Signals

February 6, 2022

0.1 Signals and basic operations

Plot the Following signals from n = −20 to n = 20. Use the command stem to plot the signals. Clearly mark the time origin in each case.

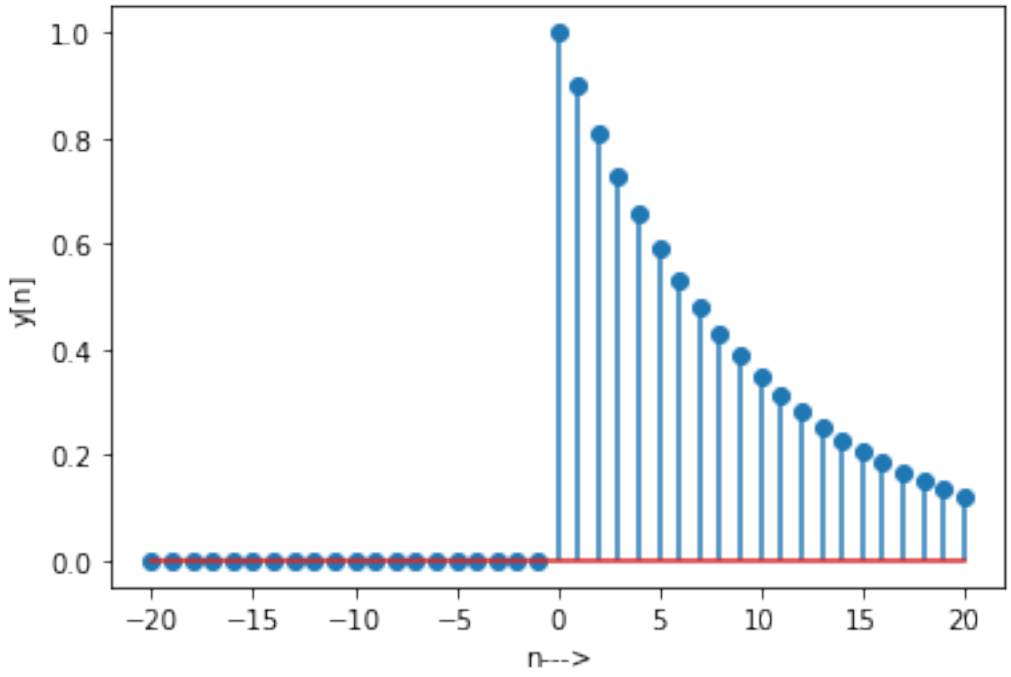
Submitted by:

Utkarsh Mahajan 201EC164

Arnav Raj 201EC109

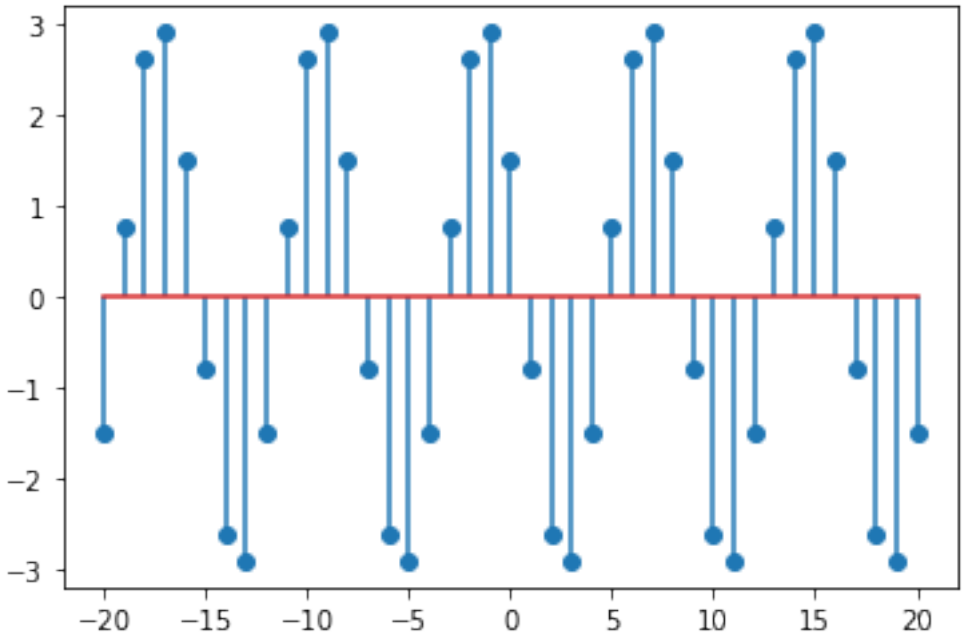
0.1.1 $\left(\frac{9}{10}\right)^n u[n]$

[118]:
```
import numpy as np
import matplotlib.pyplot as plot

n = range(-20,21);  # Range of n
x_n = np.power((9/10), n) * np.heaviside(n,1)
plot.xlabel('n--->')
plot.ylabel('y[n]')
plot.stem(n,x_n)
plot.show()
```



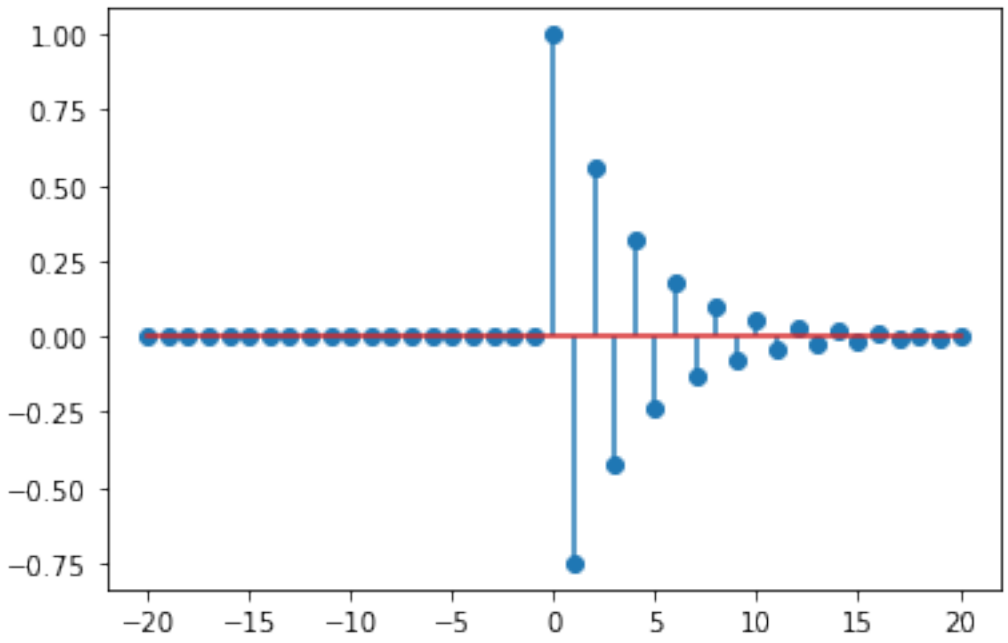
0.1.2 $3\cos\left[\frac{\pi n}{4}+\frac{\pi}{3}\right]$

[14]:
```
#write code here
import numpy as np
import matplotlib.pyplot as plot

x_c= [3*np.cos(np.pi*n/4 + np.pi/3) for n in range(-20,21)]
plot.xlabel('n--->')
plot.ylabel('y[n]')
plot.stem(n, x_c)
plot.show()
```



