December 2010
7 Tuesday

डिसेंबर २०१०
मंगळवार ७
मार्गशीर्ष शु. २, मूळ

(1) $$y[n] = \frac{1}{n+1}\sum_{k=0}^{n} x[k]$$

& $$\Rightarrow y[n-1] = \frac{1}{n}\sum_{k=0}^{n-1} x[k]$$

$$\Rightarrow y[n] = \frac{1}{n+1}\left(\left(\sum_{k=0}^{n-1} x[k]\right) + x[n]\right)$$

$$= \frac{1}{n+1}\left(n\, y[n-1] + x[n]\right)$$

$\therefore$ In recursive form,

$$\boxed{y[n] = \frac{n}{n+1} y[n-1] + \frac{1}{n+1} x[n]}$$

Assuming it's causal LTI, imposing initial rest, $y[n] = 0 \;\forall n < 0$

I take $x[n]$ as follows :-

$$x[n] = \begin{cases} 1 \;; 0 \le n \le 7 \\ 0 \;; \text{otherwise} \end{cases}$$

$$\Rightarrow \boxed{y[n] = \left(\frac{n}{n+1}\right) y[n-1] + \left(\frac{1}{n+1}\right) x[n]}$$

190020018_SSProgramming_A3

November 20, 2020

[40]:
```
import numpy as np
import matplotlib.pyplot as plt
import math
```

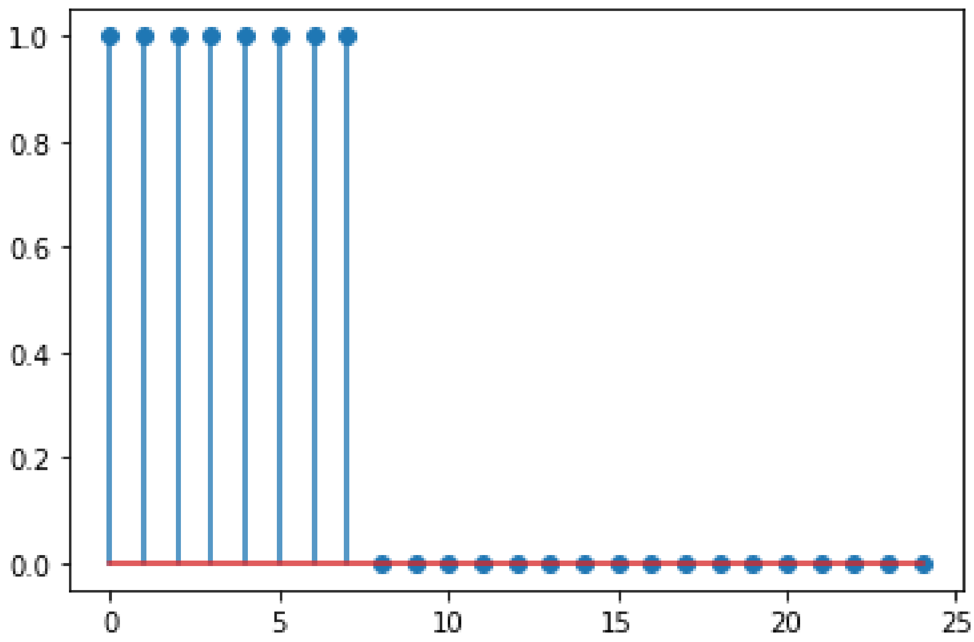
0.1 Question 1 part a:

0.2 I took x[n] = 1; if 0<=n<=7 and 0 otherwise...

[41]:
```
n=list(range(25))
x=[]
for i in range(8):
    x.append(1)
for i in range(8,25):
    x.append(0)

plt.stem(n,x)
plt.show()
```

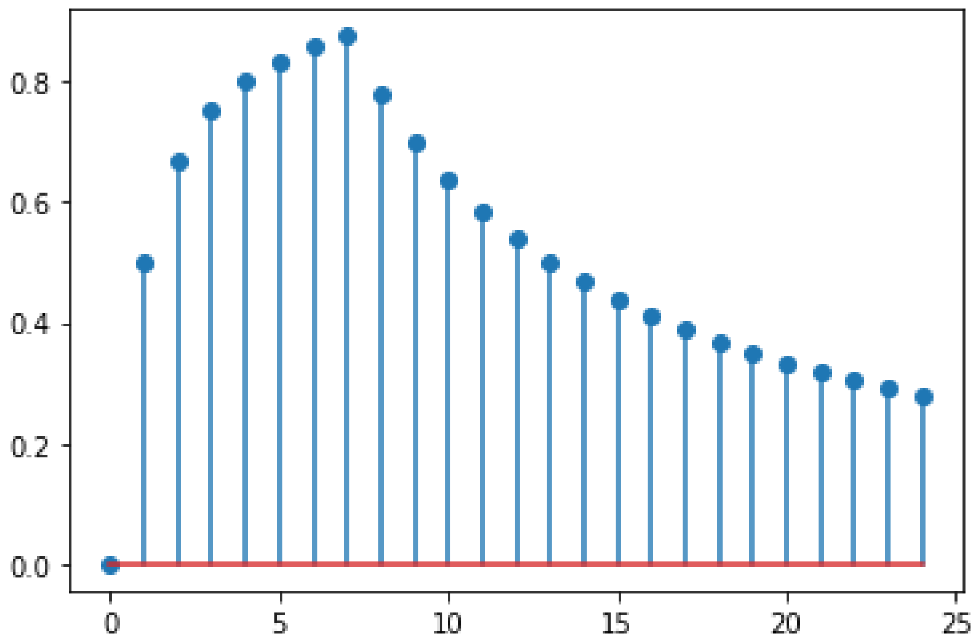
```
<ipython-input-41-d33823bc8cbb>:8: UserWarning: In Matplotlib 3.3 individual
lines on a stem plot will be added as a LineCollection instead of individual
lines. This significantly improves the performance of a stem plot. To remove
this warning and switch to the new behaviour, set the "use_line_collection"
keyword argument to True.
  plt.stem(n,x)
```



