Signal Analysis using MATLAB

Lab Exercises on March 1, 2021

Department of Electronics and Computer Engineering
Pulchowk Campus, Lalitpur

Ashlesh Pandey

PUL074BEX007

1 Some MATLAB Commands

Command	General Description	Argument Description
hold	Retain current plot when adding new plots.	• *hold on*: Retains plots in the current axes. • *hold off*: Sets the hold state to off so that new plots added to the axes clear existing.
plot	Creates a 2-D line plot.	$plot(X, Y)$: Plot data in Y versus the corresponding values in X.
subplot	Create axes in tiled positions.	$subplot(m, n, p)$: Divides the current figure into an m-by-n grid and creates axes in the position specified by p.
stem	Plot discrete sequence data	$stem(Y)$: Plots the data sequence, Y, as stems that extend from a baseline along the x-axis.
title	Add title	$title(titletext)$: Uses titletext as the title string for the current axes or standalone visualization.
xlabel	Label for X-axis.	$xlabel(text)$: Uses text as the xlabel string.
ylabel	Label for Y-axis.	$ylabel(text)$: Uses text as the ylabel string.
input	Request user input.	$input(prompt)$: Displays the text in prompt and waits for the user to input a value.
sin	Sine of argument in radians.	$sin(X)$: Sine of X.
cos	Cosine of argument in radians.	$cos(X)$: Cosine of X.
power	Element-wise power.	$power(A, B)$: Raises each element of A to the corresponding powers in B.
conv	Convolution and polynomial multiplication.	$conv(u, v)$: Returns the convolution of u and v.
fft	Fast Fourier transform.	$fft(X)$: computes the discrete Fourier transform (DFT) of X using a fast Fourier transform (FFT) algorithm.
freqz	Frequency response of digital filter.	$[h, w] = freqz(b, a, n)$: Returns the n-point frequency response vector h and the corresponding angular frequency vector w for the digital filter with transfer function coefficients stored in b and a.
abs	Absolute value and complex magnitude.	$abs(X)$: Returns the absolute value of each element in array X. If X is complex, it returns the complex magnitude.
angle	Phase angle.	$angle(z)$: Returns the phase angle in the interval $[-\pi, \pi]$ for each element of a complex array z.

Table 1: Some MATLAB commands used during the lab experiment

2 Basic Signal Visualization

In simple terms, signal is a set of data or information such that they represent the behavior of certain phenomena as a function of one or multiple independent variables. A signal is simply the dependent variable changing as a function of independent variable.

Analog/Continuous Signal

If the independent variable attains continuous values, then the signal is called analog/continuous signal. Throughout our course, analog signals have been regarded as function of time, but they aren't limited to this. In general, they are denoted as $x(t)$, $y(t)$, $h(t)$, and so on such that time (t) is the independent variable and the signals are a function of t.

Digital/Discrete Signal

If the independent variable attains only discrete values, then the signal is called digital/discrete signal. Discrete signals are represented as a function of an integer number, say n. In general, they are denoted as, $x[n]$, $y[n]$, $h[n]$, and so on such that n is the independent variable and the signals are a function of n.

2.1 Sinosoidal Signal

The sine and cosine signals were plotted in the same co-ordinate axes. This kind of visualization is key to compare signals and their behaviors. The sin and cos functions in MATLAB return the corresponding sine and cosine values for the input argument. The use of hold state allows the figure to retain the previous plots.

```
hold on;
t=-10:0.01:10;
y=sin(t);
plot(t,y,'r');
z=cos(t);
plot(t,z,'k');
xlabel('t');
ylabel('x(t), y(t)');
title('Visualization of Continuous Time
    Signals, x(t)=sin(t) and y(t)=cos(t)');
hold off;
```

Listing 1: Matlab script for visualization of continuous time signals $x(t) = sin(t)$ and $y(t) = cos(t)$

