n,x2); % Plotting function for a = 1.0
xlabel("Discrete Time");
ylabel("Discrete Amplitude");
title("Discrete Time Signal - 21IE10027");
subplot(3,1,3); stem(n,x3); % Plotting function for a = 1.5
xlabel("Discrete Time");
ylabel("Discrete Amplitude");
title("Discrete Time Signal - 21IE10027");
```

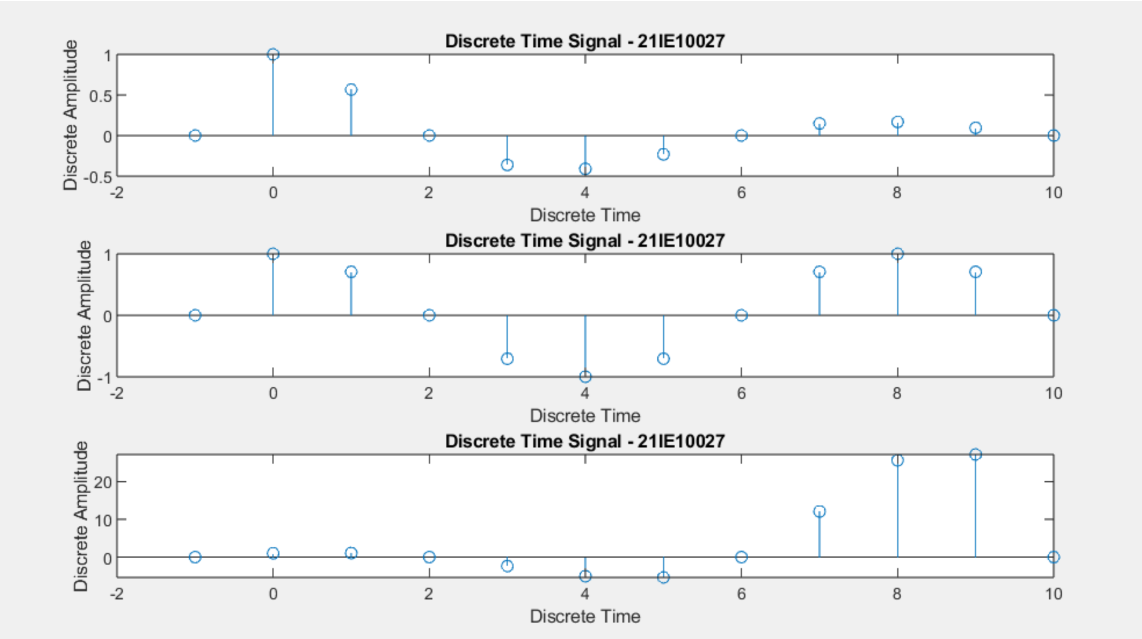


2.3 Exponential Cosine Signals

x[n] $= a^n \cos(wn)u[n], \quad n \in [-1, 10]$

The discrete time function x[n] is plotted below for three different values of a such that a=0.8, a=1.0 and a=1.5. The MATLAB code also given for respective plots.

```
n = -1:10;
w = pi/4;
a1 = 0.8; a2 = 1.0; a3 = 1.5; % Three different values of a
x1 = (a1.^n).*cos(w*n).*(n>=0);
x2 = (a2.^n).*cos(w*n).*(n>=0);
x3 = (a3.^n).*cos(w*n).*(n>=0);
subplot(3,1,1); stem(n,x1); % Plotting function for a = 0.8
xlabel("Discrete Time");
ylabel("Discrete Amplitude");
title("Discrete Time Signal - 21IE10027");
subplot(3,1,2); stem(n,x2); % Plotting function for a = 1.0
xlabel("Discrete Time");
ylabel("Discrete Amplitude");
title("Discrete Time Signal - 21IE10027");
subplot(3,1,3); stem(n,x3); % Plotting function for a = 1.5
xlabel("Discrete Time");
ylabel("Discrete Amplitude");
title("Discrete Time Signal - 21IE10027");
```

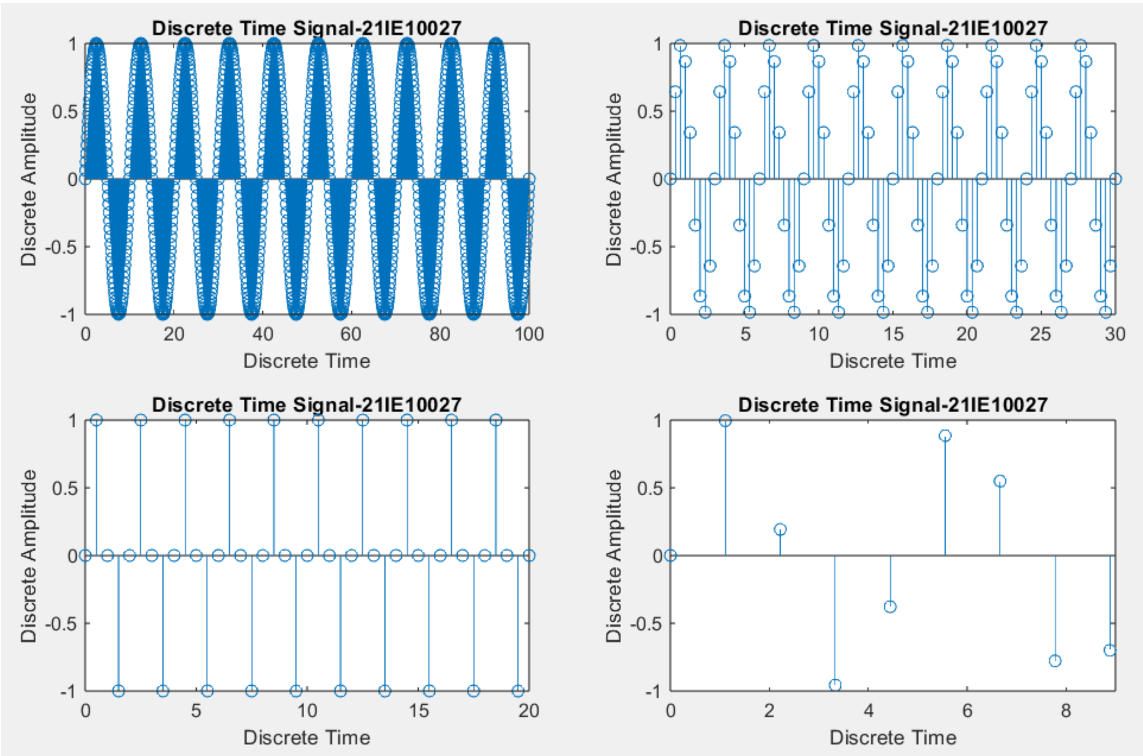


3 Sampling

The MATLAB code and plots are attached below for the function x(n) sampled with different time intervals.

