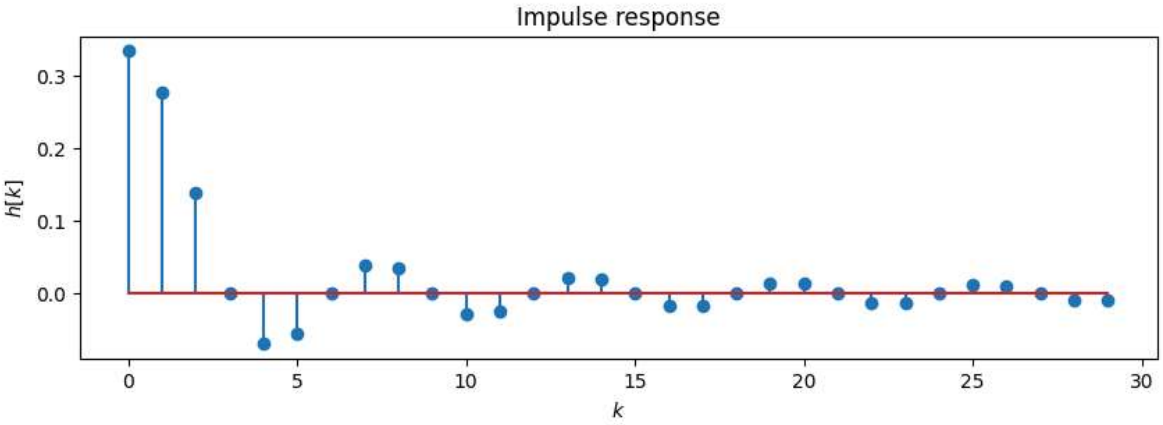
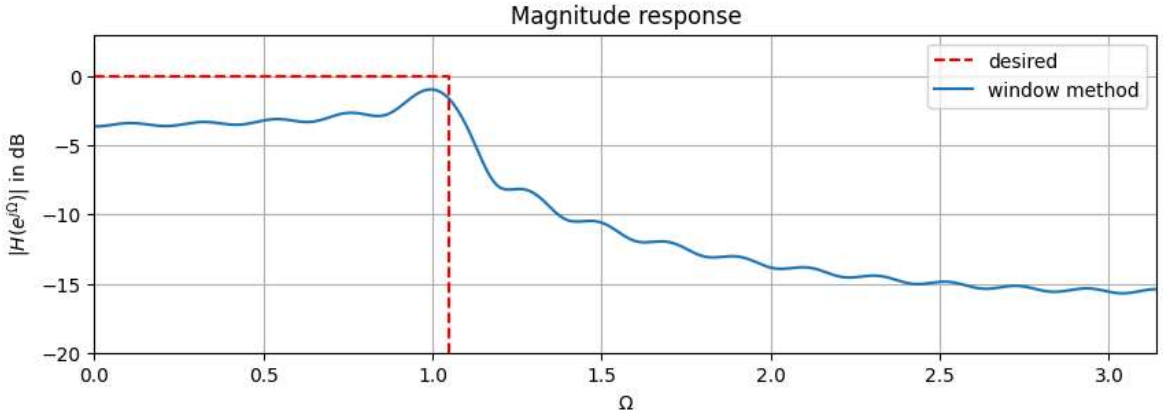
Non-recursive Filters design

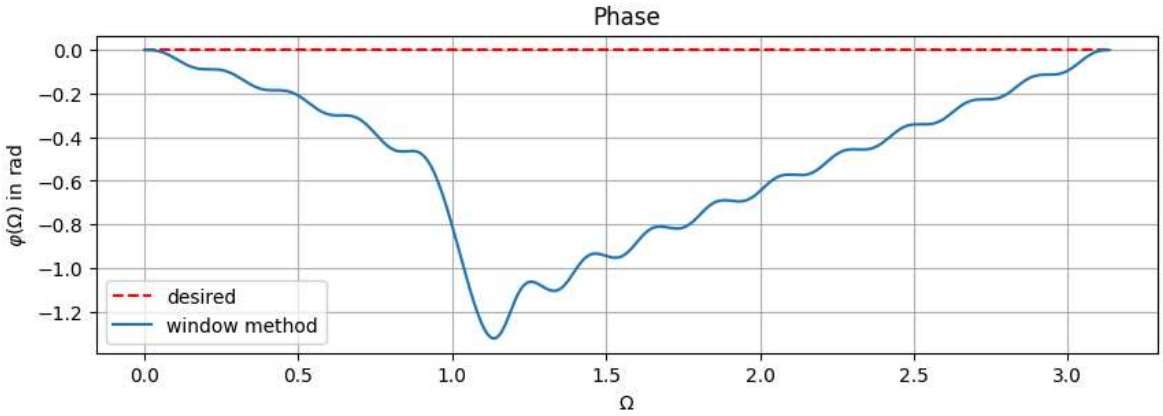
In [ ]:
```
import numpy as np
import matplotlib.pyplot as plt
import scipy.signal as sig
```

In [ ]:
```
N = 30  # length of filter
Omc = np.pi/3

# compute impulse response
k = np.arange(N)
hd = Omc/np.pi * np.sinc(k*Omc/np.pi)
# windowing
w = np.ones(N)
h = hd * w

# frequency response
Om, H = sig.freqz(h)

# plot impulse response
plt.figure(figsize=(10, 3))
plt.stem(h )
plt.title('Impulse response')
plt.xlabel(r'$k$')
plt.ylabel(r'$h[k]$')
# plot magnitude responses
plt.figure(figsize=(10, 3))
plt.plot([0, Omc, Omc], [0, 0, -100], 'r--', label='desired')
plt.plot(Om, 20 * np.log10(abs(H)), label='window method')
plt.title('Magnitude response')
plt.xlabel(r'$\Omega$')
plt.ylabel(r'$|H(e^{j \Omega})|$ in dB')
plt.axis([0, np.pi, -20, 3])
plt.grid()
plt.legend()
# plot phase responses
plt.figure(figsize=(10, 3))
plt.plot([0, Om[-1]], [0, 0], 'r--', label='desired')
plt.plot(Om, np.unwrap(np.angle(H)), label='window method')
plt.title('Phase')
plt.xlabel(r'$\Omega$')
plt.ylabel(r'$\varphi (\Omega)$ in rad')
plt.grid()
plt.legend()
```

Out[ ]:
```
<matplotlib.legend.Legend at 0x14eba952fd0>
```


Impulse response
h[k]
k


Magnitude response
desired
window method
|H(e^jΩ)| in dB
Ω


Phase
desired
window method
φ(Ω) in rad
Ω