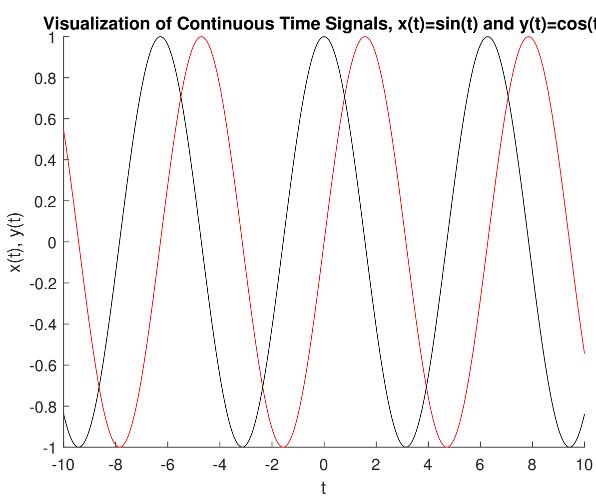

Figure 1: Obtained plot for $x(t) = sin(t)$ and $y(t) = cos(t)$

2.2 Ramp Signal

Continuous time ramp signal $y(t) = mt$ and discrete time ramp signal $y[n] = mn$ were visualized using MATLAB. The value of m is taken as user input using the input command. The value of m during the execution was selected as 2. The stem function allows us to plot discrete time signals in MATLAB.

```
t=-10:0.01:10;
n=-10:10;
m = input('Enter the value of m\n');
y=m*t;
z=m*n;
subplot(2,1,1);
plot(t,y);
xlabel('t');
ylabel('y(t)');
title('Continuous Ramp Signal');
subplot(2,1,2);
stem(n,z);
xlabel('n');
ylabel('y[n]');
title('Discrete Ramp Signal');
```

Listing 2: Matlab script for visualization of continuous time $y(t) = mt$ and discrete time $y[n] = mn$ ramp signals

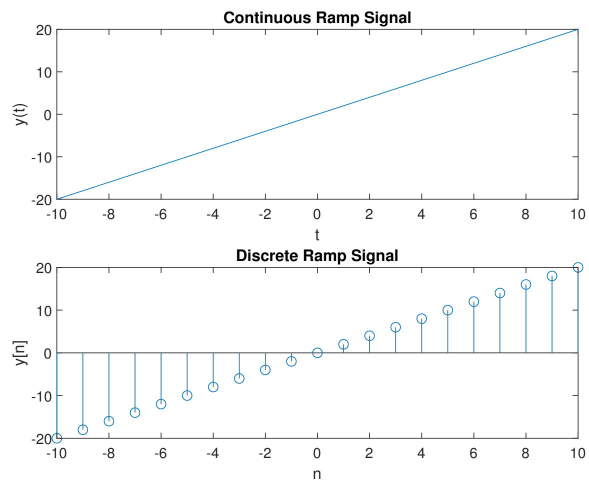

Figure 2: Obtained plot for $y(t) = mt$ and $y[n] = mn$ for $m = 2$

2.3 Exponential Signal

Continuous time exponential signal $y(t) = ca^t$ and discrete time exponential signal $y[n] = ca^n$ were visualized using MATLAB. The values of c and a were taken as user input.

```
t=-10:0.01:10;
n=-10:10;
c = input('Enter the value of c\n');
a= input('Enter the value of a\n');
y=power(c, a*t);
z=power(c, a*n);
subplot(2,1,1);
plot(t,y);
xlabel('t');
ylabel('y(t)');
title('Continuous Signal y(t)=c^{at}');
subplot(2,1,2);
stem(n,z);
xlabel('n');
ylabel('y[n]');
title('Discrete Signal y[n]=c^{an}');
```

Listing 3: Matlab script for visualization of continuous time $y(t) = ca^t$ and discrete time $y[n] = ca^n$ exponential signals

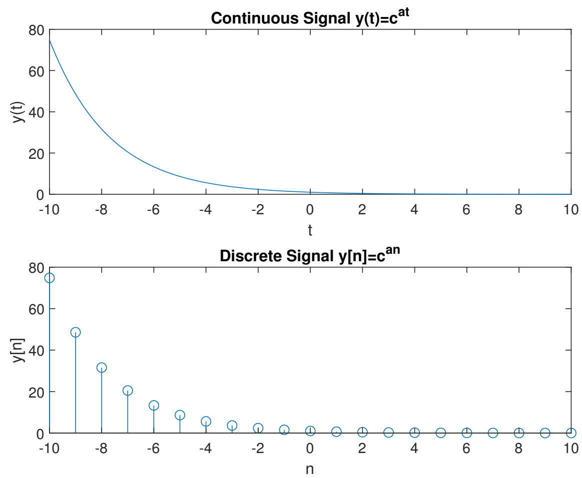

Figure 3: Obtained plot for $y(t) = ca^t$ and $y[n] = ca^n$ for $c = 0.75$ and $a = 1.5$