$x(n) = f(T_s n) = sin(2\pi T_s n)$

```
% Plot 1
Ts1 = 1/10; n1 = 0:Ts1:100; x1 = sin(2*pi*Ts1*n1);
subplot(2,2,1); stem(n1,x1);
xlabel("Discrete Time");
ylabel("Discrete Amplitude");
title("Discrete Time Signal-21IE10027");
axis([0 100 -1 1]);
% Plot 2
Ts2 = 1/3; n2 = 0:Ts2:30; x2 = sin(2*pi*Ts2*n2);
subplot(2,2,2); stem(n2,x2);
xlabel("Discrete Time");
ylabel("Discrete Amplitude");
title("Discrete Time Signal-21IE10027");
axis([0 30 -1 1]);
% Plot 3
Ts3 = 1/2; n3 = 0:Ts3:20; x3 = sin(2*pi*Ts3*n3);
subplot(2,2,3); stem(n3,x3);
xlabel("Discrete Time");
ylabel("Discrete Amplitude");
title("Discrete Time Signal-21IE10027");
axis([0 20 -1 1]);
% Plot 4
Ts4 = 10/9; n4 = 0:Ts4:9; x4 = sin(2*pi*Ts4*n4);
subplot(2,2,4); stem(n4,x4);
xlabel("Discrete Time");
ylabel("Discrete Amplitude");
title("Discrete Time Signal-21IE10027");
axis([0 9 -1 1]);
```



With a smaller sampling period like $T_s = 1/10$, the signal is sampled more frequently. This results in a more accurate representation of the original continuous signal. The discrete signal follows the shape of the original sine wave more closely. But in the case of larger sampling period like $T_s = 1/3, T_s = 1/2, T_s = 10/9$, the discrete signal might lose significant details of the original sine wave and the signal may not capture the fast oscillations of the sine wave accurately.