INSTITUTE OF ENGINEERING

ADVANCED COLLEGE OF ENGINEERING AND MANAGEMENT

KALANKI, KATHMANDU

(AFFILIATED TO TRIBHUVAN UNIVERSITY)

LAB REPORT

SUBJECT: DIGITAL SIGNAL ANALYSIS AND PROCESSING (DSAP)

LAB NO: 02

SUBMITTED BY:

NAME: DIPESH DHUNGANA

ROLL NO: ACE077BCT035

DATE: June 07, 2024

SUBMITTED TO:

DEPARTMENT OF COMPUTER AND ELECTRONICS

TITLE: ELEMENTARY SIGNALS

OBJECTIVE:

TO CONSTRUCT DIFFERENT TYPES OF ELEMENTARY SIGNAL.

THEORY:

Elementary signals are fundamental building blocks in digital signal analysis and processing. Understanding these signals is crucial as they form the basis for more complex signal manipulations and analyses. The key elementary signals include the unit step function, rectangular pulse, ramp function, triangular pulse, and impulse function.

Unit Step Function

The unit step function is a discrete signal that switches from 0 to 1 at a specified time. The unit step function is used to represent signals that start at a particular point in time and continue indefinitely.

2. Rectangular Pulse

A rectangular pulse is a signal that is 1 for a specified interval and 0 elsewhere. Rectangular pulses are used in digital communications and signal processing for windowing functions and as a model for pulse signals.

3. Ramp Function

The ramp function is a signal that increases linearly with time. Ramp functions are used in systems to model signals that increase at a constant rate.

4. Triangular Pulse

A triangular pulse is a piecewise linear function that forms a triangle shape. It can be represented as the convolution of two rectangular pulses. Triangular pulses are used in digital filtering and interpolation.

5. Impulse Function

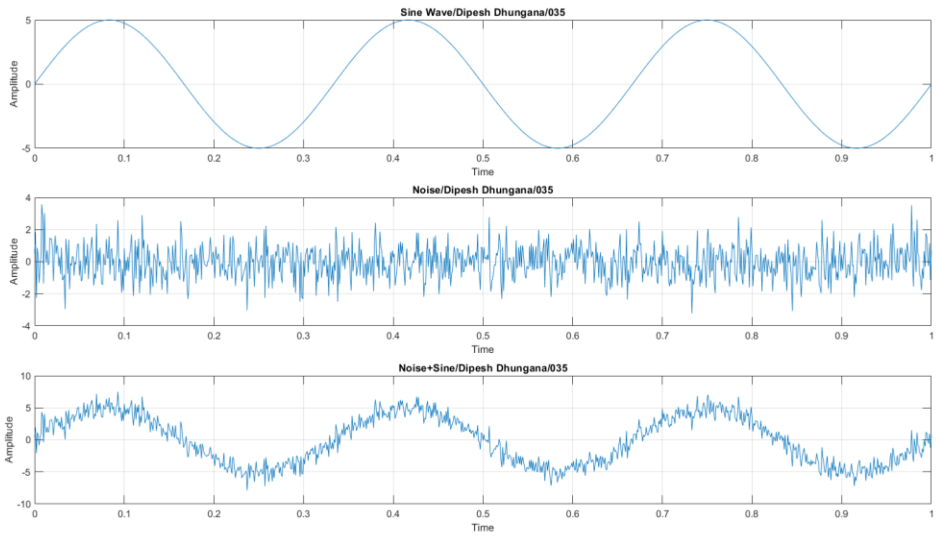
The impulse function/delta, also known as the Dirac delta function is a signal that is zero everywhere except at zero, where it is infinitely high and has an integral of one.

Q1. Construct a noise wave, now add a noise to the sine wave then plot all these three signals in one signal frame.

