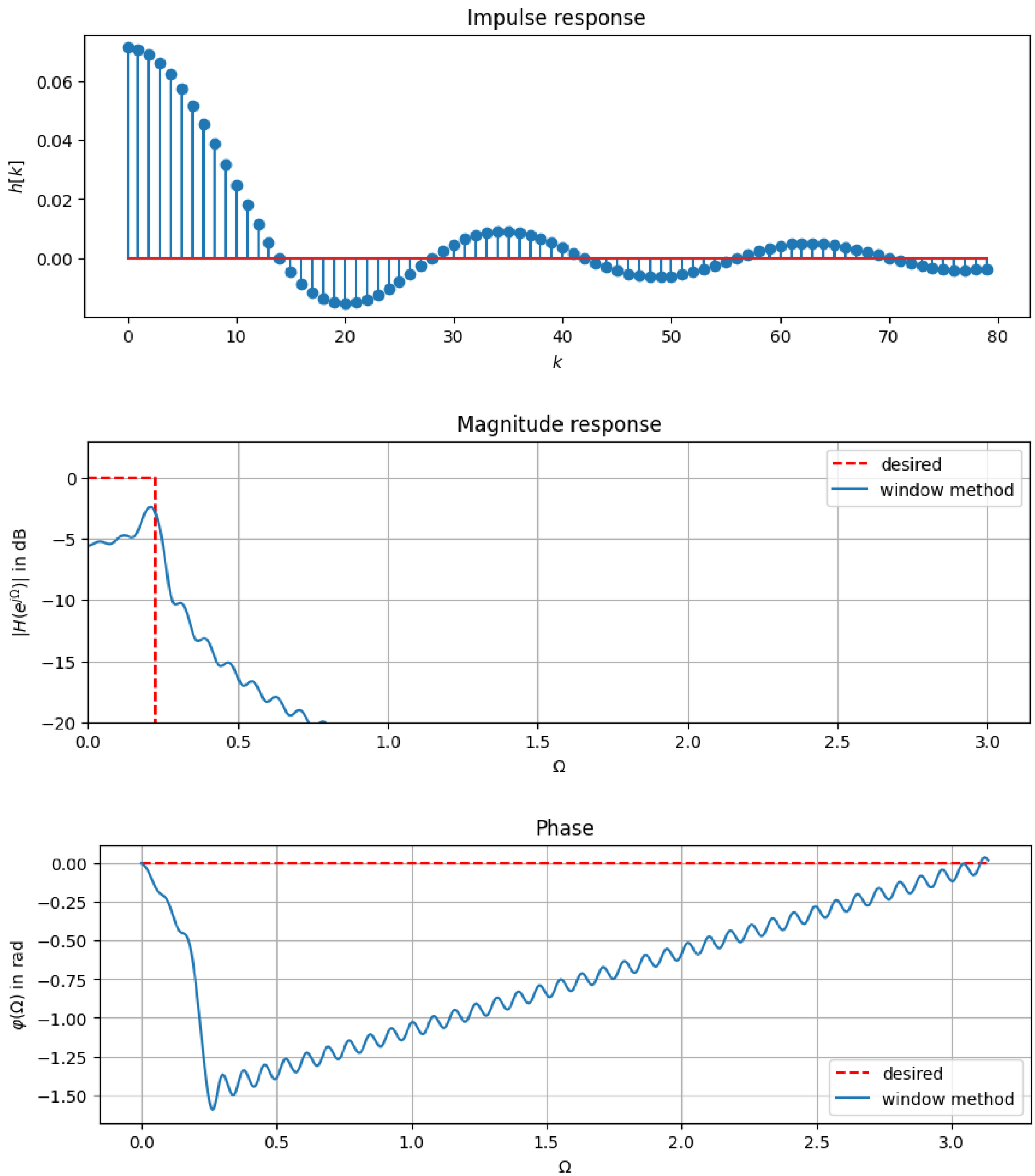
```python
import numpy as np
import matplotlib.pyplot as plt
import scipy.signal as sig
%matplotlib inline

N = 80  # length of filter
Omc = np.pi/14

# 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()
```

```
<matplotlib.legend.Legend at 0x1eacc15c260>
```



Impulse response
$h[k]$
0.06
0.04
0.02
0.00
0
10
20
30
40
50
60
70
80
k
Magnitude response
desired
window method
$|H(e^{j\Omega})|$ in dB
0
−5
−10
−15
−20
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Ω
Phase
$\varphi(\Omega)$ in rad
0.00
−0.25
−0.50
−0.75
−1.00
−1.25
−1.50
desired
window method
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Ω