

GTU ELM367 Sayısal İşaret İşlemenin Temelleri

freqz

$H(z)$ aşağıda gösterildiği gibi z 'nin negatif kuvvetlerinden oluşan TF-negative formu ile verildiğinde filtrenin frekans cevabını bulmak için `freqz()` fonksiyonu kullanılır:

$$H(z) = \frac{b_0 + b_1 z^{-1} + \dots + b_M z^{-M}}{a_0 + a_1 z^{-1} + \dots + a_N z^{-N}}$$

`scipy.signal.freqz(b, a=1, worN=512, whole=False, plot=None, fs=6.283185307179586, include_nyquist=False)`

```
In [38]: # Öncelikle gerekli kütüphaneleri yükleyiniz

import numpy as np
import scipy.signal as sgnl
import matplotlib.pyplot as plt
```

Örnek-1

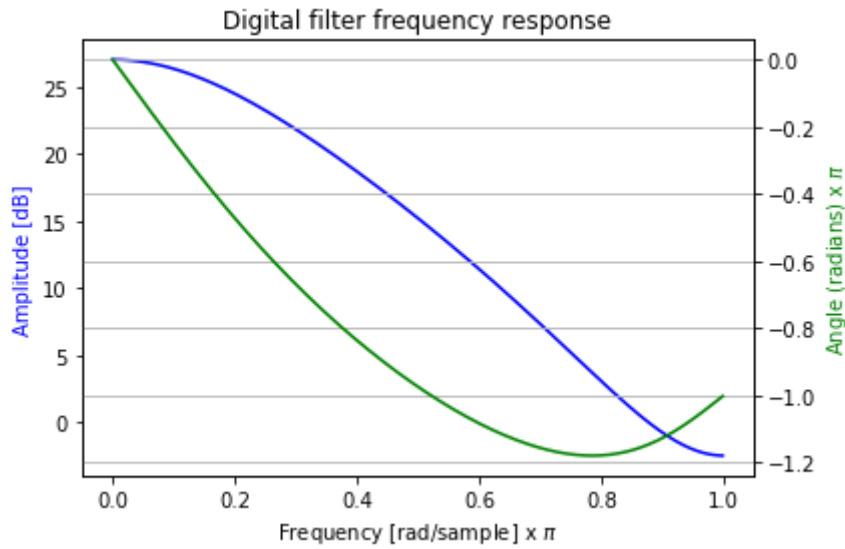
Aşağıda verilen filtrenin frekans cevabını çizdiriniz.

$$H(z) = \frac{5z^{-1} + 5z^{-2} + 1.25z^{-3}}{1 - \frac{7}{12}z^{-1} + \frac{1}{12}z^{-2}}$$

a ve b vektörlerini bu bilgiler ışığında elde edelim:

```
In [39]: b = np.array([0, 5, 5, 1.25])
a = np.array([1, -7.0/12, 1.0/12, 0])
w, h = sgnl.freqz(b, a)

fig, ax1 = plt.subplots()
ax1.set_title('Digital filter frequency response')
ax1.plot(w/np.pi, 20 * np.log10(abs(h)), 'b')
ax1.set_ylabel('Amplitude [dB]', color='b')
ax1.set_xlabel('Frequency [rad/sample] x $\pi$')
ax2 = ax1.twinx()
angles = np.unwrap(np.angle(h))
ax2.plot(w/np.pi, angles/np.pi, 'g')
ax2.set_ylabel('Angle (radians) x $\pi$', color='g')
ax2.grid()
ax2.axis('tight')
plt.show()
```



Örnek-2

5 ve 25 uzunluğunda bir pencereye sahip kayan ortalama filtresinin frekans cevabını çizdiriniz. Aşağıdaki sonuçlar incelendiğinde pencere uzunluğu arttıkça kesim frekansının azaldığını farkedebilirsiniz. Dolayısıyla, düşük bir kesim frekansı için daha uzun bir pencere kullanmanız gerekmektedir.

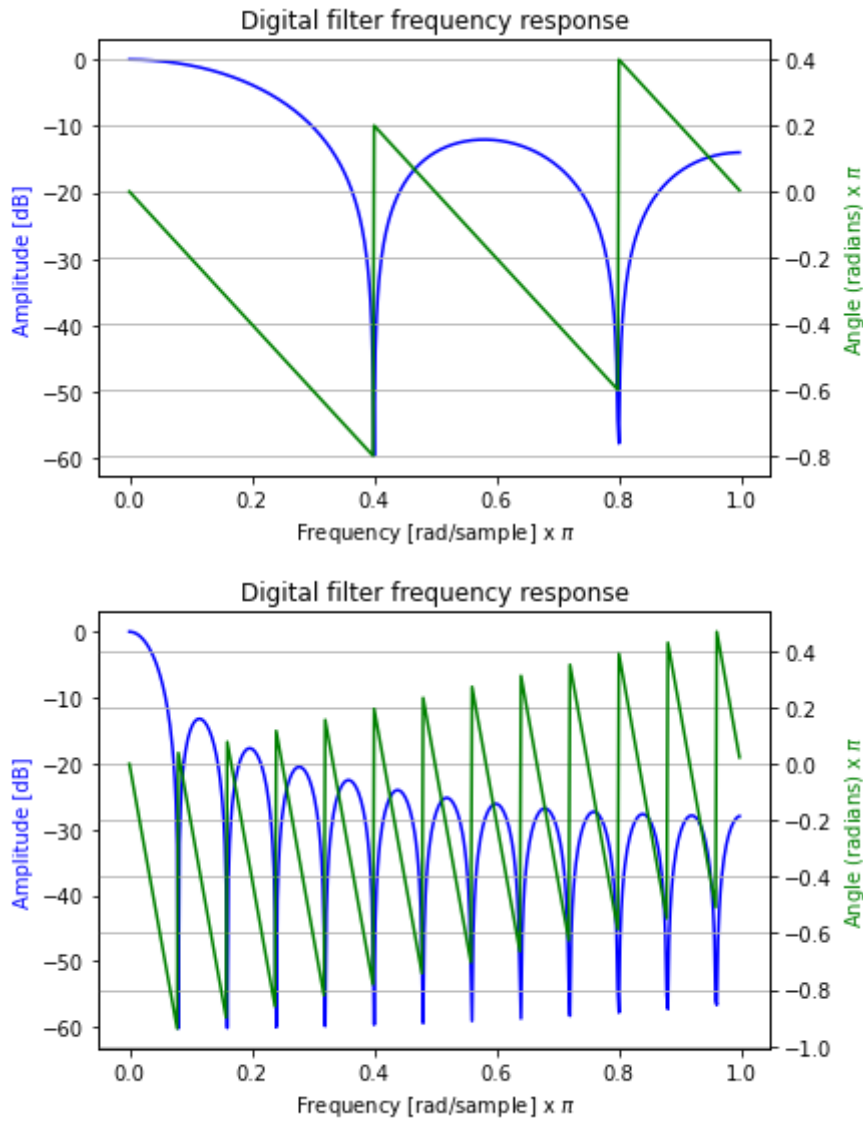
a ve b vektörlerini bu bilgiler ışığında elde edelim:

```
In [40]: b = np.ones(5)/5
a = np.array([1])
w, h = sgns.freqz(b, a)

fig, ax1 = plt.subplots()
ax1.set_title('Digital filter frequency response')
ax1.plot(w/np.pi, 20 * np.log10(abs(h)), 'b')
ax1.set_ylabel('Amplitude [dB]', color='b')
ax1.set_xlabel('Frequency [rad/sample] x  $\pi$ ')
ax2 = ax1.twinx()
angles = np.unwrap(np.angle(h))
ax2.plot(w/np.pi, angles/np.pi, 'g')
ax2.set_ylabel('Angle (radians) x  $\pi$ ', color='g')
ax2.grid()
ax2.axis('tight')
plt.show()

b = np.ones(25)/25
a = np.array([1])
w, h = sgns.freqz(b, a)

fig, ax1 = plt.subplots()
ax1.set_title('Digital filter frequency response')
ax1.plot(w/np.pi, 20 * np.log10(abs(h)), 'b')
ax1.set_ylabel('Amplitude [dB]', color='b')
ax1.set_xlabel('Frequency [rad/sample] x  $\pi$ ')
ax2 = ax1.twinx()
angles = np.unwrap(np.angle(h))
ax2.plot(w/np.pi, angles/np.pi, 'g')
ax2.set_ylabel('Angle (radians) x  $\pi$ ', color='g')
ax2.grid()
ax2.axis('tight')
plt.show()
```



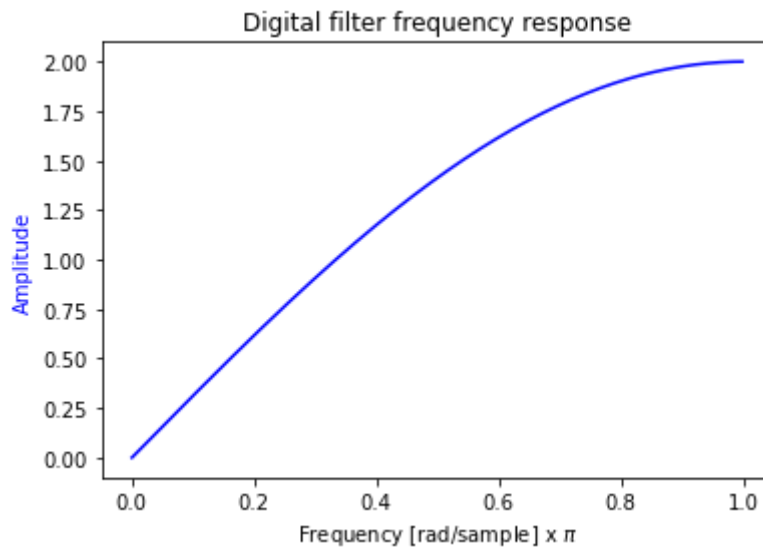
Örnek-3

Aşağıdaki filtre bir yüksek geçiren filtredir. Frekans cevabını çizdirip görelim:

$$H(z) = 1 - z^{-1}$$

```
In [41]: b = np.array([1,-1])
a = np.array([1])
w, h = signal.freqz(b, a)

fig, ax1 = plt.subplots()
ax1.set_title('Digital filter frequency response')
ax1.plot(w/np.pi, abs(h), 'b')
ax1.set_ylabel('Amplitude', color='b')
ax1.set_xlabel('Frequency [rad/sample] x  $\pi$ ')
plt.show()
```



Örnek-4

Aşağıdaki filtre Örnek-3'e göre kesim frekansı daha dar bir yüksek geçiren filtredir. Frekans cevabını çizdirip görelim:

$$H(z) = 1 - z^{-1} + z^{-2} - z^{-3} + z^{-4} - z^{-5}$$

```
In [42]: b = np.array([1,-1,1,-1,1,-1,1,-1])
a = np.array([1])
w, h = sgns.freqz(b, a)

fig, ax1 = plt.subplots()
ax1.set_title('Digital filter frequency response')
ax1.plot(w/np.pi, abs(h), 'b')
ax1.set_ylabel('Amplitude', color='b')
ax1.set_xlabel('Frequency [rad/sample] x $\pi$')
plt.show()
```