Source code (Sine Wave with Noise)

```
clc;
close all;
clear all;
a=5'
f=3;
t=0:0.001:1;
x=a*sin(2*pi*f*t);
subplot(3,1,1);
plot(t,x);
xlabel('Time');
ylabel('Amplitude');
title('Sine Wave/Dipesh Dhungana/035');
grid on;
z=randn(1,length(t));
subplot(3,1,2);
plot(t,z);
xlabel('Time');
ylabel('Amplitude');
title('Noise/Dipesh Dhungana/035');
grid on;
y=z+x;
subplot(3,1,3);
plot(t,y);
xlabel('Time');
ylabel('Amplitude');
title('Noise+Sine/Dipesh Dhungana/035');
grid on;
```

Output (Sine Wave with Noise)

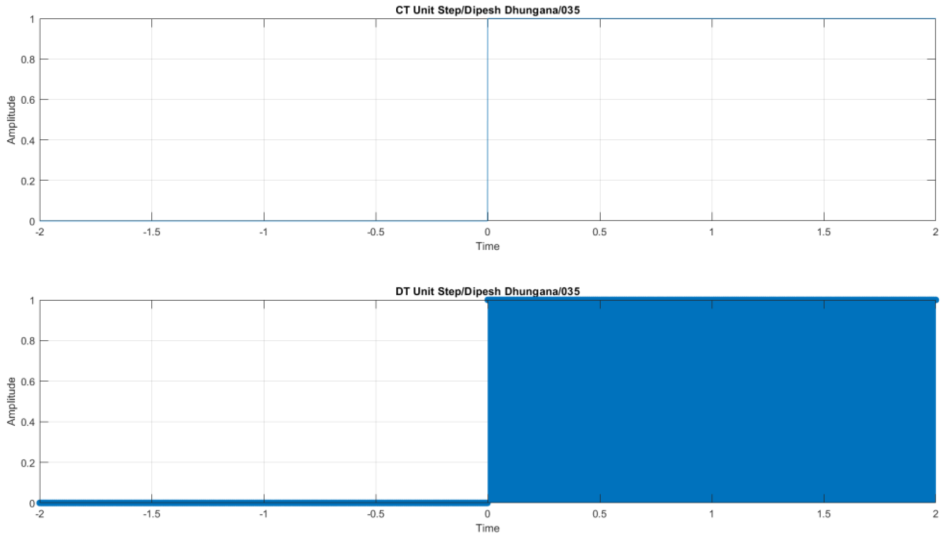


Q2. Construct both continuous and discrete unit step signal.

Source code (Continuous and Discrete Unit Step)

```
clc;
close all;
clear all;
i=1;
for t=-2:0.001:2
    if(t>=0)
        x(i)=1;
    else
        x(i)=0;
    end;
    i=i+1;
end;
t=-2:0.001:2;
subplot(2,1,1);
plot(t,x);
xlabel('Time');
ylabel('Amplitude');
title('CT Unit Step/Dipesh Dhungana/035');
grid on;
i=1;
for t=-2:0.001:2;
    if(t>=0)
        x(i)=1;
    else
        x(i)=0;
    end;
    i=i+1;
end;
t=-2:0.001:2;
subplot(2,1,2);
stem(t,x);
xlabel('Time');
ylabel('Amplitude');
title('DT Unit Step/Dipesh Dhungana/035');
grid on;
```

Output (Continuous and Discrete Unit Step)



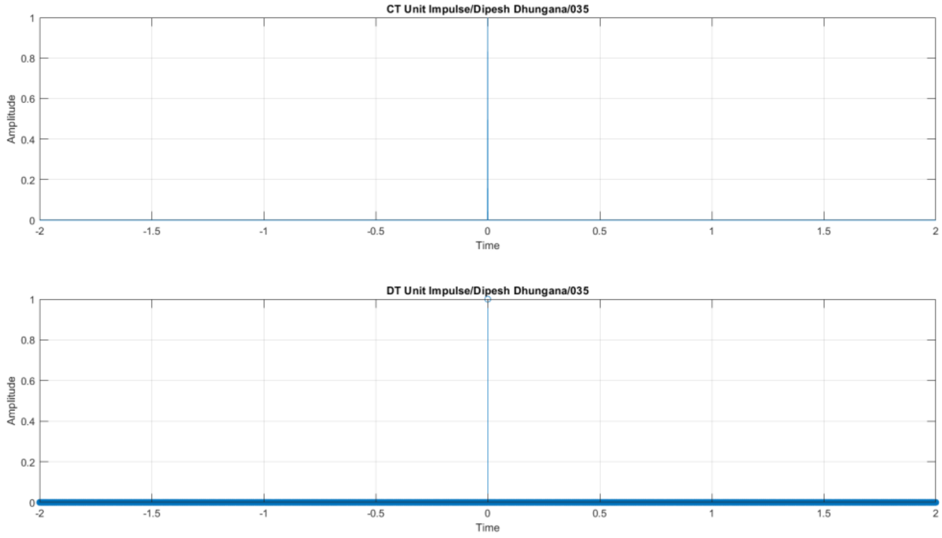
Q3. Construct both continuous and discrete unit impulse signal.

Source code (Continuous and Discrete Unit Impulse)

```
clc;
close all;
clear all;
i=1;
for t=-2:0.001:2
    if(t==0)
        x(i)=1;
    else
        x(i)=0;
    end;
    i=i+1;
end;
t=-2:0.001:2;
subplot(2,1,1);
plot(t,x);
xlabel('Time');
ylabel('Amplitude');
title('CT Unit Impulse/Dipesh Dhungana/035');
grid on;
i=1;
for t=-2:0.001:2;
    if(t==0)
        x(i)=1;
    else
        x(i)=0;
    end;
    i=i+1;
end;
t=-2:0.001:2;
subplot(2,1,2);
stem(t,x);
xlabel('Time');
```

```
ylabel('Amplitude');
title('DT Unit Impulse/Dipesh Dhungana/035');
grid on;
```

<u>Output (Continuous and Discrete Unit Impulse)</u>



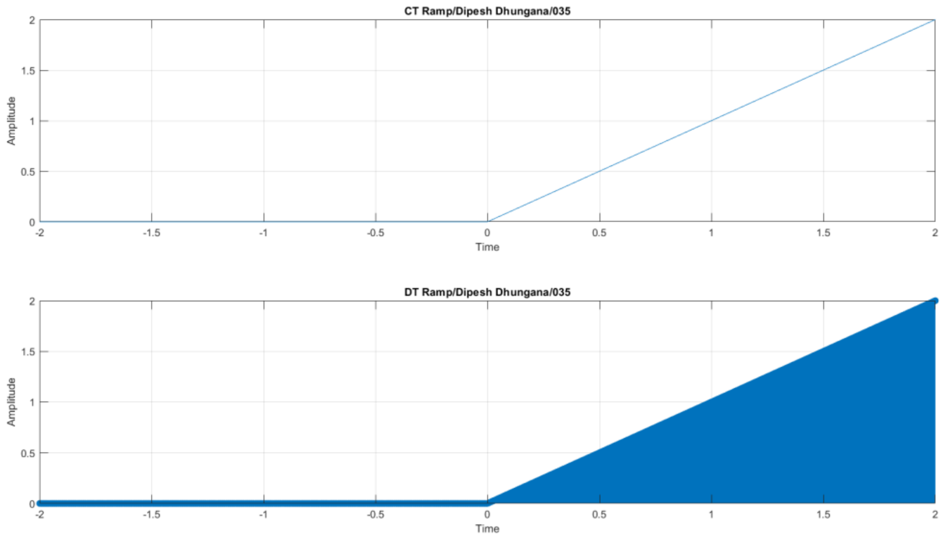
Q4. Construct both continuous and discrete ramp signal.