0.1.3 $(-0.75)^n u[n]$

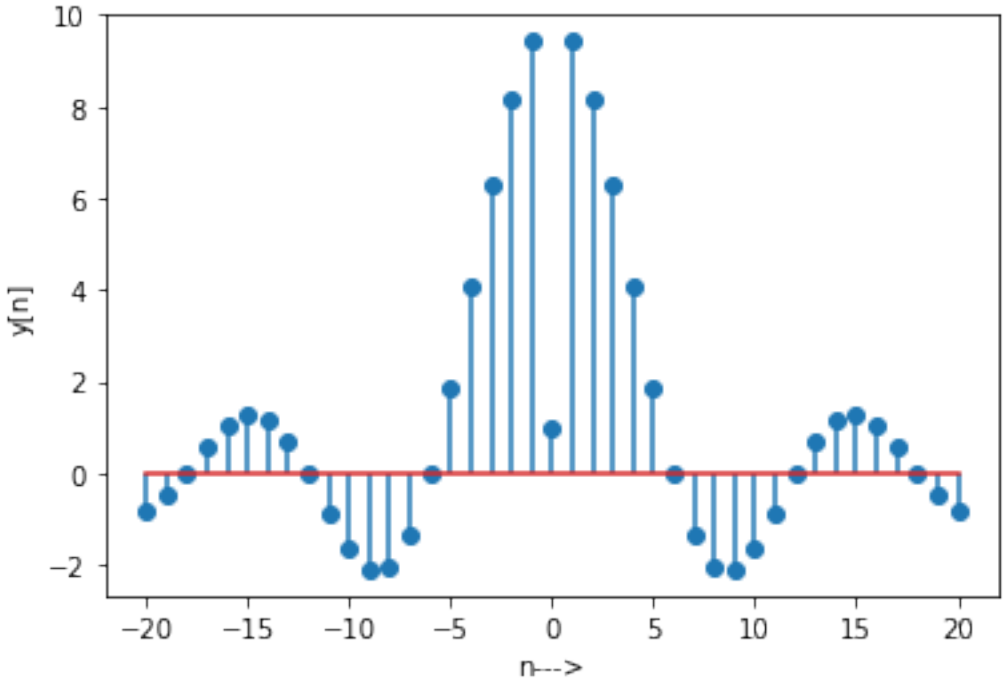
[15]:
```
#write code here
import numpy as np
import matplotlib.pyplot as plot
n = range(-20,21);  # Range of n
f_x = np.power(-0.75, n)* np.heaviside(n,1)
plot.xlabel('n--->')
plot.ylabel('y[n]')
plot.stem(n, f_x)
plot.show()
```



0.1.4 $\frac{\sin(\pi n/6)}{\pi n/6}$

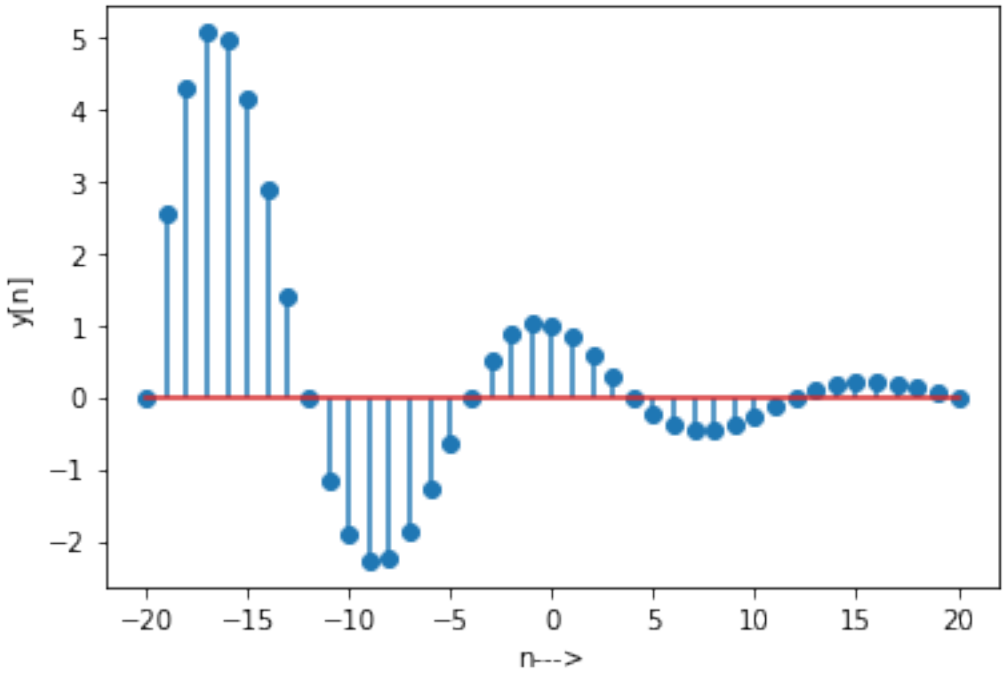
Please make sure that the value at n=0 is not undefined. Consider the limit value

[128]:
```
import numpy as np
import matplotlib.pyplot as plot
n=np.arange(-20,21)
np.seterr(divide='ignore', invalid='ignore')
f_sx = (np.sin(np.pi*n/6)*6)/n*np.pi
f_sx[20] =1
plot.xlabel('n--->')
plot.ylabel('y[n]')
plot.stem(n, f_sx)
plot.show()
```



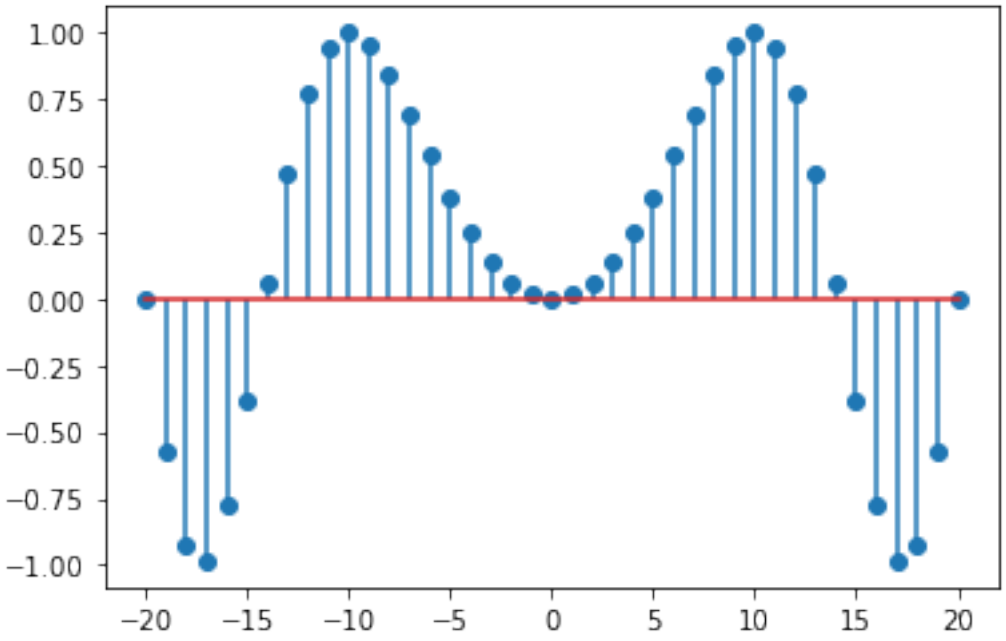
0.1.5 $e^{-0.1n}\cos\left(\frac{\pi n}{8}\right)$

[126]:
```
#write code here
import numpy as np
import matplotlib.pyplot as plot
n = np.arange(-20,21);
f_ecx = np.exp(-0.1*n)*np.cos(np.pi*n/8)
plot.xlabel('n--->')
plot.ylabel('y[n]')
plot.stem(n,f_ecx)
plot.show()
```