[42]:
```
y=[0]
for i in range(1,25):
    y_current = (i*y[i-1]/(i+1))+(x[i]/(i+1))
    y.append(y_current)

plt.stem(n,y)
plt.show()
```

```
<ipython-input-42-45ce8751db60>:6: UserWarning: In Matplotlib 3.3 individual
lines on a stem plot will be added as a LineCollection instead of individual
lines. This significantly improves the performance of a stem plot. To remove
this warning and switch to the new behaviour, set the "use_line_collection"
keyword argument to True.
  plt.stem(n,y)
```



0.3 Question 1 part b:

0.4 x[n]=delta[n-1]

[43]:

```
n=list(range(25))
x=[0,1]
for i in range(2,25):
    x.append(0)

plt.stem(n,x)
plt.show()
```

```
<ipython-input-43-a17c5142a13a>:6: UserWarning: In Matplotlib 3.3 individual
lines on a stem plot will be added as a LineCollection instead of individual
lines. This significantly improves the performance of a stem plot. To remove
this warning and switch to the new behaviour, set the "use_line_collection"
keyword argument to True.
  plt.stem(n,x)
```

Given, a causal LTI system is described by

$$\boxed{y[n] = \frac{1}{4} y[n-1] + x[n]}$$

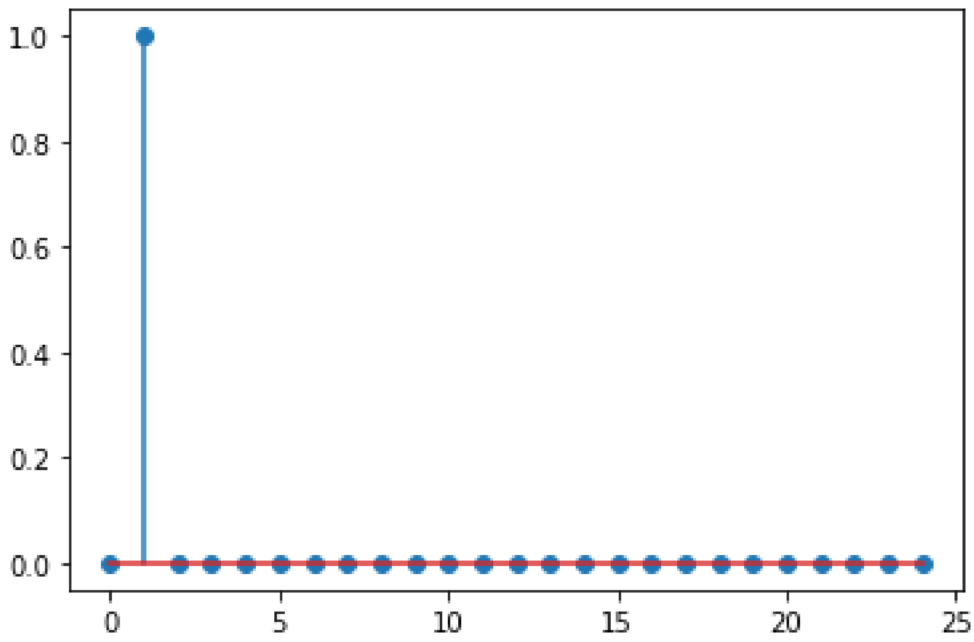
Given,

$$x[n] = \delta[n-1]$$

And as its causal, imposing initial rest,

$$y[n] = 0 \quad \forall n \leq 0$$

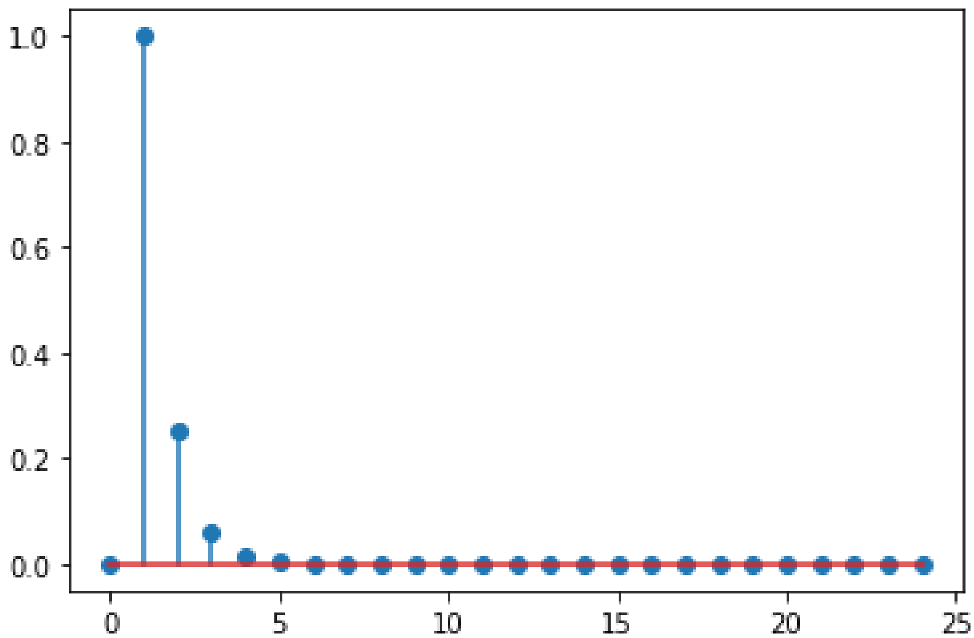
$$\because \; x[n] = 0 \quad \forall n \leq 0.$$



[44]:
```
y=[0]
for i in range(1,25):
    y_current = x[i]+(1/4)*y[i-1]