<u>Source code (Continuous and Discrete Ramp)</u>

```
clc;
close all;
clear all;
i=1;
for t=-2:0.001:2
    if(t>=0)
        x(i)=t;
    else
        x(i)=0;
    end;
    i=i+1;
end;
t=-2:0.001:2;
subplot(2,1,1);
plot(t,x);
xlabel('Time');
ylabel('Amplitude');
title('CT Ramp/Dipesh Dhungana/035');
grid on;
i=1;
for t=-2:0.001:2;
    if(t>=0)
        x(i)=t;
    else
        x(i)=0;
    end;
    i=i+1;
end;
t=-2:0.001:2;
```

```
subplot(2,1,2);
stem(t,x);
xlabel('Time');
ylabel('Amplitude');
title('DT Ramp/Dipesh Dhungana/035');
grid on;
```

Output (Continuous and Discrete Ramp)



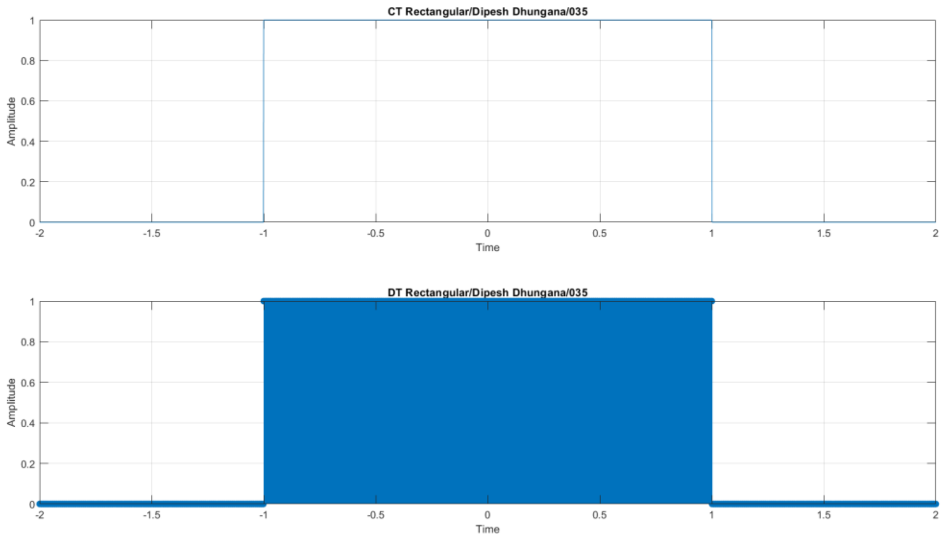
Q5. Construct both continuous and discrete rectangular signal.

Source code (Continuous and Discrete Rectangular)

```
clc;
close all;
clear all;
pulse_width = 2;
i = 1;
for t = -2:0.001:2
    if(abs(t) > pulse_width/2)
        x(i) = 0;
    else
        x(i) = 1;
    end
    i = i + 1;
end
t = -2:0.001:2;
subplot(2,1,1);
plot(t, x);
xlabel('Time');
ylabel('Amplitude');
title('CT Rectangular/Dipesh Dhungana/035');
grid on;

i = 1;
for t = -2:0.001:2
    if(abs(t) > pulse_width/2)
        x(i) = 0;
```

```
    else
        x(i) = 1;
    end
    i = i + 1;
end
t = -2:0.001:2;
subplot(2,1,2);
stem(t, x);
xlabel('Time');
ylabel('Amplitude');
title('DT Rectangular/Dipesh Dhungana/035');
grid on;
```