0.1.6 $\sin(\pi n^2/200)$

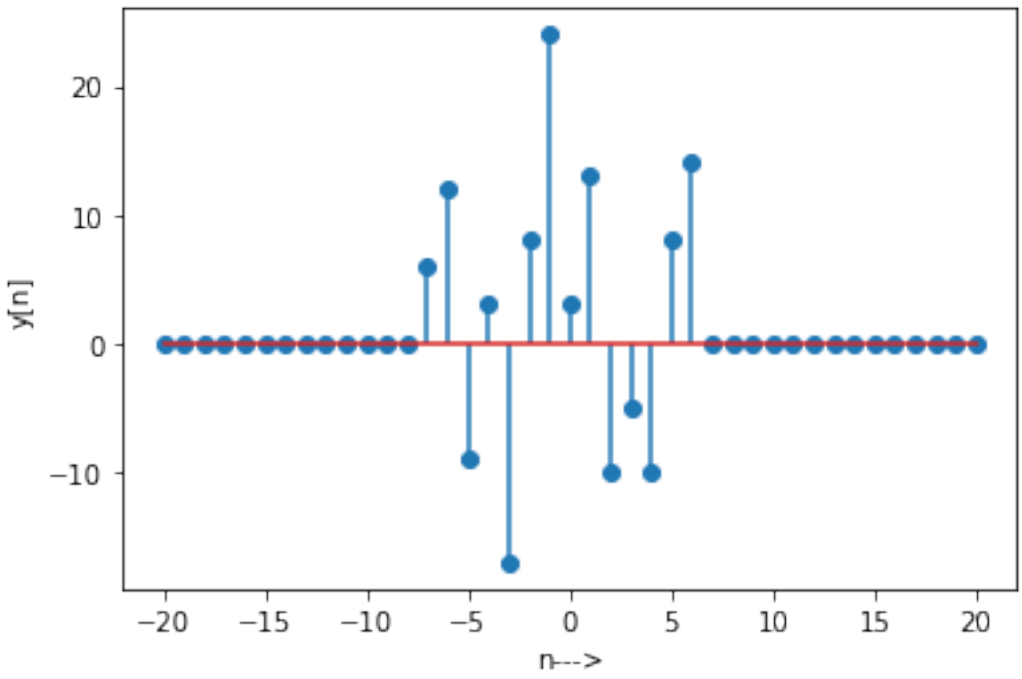
[21]:
```
#write code here
n = range(-20,21);  # Range of n
n=np.array(n)
p=np.power(n,2)
f_nnx=np.sin(np.pi*p/200)
plot.xlabel('n--->')
plot.ylabel('y[n]')
plot.stem(n,f_nnx)
plot.show()
```



Let $x[n] = [2, 4, -3, 1, -5, 4, 7]; -3 \leq n \leq 3$. Generate and plot the samples of the following sequences using the stem function. (You might find fliplr function from numpy useful for computing time reversal)

$x_1[n] = 2x[n-3] + 3x[n+4] - x[n]$ $x_2[n] = x[-n-4]$ $

[2]:
```
import numpy as np
import matplotlib.pyplot as plot
x=[2,4,-3,1,-5,4,7]
zeroes = [0]*100
x = [*x[3:7],*zeroes, *x[0:3]] # to get proper x[n] values from range [-3:3]
#for x1
x1 = [2*x[n-3] + 3*x[n+4]-x[n] for n in range(-20,21)]
plot.xlabel('n--->')
plot.ylabel('y[n]')
plot.stem(range (-20,21),x1)
plot.show()
print(x1)
#for x2
x2 = [x[-n-4] for n in range(-20,21)]
plot.xlabel('n--->')
plot.ylabel('y[n]')
plot.stem(range (-20,21),x2)
plot.show()
print(x2)
```


20
10
0
−10
y[n]
−20
−15
−10
−5
0
5
10
15
20
n--->
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 6, 12, -9, 3, -17, 8, 24, 3, 13, -10, -5, -10, 8, 14, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
6
4
2
0
−2
−4
y[n]
−20
−15
−10
−5
0
5
10
15
20
n--->

```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 7, 4, -5, 1, -3, 4, 2, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```