    y.append(y_current)

plt.stem(n,y)
plt.show()
```

```
<ipython-input-44-43185da32a50>:6: UserWarning: In Matplotlib 3.3 individual
lines on a stem plot will be added as a LineCollection instead of individual
lines. This significantly improves the performance of a stem plot. To remove
this warning and switch to the new behaviour, set the "use_line_collection"
keyword argument to True.
  plt.stem(n,y)
```



0.5 Question 2

0.6 Plot of x[n]

[45]:
```
n=list(range(-12,13))
x=[]
for i in range(25):
    if i%4 == 0:
        x.append(4)
    elif i%4 == 1:
        x.append(8)
    else:
        x.append(0)
plt.stem(n,x)
plt.show()
```

```
<ipython-input-45-ffcbc9dfbdb4>:10: UserWarning: In Matplotlib 3.3 individual
lines on a stem plot will be added as a LineCollection instead of individual
lines. This significantly improves the performance of a stem plot. To remove
this warning and switch to the new behaviour, set the "use_line_collection"
keyword argument to True.
  plt.stem(n,x)
```

2

$$x[n] = \sum_{m=-\infty}^{+\infty} \left(4\delta[n-4m] + 8\delta[n-1-4m] \right)$$

We see that, (from def^n of Impulse)

$$x[0] = \sum_{m=-\infty}^{+\infty} \left(4\delta[-4m] + 8\delta[-1-4m] \right)$$

$$= 4\delta[0] = 4.$$

$$x[1] = 8\delta[0] = 8$$

$$x[2] = 0 \; ; \; x[3] = 0$$

$$\& \; x[4] = x[0] = 4\delta[0] = 4$$

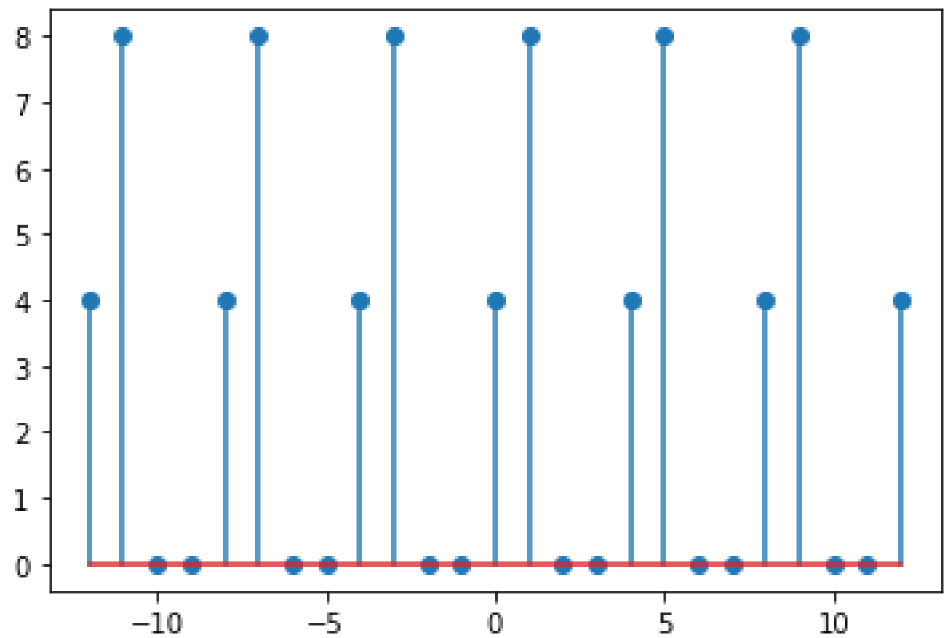
$$\& \; x[5] = x[1] = 8\delta[0] = 8$$

⋮ and so on.