<u>Output (Continuous and Discrete Rectangular)</u>



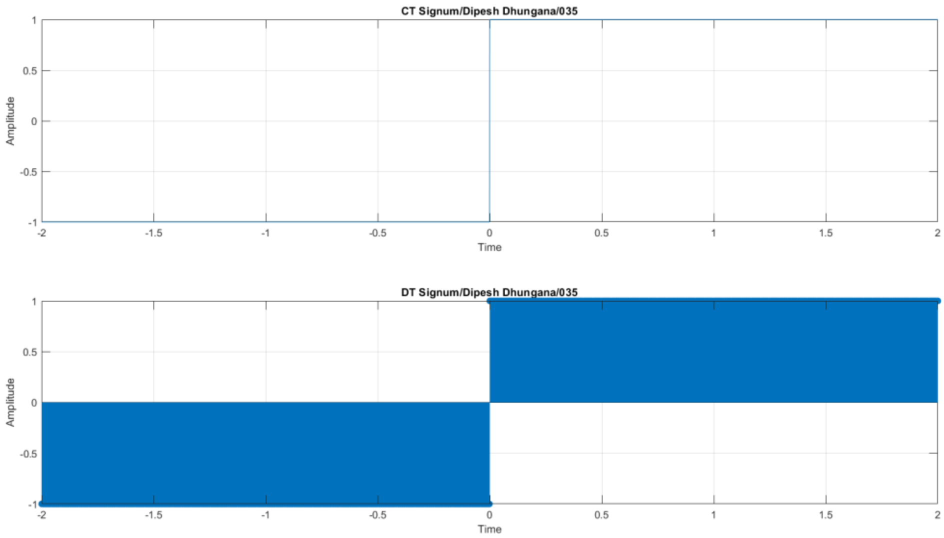
Q6. Construct both continuous and discrete signum signal.

<u>Source code (Continuous and Discrete Signum)</u>

```
clc;
close all;
clear all;
i=1;
for t=-2:0.001:2
    if(t>=0)
        x(i)=1;
    else if(t<0)
        x(i)=-1;
    else
        x(i)=0;
    end;
    end;
    i=i+1;
end;
t=-2:0.001:2;
subplot(2,1,1);
plot(t,x);
xlabel('Time');
```

```
ylabel('Amplitude');
title('CT Signum/Dipesh Dhungana/035');
grid on;
i=1;
for t=-2:0.001:2
    if(t>=0)
        x(i)=1;
    else if(t<0)
        x(i)=-1;
    else
        x(i)=0;
    end;
    end;
    i=i+1;
end;
t=-2:0.001:2;
subplot(2,1,2);
stem(t,x);
xlabel('Time');
ylabel('Amplitude');
title('DT Signum/Dipesh Dhungana/035');
grid on;
```

Output (Continuous and Discrete Signum)



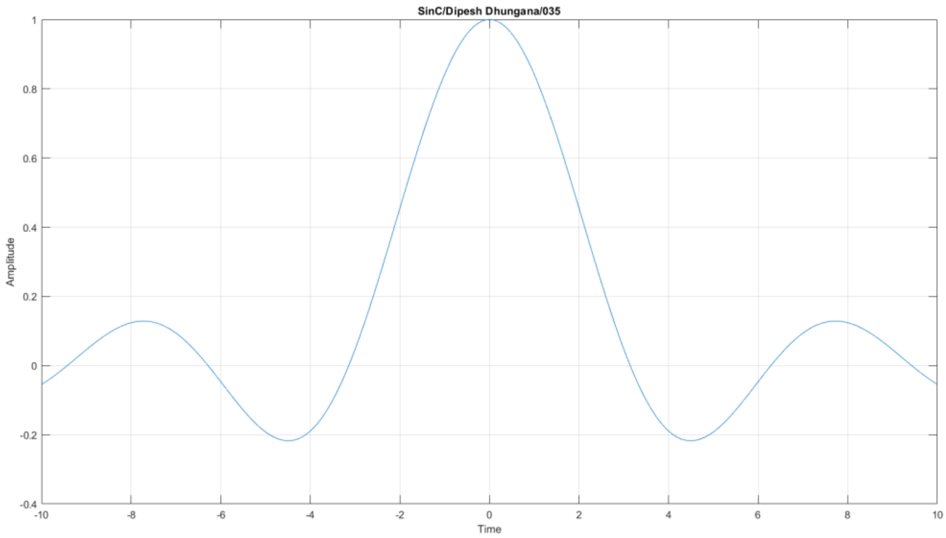
Q7. Construct SinC signal.