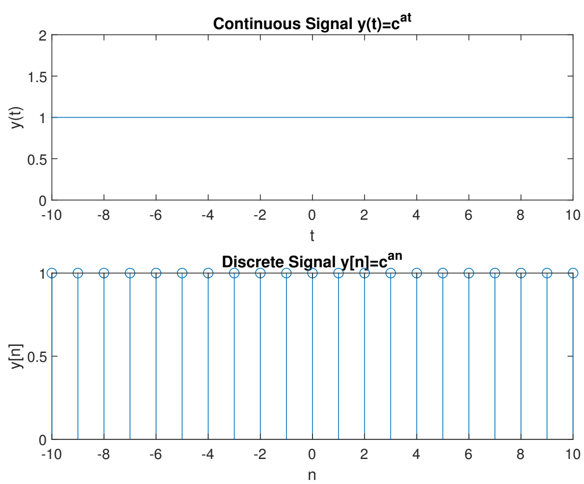

Figure 4: Obtained plot for $y(t) = ca^t$ and $y[n] = ca^n$ for $c = 0.75$ and $a = 0$

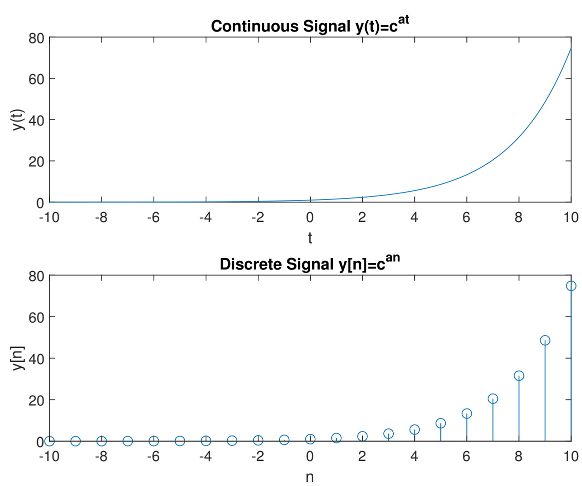

Figure 5: Obtained plot for $y(t) = ca^t$ and $y[n] = ca^n$ for $c = 0.75$ and $a = -1.5$

Likewise continuous time exponential signal $y(t) = ce^{at}$ and discrete time exponential signal $y[n] = ce^{an}$ were visualized using MATLAB. The values of c and a were taken as user input.

```
t=-10:0.01:10;
n=-10:10;
c = input('Enter the value of c\n');
a= input('Enter the value of a\n');
y=c*exp(a*t);
z=c*exp(a*n);
subplot(2,1,1);
plot(t,y);
xlabel('t');
ylabel('y(t)');
title('Continuous Exponential Signal y(t)=c.e^{at}');
subplot(2,1,2);
stem(n,z);
xlabel('n');
ylabel('y[n]');
title('Discrete Exponential Signal y[n]=c.e^{an}');
```

Listing 4: Matlab script for visualization of continuous time $y(t) = ce^{at}$ and discrete time $y[n] = ce^{an}$ exponential signals

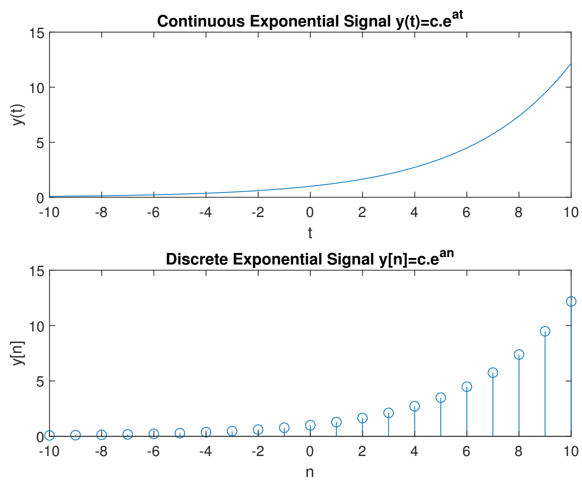

Figure 6: Obtained plot for $y(t) = ce^{at}$ and $y[n] = ce^{an}$ for $c = 1$ and $a = 0.25$

