

In [15]:

```
%reset -f
import matplotlib.pyplot as plt
%matplotlib
import numpy as np
from scipy.fft import fft, fftfreq, fftshift, ifft
from scipy.signal import windows as win
import scipy.signal as sig
```

Using matplotlib backend: Qt5Agg

In [16]:

```
fs = 256 #sample rate
t = np.arange(0,1,1/fs) #time vector
phase_ang = np.pi/4

#time domain signal consisting of two frequency components and phase angle offset
x0 = np.cos(2*np.pi*12*t)
x1 = np.cos(2*np.pi*24*t + phase_ang)

x = x0 + x1

plt.figure(1)
plt.plot(t,x)

out = fft(x)
fv = fftfreq(len(t),1/fs)

fig, axs = plt.subplots(2)
axs[0].plot(abs(out))
axs[1].plot(fv)

sp = fftshift(out)*2/len(t)
freq = fftshift(fv)

plt.figure(3)
plt.plot(freq,abs(sp))
```

Out[16]:

[<matplotlib.lines.Line2D at 0x2b879ffdf40>]

In [124]:

```
#time domain signal consisting of two frequency components and phase angle offset
x0 = np.cos(2*np.pi*12.5*t)
x1 = np.cos(2*np.pi*24*t + phase_ang)

x = x0 + x1

plt.close()

plt.figure(1)
plt.plot(t,x)

out = fft(x)
sp1 = fftshift(out)*2/len(t)

plt.figure(3)
plt.plot(freq,abs(sp1),label='no win')
plt.legend()
```

Out[124]:

<matplotlib.legend.Legend at 0x1aa8a923ac0>

In [125]:

```
hw = win.hann(len(t))
plt.figure(1)
plt.plot(t,hw)
plt.show()

plt.plot(t,x*hw,'r')
plt.show()

outh = fft(x*hw)
sph = fftshift(outh)*2/len(t)

plt.figure(3)
plt.plot(freq,abs(sph))

plt.plot(freq,abs(sph),label='win')
plt.legend()
```

Out[125]:

<matplotlib.legend.Legend at 0x1aa8af87be0>

In [126]:

```
Aw = len(hw)/sum(hw)
plt.figure(3)
plt.plot(freq,abs(sph)*Aw,label='win, corr')
plt.legend()
```

Out[126]:

<matplotlib.legend.Legend at 0x1aa88727820>

In [47]:

```

%reset -f
import matplotlib.pyplot as plt
import numpy as np
from scipy.fft import fft, fftfreq, fftshift, ifft
from scipy.signal import windows as win
import scipy.signal as sig

fs = 1024 #sample rate
t = np.arange(0,1,1/fs) #time vector
phase_ang = np.pi/4

#time domain signal consisting of two frequency components and phase angle offset
x0 = np.cos(2*np.pi*12*t)
x1 = np.cos(2*np.pi*250*t + phase_ang)

x = x0 + x1

b,a = sig.butter(8,2*np.pi*18,'low',analog=True)
w, h = sig.freqs(b, a)

plt.figure(4)
plt.semilogx(w, 20 * np.log10(abs(h)),2*np.pi*18,-3,'o')

sos = sig.butter(8,2*np.pi*12,'low',output='sos',fs = fs)
xf = sig.sosfilt(sos,x)

plt.figure(5)
plt.plot(t,x,t,xf)

hw = win.hann(len(t))
Aw = len(hw)/sum(hw)

out = fft(x*hw)
sp = fftshift(out)*2/len(t) * Aw
freq = fftshift(fftfreq(len(t),1/fs))

outf = fft(xf*hw)
spf = fftshift(outf)*2/len(t) * Aw

plt.figure(6)
plt.plot(freq,abs(sp),label='orig')
plt.plot(freq,abs(spf),label='filt')
plt.legend()

```

Out[47]:

&lt;matplotlib.legend.Legend at 0x2b87c6b6220&gt;

In [ ]: