200014B Group A-1

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.fftpack import fft,fftshift,ifft
from scipy import signal

def square(t):
  if t % 1 < 0.25 or t % 1 > 0.75:
    s = 1
  elif t % 1 == 0.25 or t % 1 == 0.75:
    s = 0.5
  else:
    s = 0
  return s

# Fourier series coefficients
def a(k):
  if k == 0:
    a_k = 1/2
  else:
    a_k = np.sin(k*np.pi*t)/k*np.pi
  return a_k
```

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.fftpack import fft,fftshift,ifft
from scipy import integrate
from scipy import signal

def square(t):
    if t % 1 < 0.25 or t % 1 > 0.75:
```



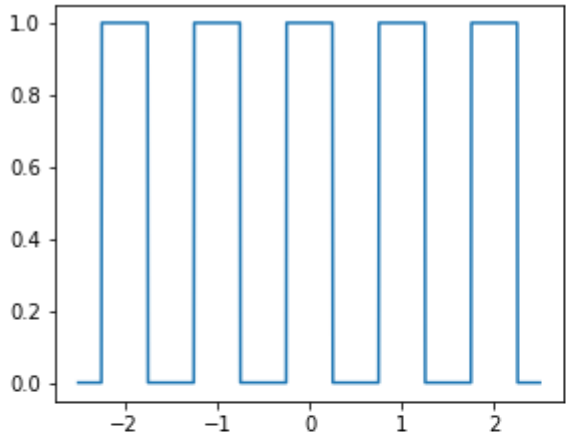
```
    else:
        s = 0
    return s


def a(k):

    if k==0:
        return 1/2
    else :
        a_k = np.sin(k*np.pi/2)/(k*np.pi)

    return a_k
```

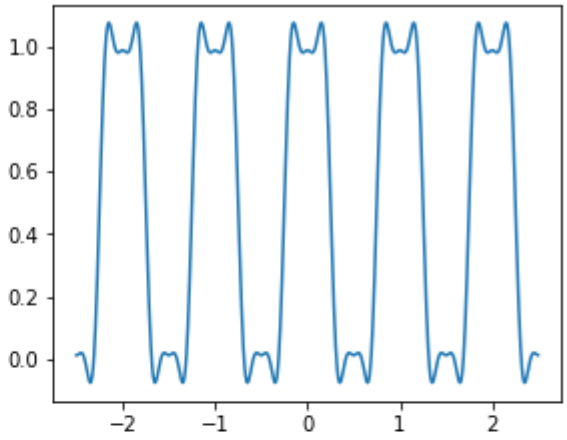
```
def fs_approx(t, N):
    global w
    w=2*np.pi
    x_t=0

    for n in range(-N,N):
        x_t+=a(n)*np.exp(1j*n*w*t)

    return x_t

time=np.arange(-2.5,2.5,5/1000)
x=[]
y=[]
N=5
for t in time:
    x.append(square(t))
    y.append(fs_approx(t,N))

fig,ax=plt.subplots(2,2,figsize=(10,8))
ax[0,0].plot(time, [square(t_) for t_ in time])
ax[0,1].plot(time, [fs_approx(t_,N) for t_ in time])

time=np.arange(-2.5,2.5,5/1000)
x=[]
y=[]
N=50
for t in time:
    x.append(square(t))
    y.append(fs_approx(t,N))


ax[1,0].plot(time, [square(t_) for t_ in time])
ax[1,1].plot(time, [fs_approx(t_,N) for t_ in time])
```

Automatic saving failed. This file was updated remotely or in another tab. Show diff

```
plt.show()
```

```
/usr/local/lib/python3.7/dist-packages/matplotlib/cbook/__init__.py:1317: ComplexWarning
  return np.asarray(x, float)
/usr/local/lib/python3.7/dist-packages/matplotlib/cbook/__init__.py:1317: ComplexWarning
  return np.asarray(x, float)
```

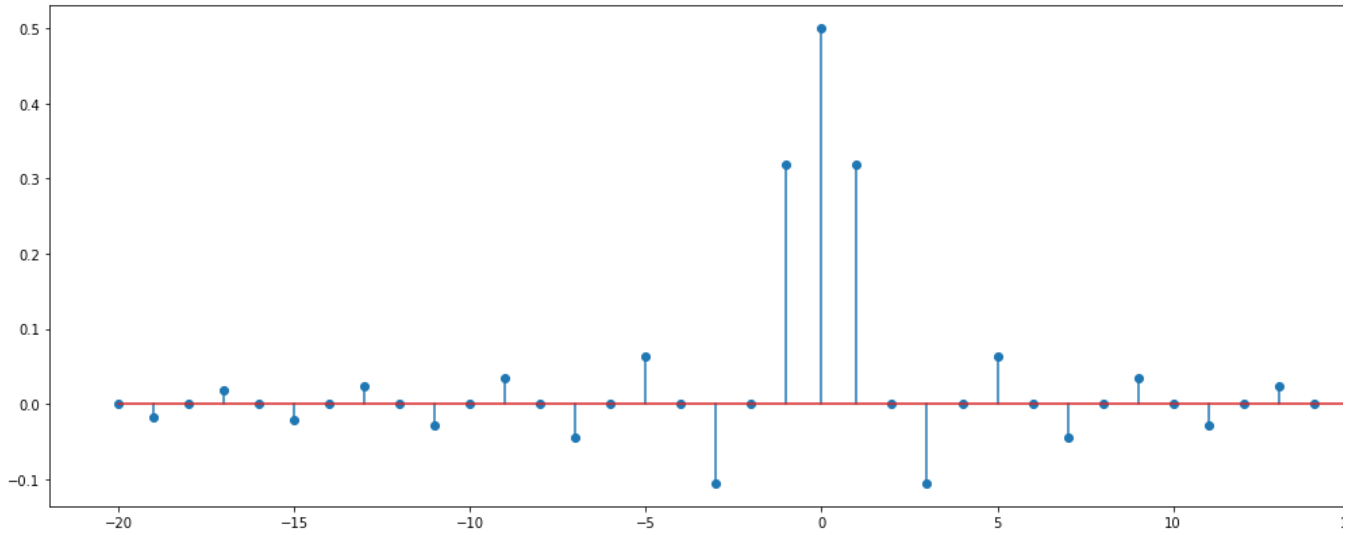


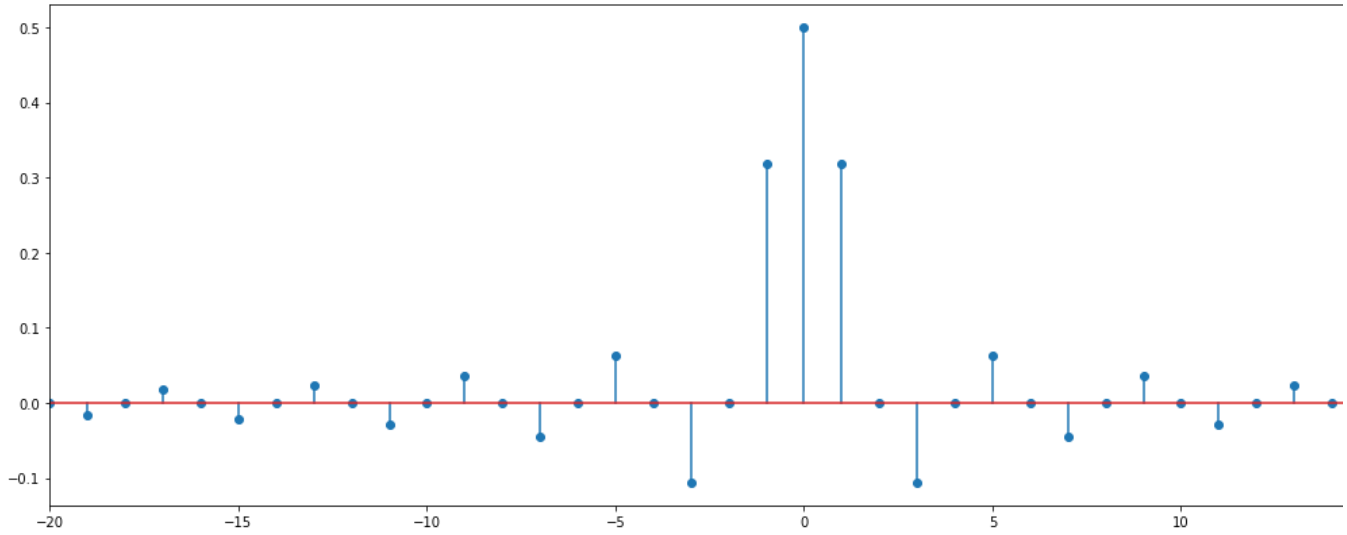



```
N = 200
t = np.linspace(0, 1 -1/N, N)
x = []
for i in t:
    x.append(square(i))

X = fftshift(fft(x))
X_norm = X.real/N

K=np.arange(-20,20)
a_k=[]
for i in K:
    a_k.append(a(i))

fig,ax=plt.subplots(2,1,figsize=(20,15))
ax[0].stem(K,[a_k[c] for c in range(len(K))])

ax[1].stem(k,X_norm)
```