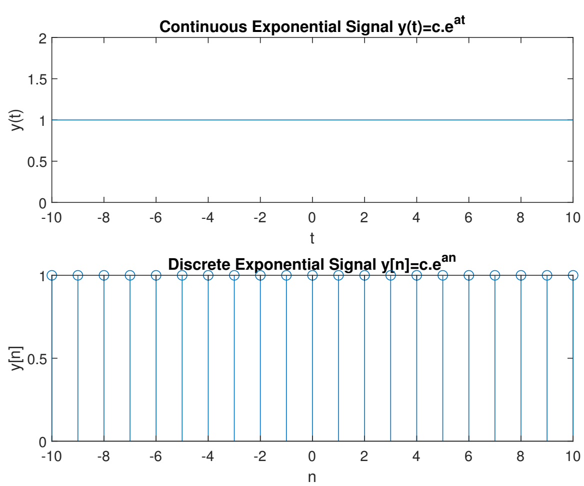

Figure 7: Obtained plot for $y(t) = ce^{at}$ and $y[n] = ce^{an}$ for $c = 1$ and $a = 0$

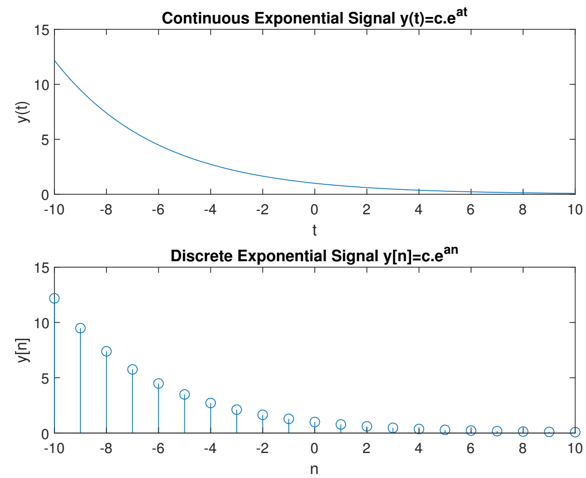

Figure 8: Obtained plot for $y(t) = ce^{at}$ and $y[n] = ce^{an}$ for $c = 1$ and $a = -0.25$

2.4 Unit Step Signal

Continuous time unit step signal $y(t) = u(t)$ and discrete time ramp signal $y[n] = u[n]$ were visualized using MATLAB. It takes a value 1 for $t >= 0$ or $n >= 0$ and 0 otherwise.

```
t=-10:0.01:10;
y=zeros(size(t));
y(t>=0)=1;
subplot(2,1,1);
plot(t,y);
xlabel('t');
ylabel('y(t)');
title('Continuous Unit Step Function');
subplot(2,1,2);
hold on;
for n=-10:10
    if(n<0)
        stem(n,0);
    else
        stem(n,1);
    end
end
xlabel('n');
ylabel('y[n]');
title('Discrete Unit Step Function');
hold off;
```

Listing 5: Matlab script for visualization of continuous time $y(t) = u(t)$ and discrete time $y[n] = u[n]$ unit step signals