So, we see that

$$x[n] = x[n+4]$$

$\Rightarrow$ $x[n]$ is Periodic with Fundamental Period $N = 4$.



0.7 Clearly, the above signal is periodic with N=4

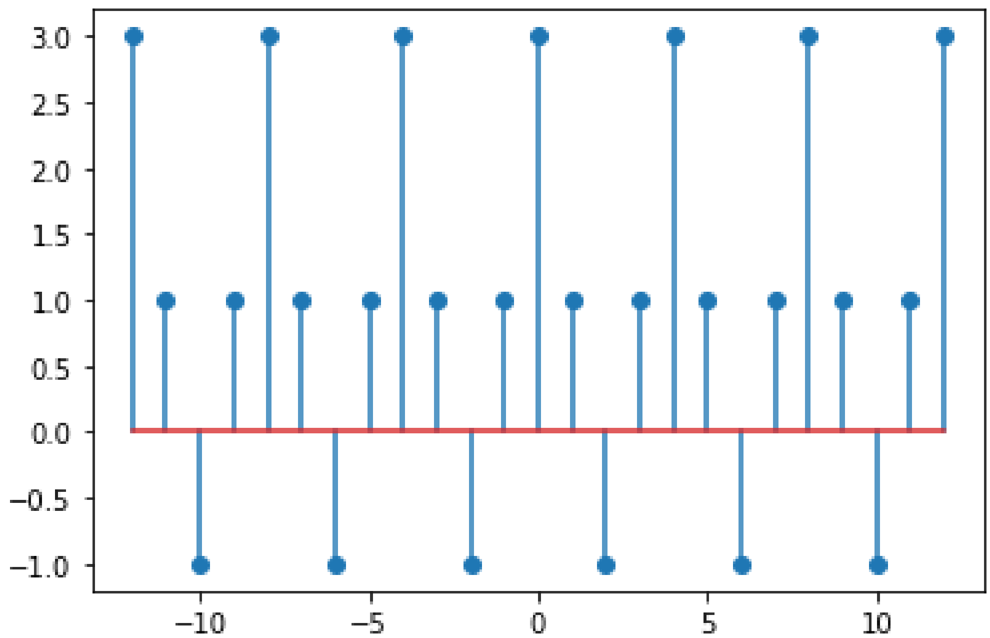
0.8 Plot of DTFS coefficients of x[n]:

0.9 Plot of the real part:

[46]:
```
ak_real=[]
for k in range(4):
    ak_real_current=0
    for i in range(4):
        ak_real_current+=(x[i]*math.cos(k*i*math.pi/2)/4)
    ak_real.append(ak_real_current)
for k in range(4,25):
    ak_real_current=ak_real[k%4]
    ak_real.append(ak_real_current)

plt.stem(n,ak_real)
plt.show()
```

```
<ipython-input-46-dd3d4eaab61c>:11: UserWarning: In Matplotlib 3.3 individual
lines on a stem plot will be added as a LineCollection instead of individual
lines. This significantly improves the performance of a stem plot. To remove
this warning and switch to the new behaviour, set the "use_line_collection"
keyword argument to True.
  plt.stem(n,ak_real)
```



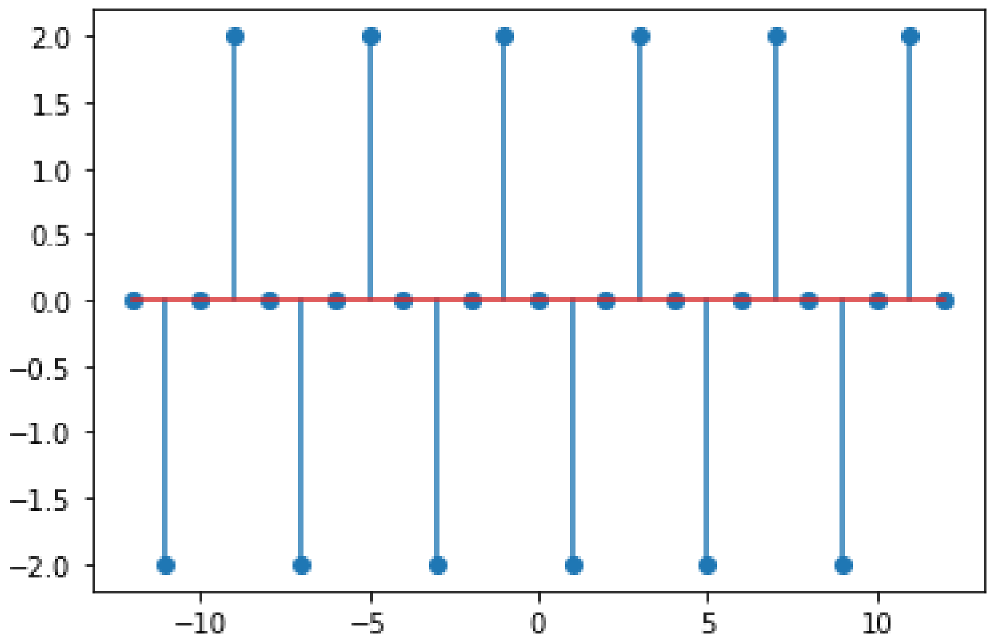
0.10 As it's a real-valued signal, Real part of DTFS coefficients must be an even function and periodic with period 4 which is evident from the above plot...