Automatic saving failed. This file was updated remotely or in another tab. Show diff

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning: In Matplot
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:18: UserWarning: In Matplot
```

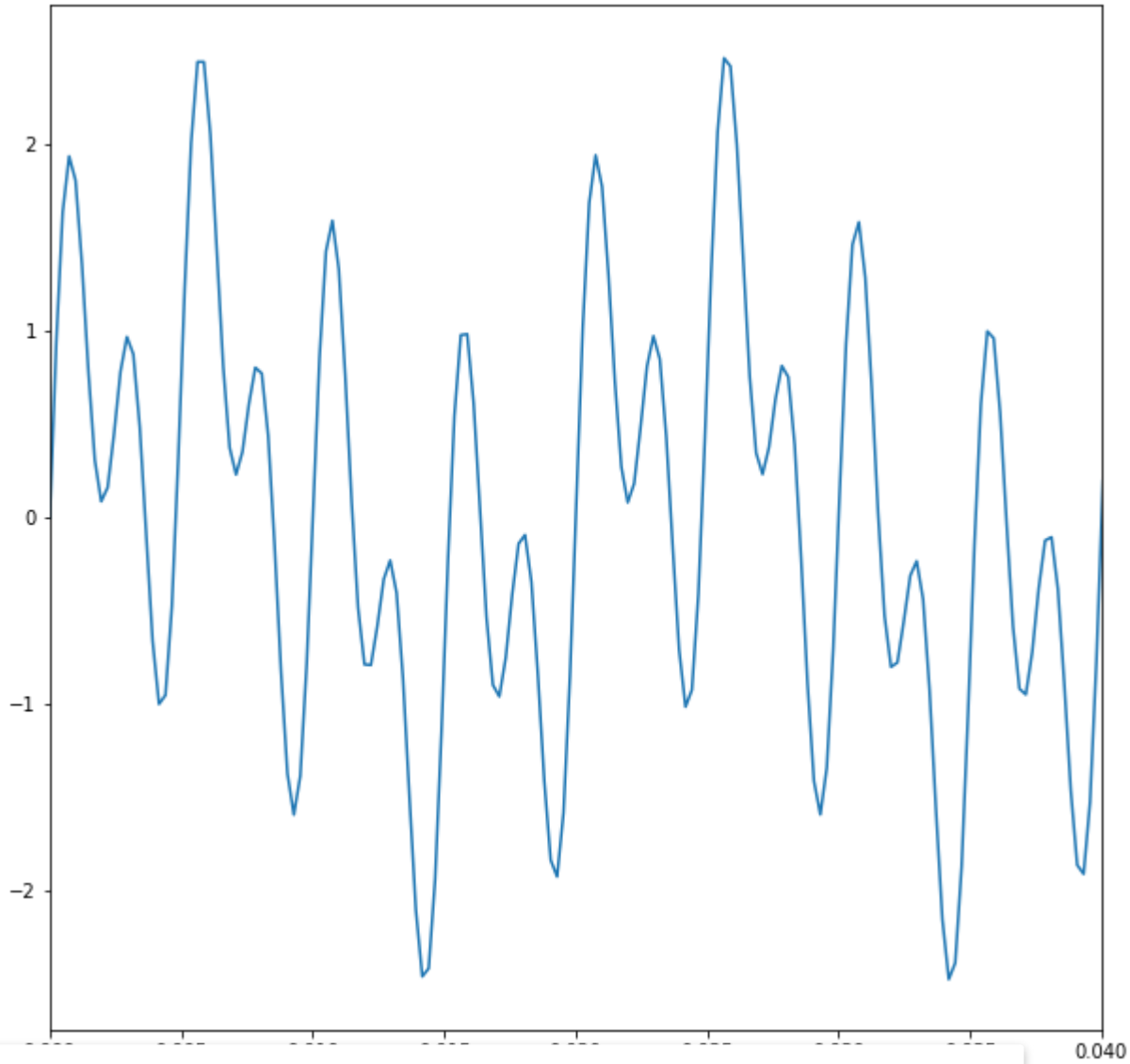




Automatic saving failed. This file was updated remotely or in another tab. Show diff

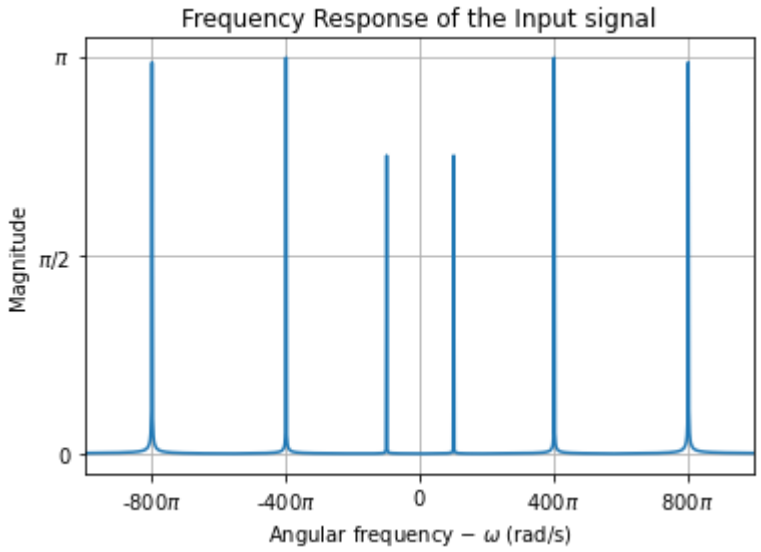
```
w2 = 400*np.pi
w3 = 800*np.pi
a1 = 0.75
a2 = 1
a3 =  0.5
fs = 4095
ws = 2*np.pi*fs
def x(t):
  x_t = a1*np.sin(w1*t) + a2*np.sin(w2*t) + a2*np.sin(w3*t)
  return x_t

time = np.linspace(0,1,fs+1)
xt = [x(t_) for t_ in time]
```

```
fig,ax=plt.subplots(1,1,figsize=(10,10))
ax.plot(time, [xt[t_] for t_ in range(len(time))])
ax.set_xlim(0,0.04)
plt.show()
```




```
Xw = fft(xt, 4096)*2*np.pi/fs
Xw = fftshift(Xw)
k = np.arange(1,4097)
w = k/4096*ws - ws/2

fig, ax = plt.subplots()
Xw_ = np.abs(Xw)
ax.plot(w, Xw_)

ax.set_title("Frequency Response of the Input signal")
ax.set_xlabel("Angular frequency −"+r" $\omega$ (rad/s)")
ax.set_ylabel('Magnitude')
ax.set_xticks(np.arange(-1200*np.pi, 1200*np.pi+1,400*np.pi))
ax.set_xticklabels([str(i)+(r'$\pi$' if i else '') for i in range(-1200,1210,400)])
```

```
ax.set_xlim(-1000*np.pi, 1000*np.pi)
ax.set_yticks([0,np.pi/2,np.pi])
ax.set_yticklabels([0,r'$\pi$/2',r'$\pi$'])
plt.grid()
```




```
wc1 = (w1+w2)/2
wc2 = (w2+w3)/2
def ideal_filter(w):
  if wc1<w<wc2:
    gain = 1
  else:
    gain = 0
  return gain
```

Automatic saving failed. This file was updated remotely or in another tab. Show diff

✓ 0s completed at 9:46 PM

Automatic saving failed. This file was updated remotely or in another tab. Show diff