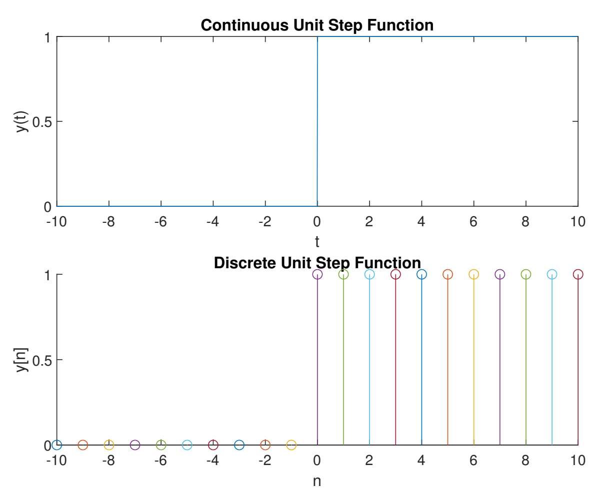

Figure 9: Obtained plot for $y(t) = u(t)$ and $y[n] = u[n]$

2.5 Unit Impulse Signal

Continuous time unit step signal $y(t) = \delta(t)$ and discrete time ramp signal $y[n] = \delta[n]$ were visualized using MATLAB. It takes a value 1 for $t = 0$ or $n = 0$ and 0 otherwise.

```
t=-10:0.01:10;
y=zeros(size(t));
y(t==0)=1;
subplot(2,1,1);
plot(t,y);
xlabel('t');
ylabel('y(t)');
title('Continuous Unit Impulse Function');
subplot(2,1,2);
hold on;
for n=-10:10
    if(n==0)
        stem(n,1);
    else
        stem(n,0);
    end
end
xlabel('n');
ylabel('y[n]');
title('Discrete Unit Impulse Function');
hold off;
```

Listing 6: Matlab script for visualization of continuous time $y(t) = \delta(t)$ and discrete time $y[n] = \delta[n]$ unit step signals

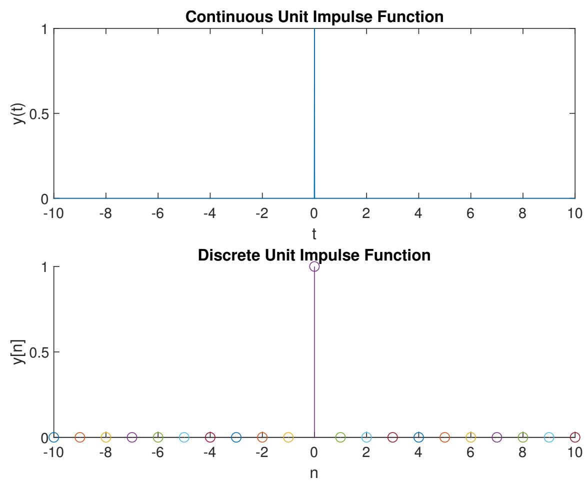

Figure 10: Obtained plot for $y(t) = \delta(t)$ and $y[n] = \delta[n]$

3 Fourier Series

For a periodic continuous time signal $x(t)$ with T period, the fourier series representation is,

$$x(t) = \sum_{k=-\infty}^{\infty} a_k e^{jk\omega_0 t}$$

such that $\omega_0 = \frac{2\pi}{T}$ and the fourier coefficients are given as,

$$a_k = \int_T x(t) e^{-j\omega_0 t} dt$$

Likewise, the fourier series representation of a square wave with period T and amplitude a is mathematically given as,

$$x(t) = \frac{4a}{\pi} \sum_{k=1}^{\infty} \frac{sin((2k-1)\omega_0 t)}{2k-1}$$

The sum of all the odd harmonics of the sinosoidal signal forms a square wave approximation.

```
x=0;
t=-10:0.01:10;
a=40;
w=5;
n=input('Enter the value of n\n');
for i=1:2:n
    x=x+(4*(a/pi))*(sin(i*w*t)*1/i);
end
plot(t,x)
xlabel('t');
ylabel('x(t)');
title('Fourier Series Representation of a Square Wave');
```