Source code (SinC)

```
clc;
close all;
clear all;
t=-10:0.0001:10;
y=sin(t)./t;
plot(t,y);
xlabel('Time');
ylabel('Amplitude');
```

```
title('SinC/Dipesh Dhungana/035');
grid on;
```

Output (SinC)



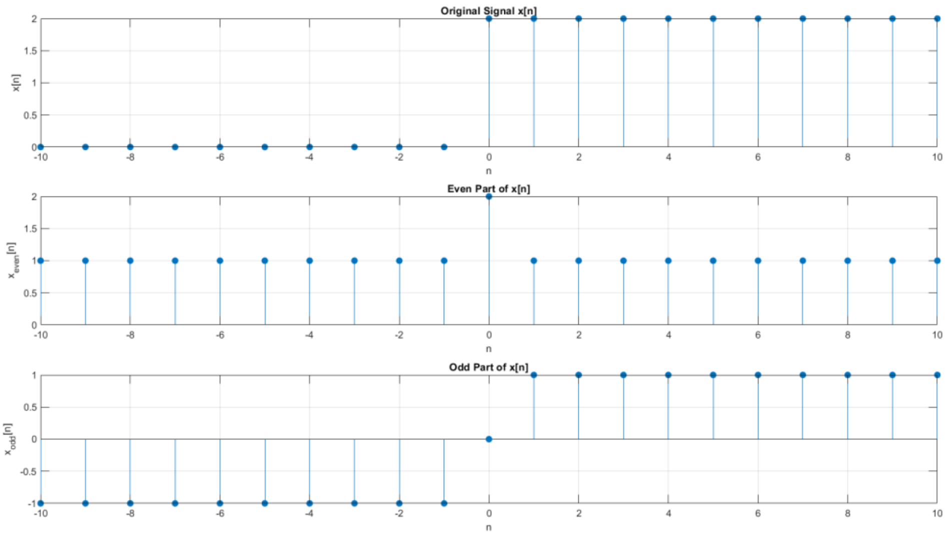
Q8. How do you decompose a discrete-time signal into its even and odd parts?

Solution

```
clc;
close all;
clear all;

% Define the discrete-time signal
n = -10:10; % Define a range for n
x = zeros(size(n));
x(n >= 0) = 2;

% Compute the even part of the signal
x_even = 0.5 * (x + fliplr(x));

% Compute the odd part of the signal
x_odd = 0.5 * (x - fliplr(x));

% Plot the original signal
subplot(3,1,1);
stem(n, x, 'filled');
xlabel('n');
ylabel('x[n]');
title('Original Signal x[n]');
grid on;

% Plot the even part of the signal
subplot(3,1,2);
stem(n, x_even, 'filled');
xlabel('n');
ylabel('x_{even}[n]');
title('Even Part of x[n]');
grid on;
```

```
% Plot the odd part of the signal
subplot(3,1,3);
stem(n, x_odd, 'filled');
xlabel('n');
ylabel('x_{odd}[n]');
title('Odd Part of x[n]');
grid on;
```

Output



DISCUSSION AND CONCLUSION:

In this lab, we generated and analyzed various elementary signals using MATLAB, including sine waves with noise, unit step functions, unit impulse functions, ramp functions, rectangular pulses, signum functions, and sinc functions. Each signal was examined to understand its properties and applications. For example, we observed how noise affects a sine wave and the importance of unit step and impulse functions in representing signals that start or change instantaneously.

The lab effectively demonstrated the construction and significance of elementary signals in digital signal processing. Understanding these signals is crucial for more advanced signal analysis and processing tasks. The exercises reinforced theoretical concepts and provided practical experience in signal generation and analysis, highlighting the foundational role of these elementary signals in the field.