0.11 Plot of the imaginary part:

[47]:
```
ak_img=[]
for k in range(4):
    ak_img_current=0
    for i in range(4):
        ak_img_current+=(x[i]*(-1)*math.sin(k*i*math.pi/2)/4)
    ak_img.append(ak_img_current)
for k in range(4,25):
    ak_img_current=ak_img[k%4]
    ak_img.append(ak_img_current)

plt.stem(n,ak_img)
plt.show()
```

```
<ipython-input-47-7e8cebac67ff>:11: UserWarning: In Matplotlib 3.3 individual
lines on a stem plot will be added as a LineCollection instead of individual
lines. This significantly improves the performance of a stem plot. To remove
this warning and switch to the new behaviour, set the "use_line_collection"
keyword argument to True.
  plt.stem(n,ak_img)
```



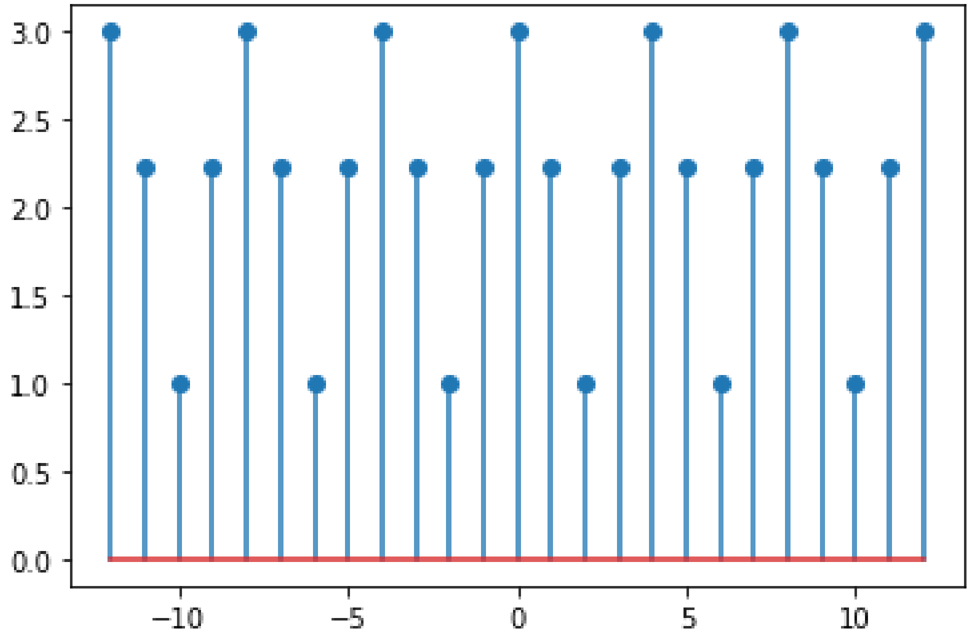
0.12 As it's a real-valued signal, Imaginary part of DTFS coefficients must be an odd function and periodic with period 4 which is evident from the above plot…¶

0.13 Plot of |ak|: (This must be even function and periodic with period N=4 which is evident from the below plot)

[48]:
```
ak_mag=[]
for i in range(25):
    ak_mag_current=math.sqrt((ak_real[i]**2)+(ak_img[i]**2))
    ak_mag.append(ak_mag_current)

plt.stem(n,ak_mag)
plt.show()
```

```
<ipython-input-48-4e2b6060d5d0>:6: UserWarning: In Matplotlib 3.3 individual
lines on a stem plot will be added as a LineCollection instead of individual
lines. This significantly improves the performance of a stem plot. To remove
this warning and switch to the new behaviour, set the "use_line_collection"
keyword argument to True.
  plt.stem(n,ak_mag)
```