Listing 7: Matlab script for fourier series representation of a square wave

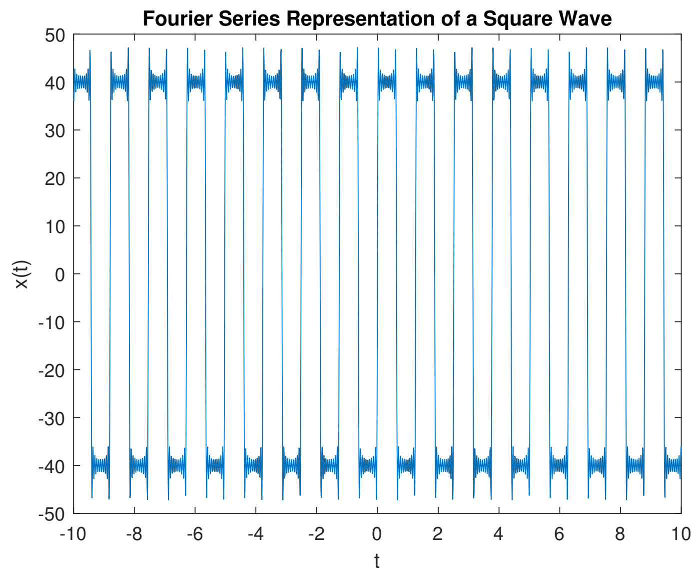

Figure 11: Obtained plot for fourier series of $x(t)$ with first 20 terms

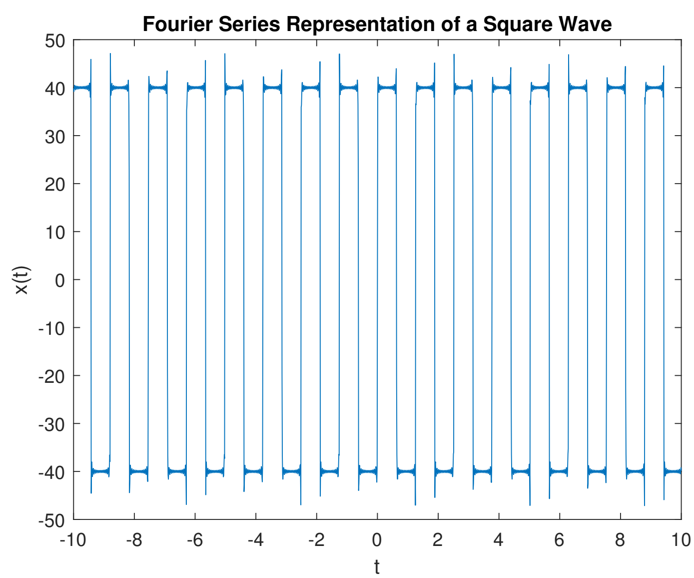

Figure 12: Obtained plot for fourier series of $x(t)$ with first 100 terms

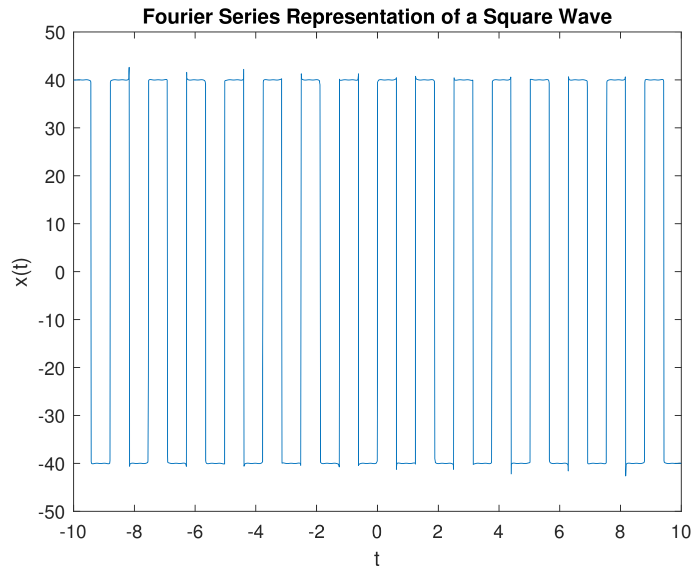

Figure 13: Obtained plot for fourier series of $x(t)$ with first 1000 terms

With the increase in number of terms used, the ripples also increases based on the Gibbs phenomenon.

4 Convolution

The convolution of two continuous time signals $x(t)$ and $h(t)$, called the convolution integral, is mathematically given as,

$$y(t) = x(t) * h(t) = \int_{-\infty}^{\infty} x(u)h(t-u)du$$

Likewise, the convolution of two discrete time signals $x[n]$ and $h[n]$, called the convolution sum, is mathematically given as,

$$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[n]h[n-k]$$

The conv function returns the convolution for two discrete sequences in MATLAB.

```
x=[1,3,2,1,1];
h=[1,2,-1,1];
y=conv(x,h);
subplot(3,1,1);
stem(x);
xlabel('n');
ylabel('x[n]');
title('Input Signal x[n]');
subplot(3,1,2);
stem(h);
xlabel('n');
ylabel('h[n]');
title('Impulse Response Signal h[n]');
subplot(3,1,3);
stem(y);
xlabel('n');
ylabel('y[n]');
title('Output Signal y[n]');
```

Listing 8: Matlab script for convolution for two discrete sequences $x[n] = [1, 3, 2, 1, 1]$ and $h[n] = [1, 2, -1, 1]$

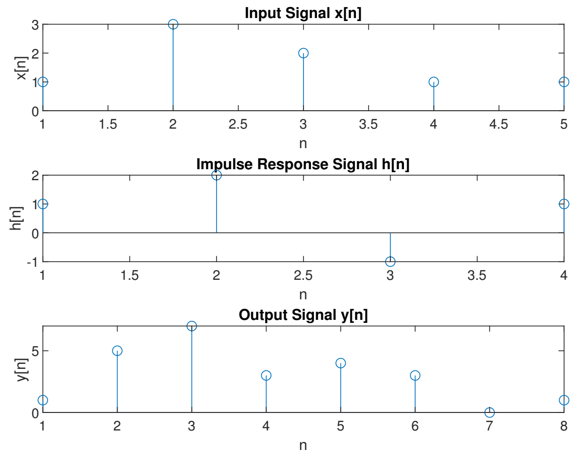

Figure 14: Obtained plot for convolution of $x[n] = [1, 3, 2, 1, 1]$ and $h[n] = [1, 2, -1, 1]$

```
n=0:10;
h=ones(1,11);
x= power(0.5,n);
y=conv(x,h);
subplot(3,1,1);
stem(x);
xlabel('n');
ylabel('x[n]');
title('Input Signal x[n]');
subplot(3,1,2);
stem(h);
xlabel('n');
ylabel('h[n]');
title('Impulse Response Signal h[n]');
subplot(3,1,3);
stem(y);
xlabel('n');
ylabel('y[n]');
title('Output Signal y[n]');
```

Listing 9: Matlab script for convolution for two discrete sequences $x[n] = 0.5^n$ and $h[n] = u[n]$

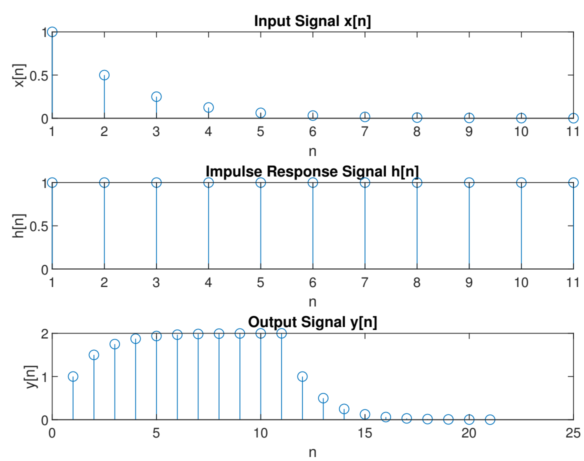

Figure 15: Obtained plot for convolution of $x[n] = 0.5^n$ and $h[n] = u[n]$

5 Fourier Transform

The fourier transform of a continuous time signal $x(t)$ is mathematically given as,

$$X(j\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt$$

Likewise, the fourier transform of a discrete time signal $x[n]$ is mathematically given as,

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}$$

The fft function returns the real and imaginary parts of the fast fourier transform for the input argument. The real and imaginary parts are plotted separately.

```
x=[0,1,2,3];
y=fft(x,4);
subplot(3,1,1);
stem(x);
xlabel('n');
ylabel('x[n]');
title('Discrete Signal x[n]');
subplot(3,1,2);
stem(real(y));
xlabel('n');
ylabel('Re(X(\omega))');
title('Real part of DTFT for x[n]');
subplot(3,1,3);
stem(imag(y));
xlabel('n');
ylabel('Im(X(\omega))');
title('Imaginary part of DTFT for x[n]');
```

Listing 10: Matlab script for fourier transform of discrete sequence $x[n] = [0, 1, 2, 3]$

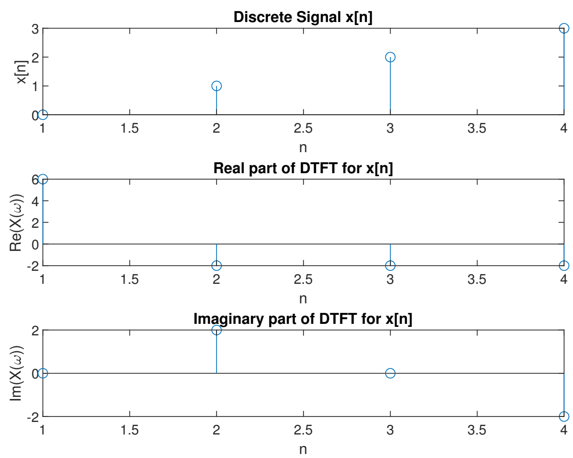

Figure 16: Obtained plot for fourier transform of $x[n] = [0, 1, 2, 3]$

6 Frequency Response of a System

For a system with $h(t)$ as the impulse response, and $x(t)$ as the input signal, the output $y(t)$ is related to the input as the convolution,

$$y(t) = x(t) * h(t) = \int_{-\infty}^{\infty} x(u)h(t-u)du$$

According to the convolution property of fourier transforms, the fourier transforms of the signals are related as,

$$Y(j\omega) = H(j\omega)X(j\omega)$$

where $H(j\omega)$, the fourier transform of the impulse response of the system is the frequency response of the system. Likewise, for discrete time input signal $x[n]$ to a system with the impulse response $h[n]$, the output in frequency domain is mathematically given as,

$$Y(e^{j\omega}) = H(e^{j\omega})X(e^{j\omega})$$

where $H(e^{j\omega})$, the fourier transform of the impulse response of the system is the frequency response of the system. During the lab experiment, we plotted the frequency response given as,

$$H(z) = \frac{0.008 - 0.033z + 0.05z^2 - 0.033z^3 + 0.008z^4}{1 + 2.37z + 2.7z^2 + 1.6z^3 + 0.5z^4}$$

The freqz function returns the frequency response of a system whose amplitude and phase were plotted separately in MATLAB.

```
num=[0.008,-0.033,0.05,-0.033,0.008];
den=[1,2.37,2.7,1.6,0.4];
w=200;
x=freqz(num,den,w);
subplot(2,1,1);
plot(abs(x));
xlabel('\omega');
ylabel('|H(\omega)|');
title('Magnitude Plot of Frequency Response'
    );
subplot(2,1,2);
plot(angle(x));
xlabel('\omega');
ylabel('\angle H(\omega)');
title('Phase Plot of Frequency Response');
```

Listing 11: Matlab script for plotting frequency response of$H(z)$

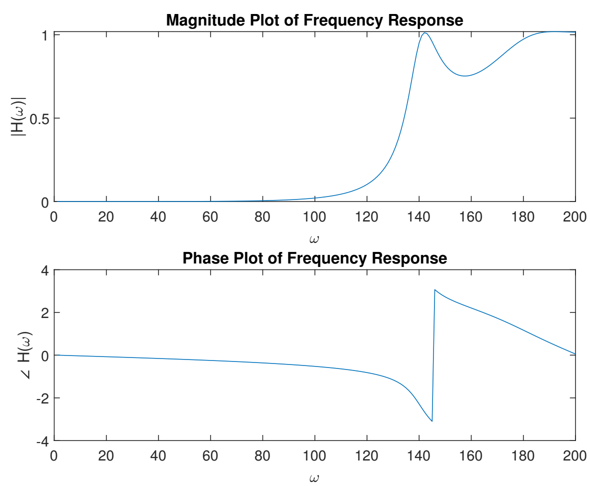

Figure 17: Obtained plot for frequency response of $H(z)$

7 Conclusion

In this lab experiment, we dealt with the basics of signal analysis using MATLAB. The experiments included visualization of basic signals such as sinosoidal, ramp, exponential, unit step and unit impulse. The fourier series approximation of a square wave was also visualized using sinosoidal signal's odd harmonics. Convolutions of discrete signals were also plotted in MATLAB. Likewise, the fourier transform and frequency response of given signals were plotted using the respective functions. Hence, the overall objectives of the lab experiment were fulfilled.