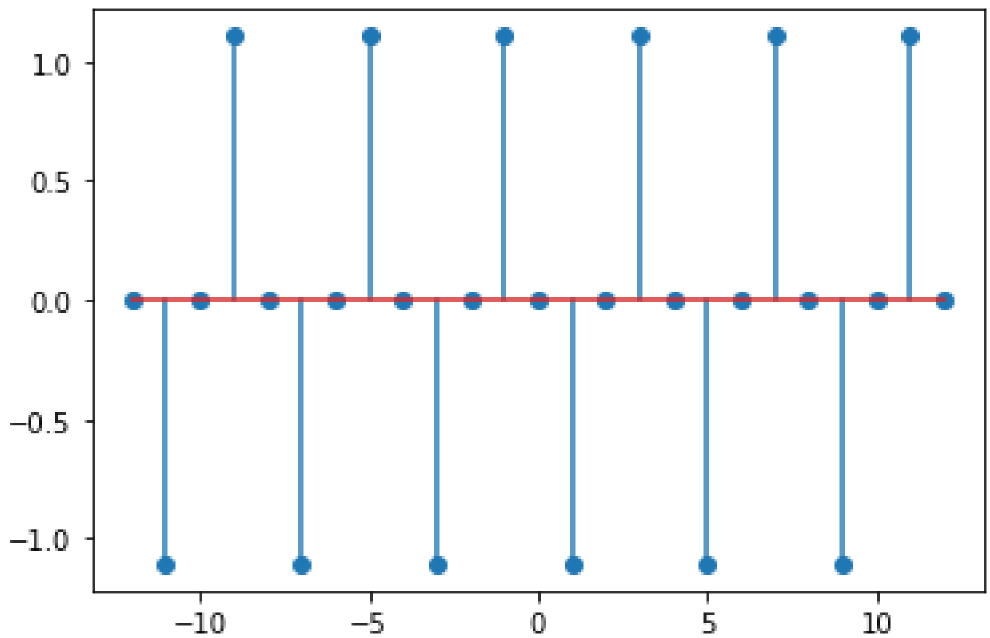


0.14 Phase Plot of ak: (This must be odd function and periodic with period N=4 which is evident from the below plot)

[49]:
```
ak_phase=[]
for i in range(25):
    ak_phase_current=math.atan(ak_img[i]/ak_real[i])
    ak_phase.append(ak_phase_current)

plt.stem(n,ak_phase)
plt.show()
```

```
<ipython-input-49-32a5a1070e27>:6: UserWarning: In Matplotlib 3.3 individual
lines on a stem plot will be added as a LineCollection instead of individual
lines. This significantly improves the performance of a stem plot. To remove
this warning and switch to the new behaviour, set the "use_line_collection"
keyword argument to True.
  plt.stem(n,ak_phase)
```



0.15 QUESTION 3:

0.16 Plot of one period of x[n]:

0.17 Here, I am considering n=0,1,2,3 as one period...

[50]:
```python
x=[]
for i in range(25):
    x.append(0)
x[12]=4
x[13]=8

plt.stem(n,x)
plt.show()
```

```
<ipython-input-50-a49d1929f2fb>:7: UserWarning: In Matplotlib 3.3 individual
lines on a stem plot will be added as a LineCollection instead of individual
lines. This significantly improves the performance of a stem plot. To remove
this warning and switch to the new behaviour, set the "use_line_collection"
keyword argument to True.
  plt.stem(n,x)
```

Q3) I considered

$n = 0, 1, 2, 3$ values

as a period & thus

$x[0] = 4$

$x[1] = 8$ & $x[n] = 0$ $\forall$ remaining 'n'.

$\therefore$ Its DTFT is given by

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

$$= x[0] e^{-j\omega(0)} + x[1] e^{-j\omega(1)}.$$

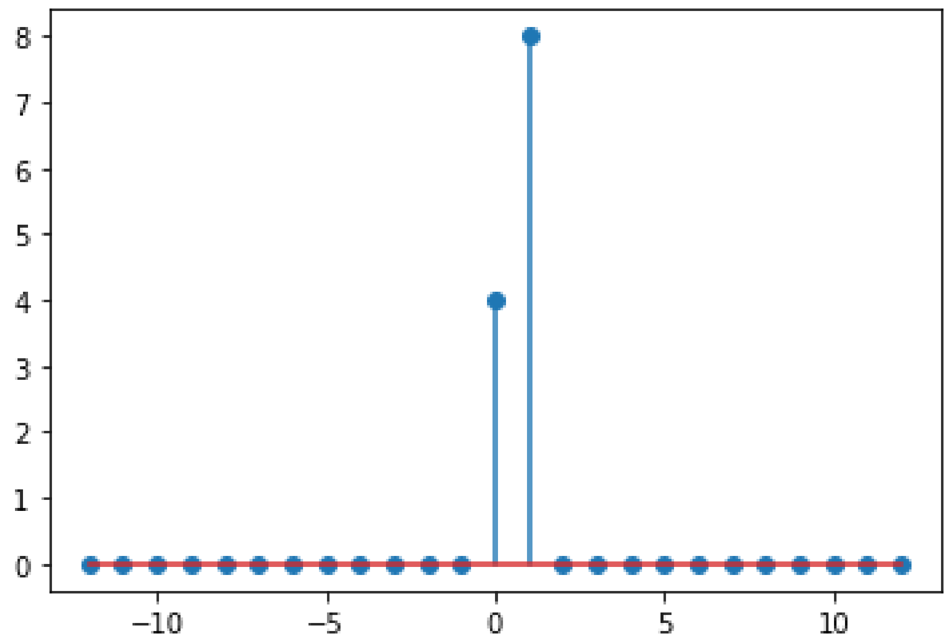
$\because x[n] = 0$ ($\forall$ remaining 'n')

$$X(e^{j\omega}) = x[0] + x[1] \, e^{-j(\omega)}$$

$$\Rightarrow X(e^{j\omega}) = 4 + 8e^{-j\omega}$$

$$= 4 + 8(\cos\omega - j\sin\omega)$$

$$= \underbrace{4 + 8\cos\omega}_{Re\{X(e^{j\omega})\}} + j\underbrace{(-8\sin\omega)}_{Im\{X(e^{j\omega})\}}$$

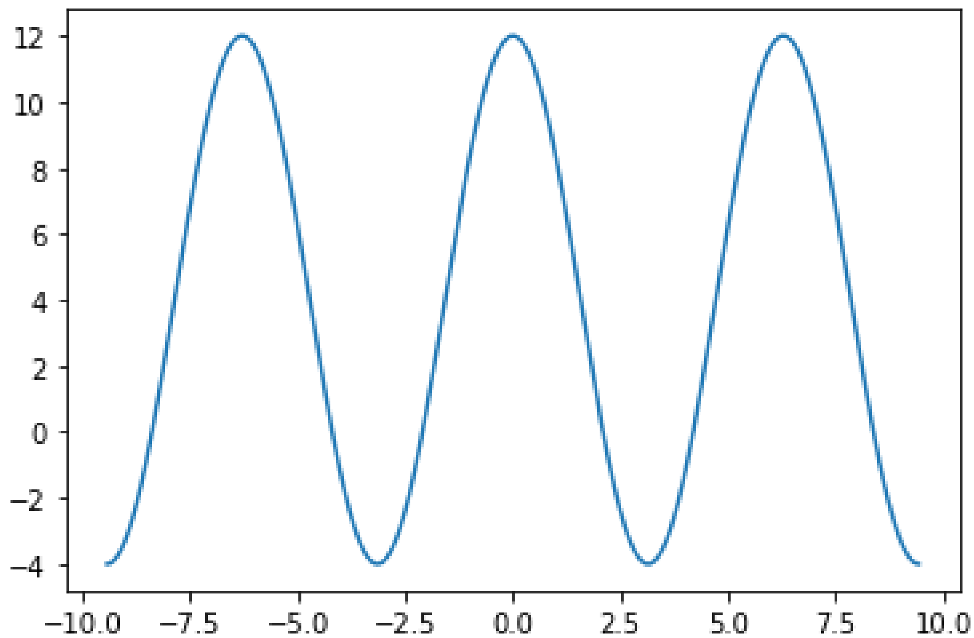


[51]:
```
omega=np.linspace(-3*np.pi,3*np.pi,500)
real_X= 4+8*np.cos(omega)
img_X= -8*np.sin(omega)
mag_X= np.sqrt(real_X**2+img_X**2)
phase_X=np.arctan(img_X/real_X)
```

0.18 PLOT of REAL PART of DTFT of one period of x[n]:

0.19 It must be even function and periodic which is evident from below plot:

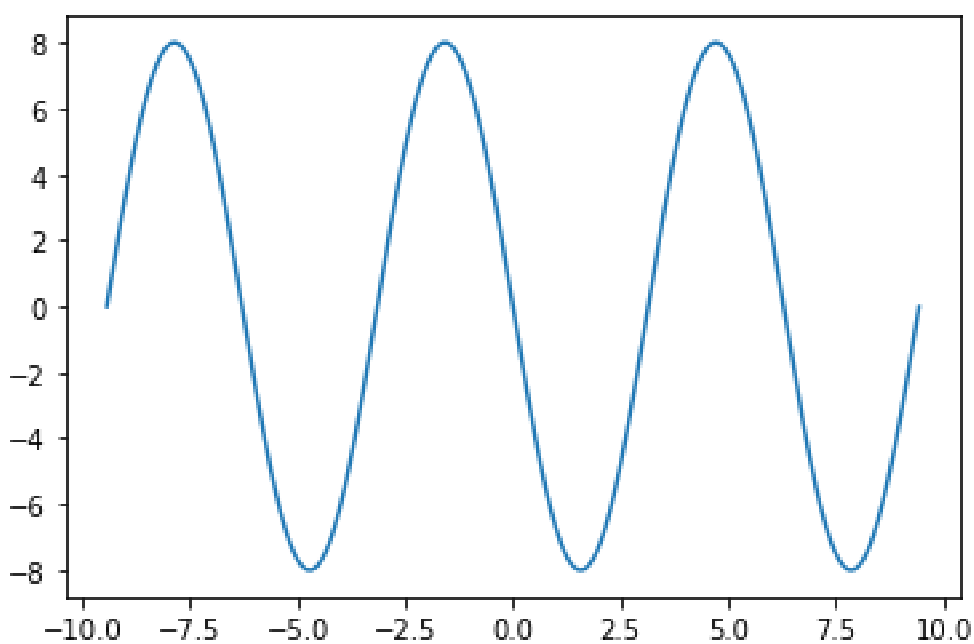
[52]:
```
plt.plot(omega,real_X)
plt.show()
```



0.20 PLOT of IMAGINARY PART of DTFT of one period of x[n]:

0.21 It must be odd function and periodic which is evident from below plot:

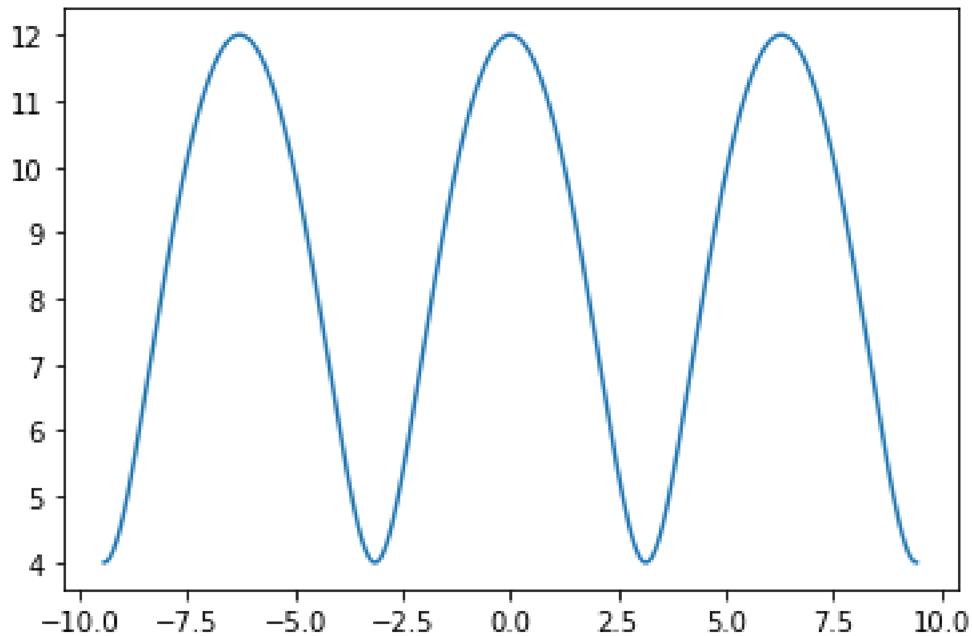
[53]:
```
plt.plot(omega,img_X)
plt.show()
```



0.22 PLOT of MAGNITUDE of DTFT of one period of x[n]:

0.23 It must be even function and periodic which is evident from below plot:

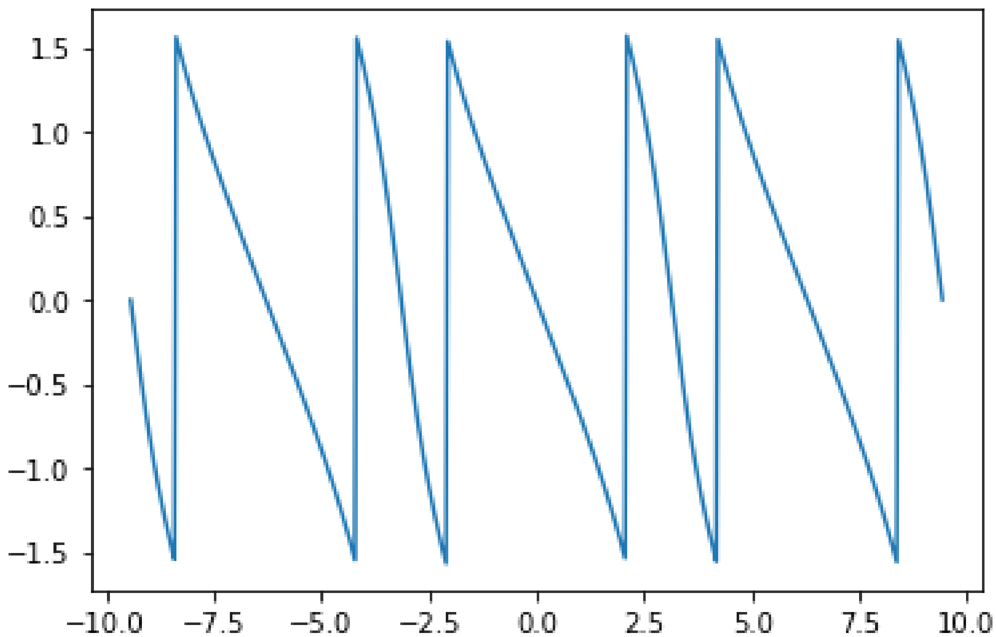
[54]:
```
plt.plot(omega,mag_X)
plt.show()
```



0.24 PHASE PLOT of DTFT of one period of x[n]:

0.25 It must be odd function and periodic which is evident from below plot:

[55]:
```
plt.plot(omega,phase_X)
plt.show()
```



1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0


```
[ ]:
```

```
[ ]:
```