



GEBZE TEKNİK ÜNİVERSİTESİ
ELEKTRONİK MÜHENDİSLİĞİ

ELM367

Sayısal İşaret İşlemenin Temelleri

Ödev 2

Bilgisayar Hesaplamaları

Hazırlayan
171024086 Berat KIZILARMUT

Ödev 2 Bilgisayar Hesaplamaları

Ad Soyad: Berat KIZILARMUT

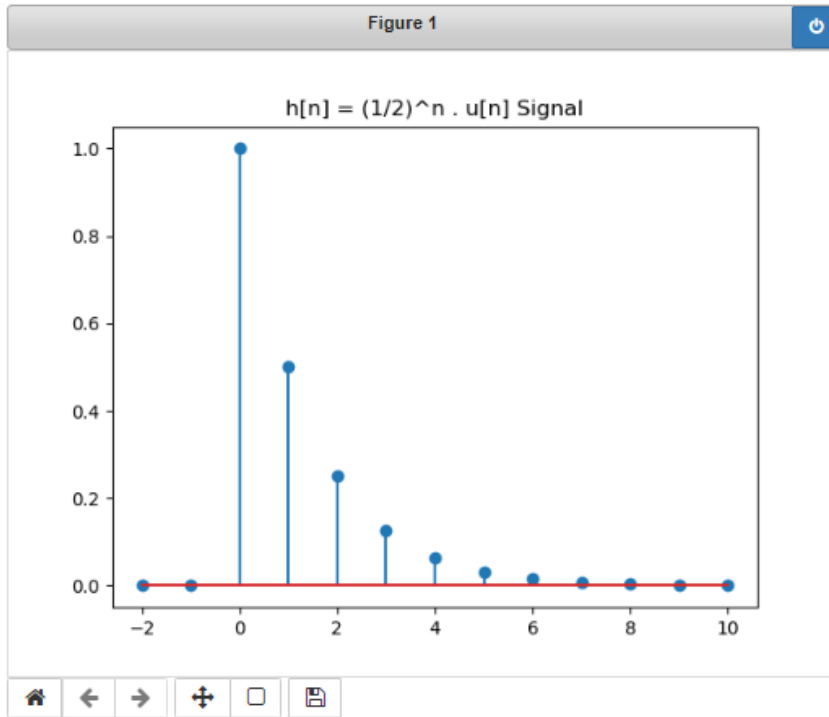
Öğrenci No: 171024086

```
In [1]: import numpy as np
import matplotlib.pyplot as plt
%matplotlib notebook
```

Soru 6

2.18 a)

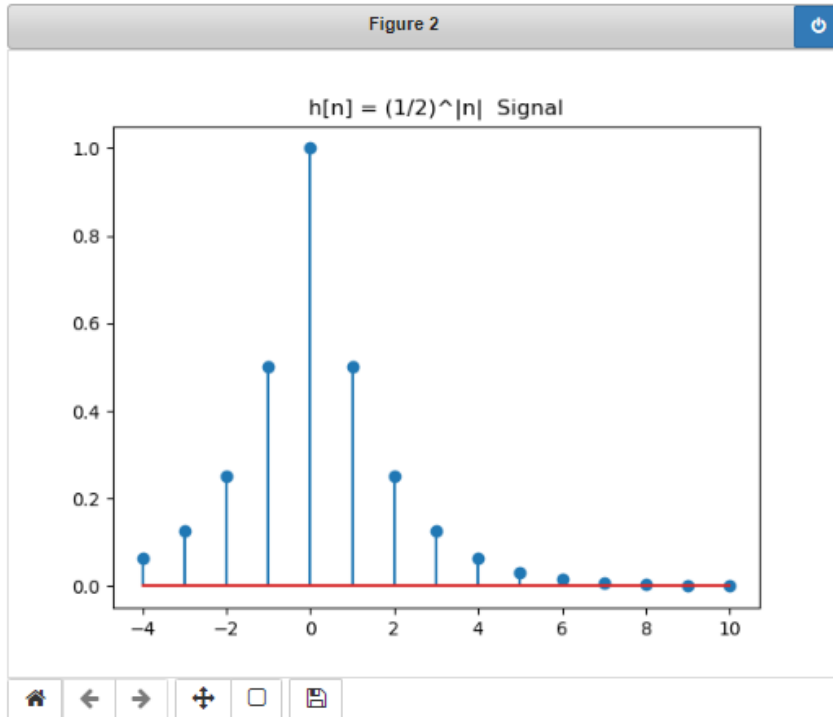
```
In [2]: # -2'den 11'e kadar 1'er değer ile artan bir dizi oluşturulması
n = np.arange(-2,11,1)
h = ((1/2)**n) * np.heaviside(n, 1)
# Ayırık zamanlı h işaretinin yazdırılması
plt.figure()
plt.stem(n,h)
plt.title('h[n] = (1/2)^n . u[n] Signal')
plt.show()
```



```
In [3]: # İşaretin sıfırdan önce değeri bulunmadığı için nedenseldir
```

2.18 c)

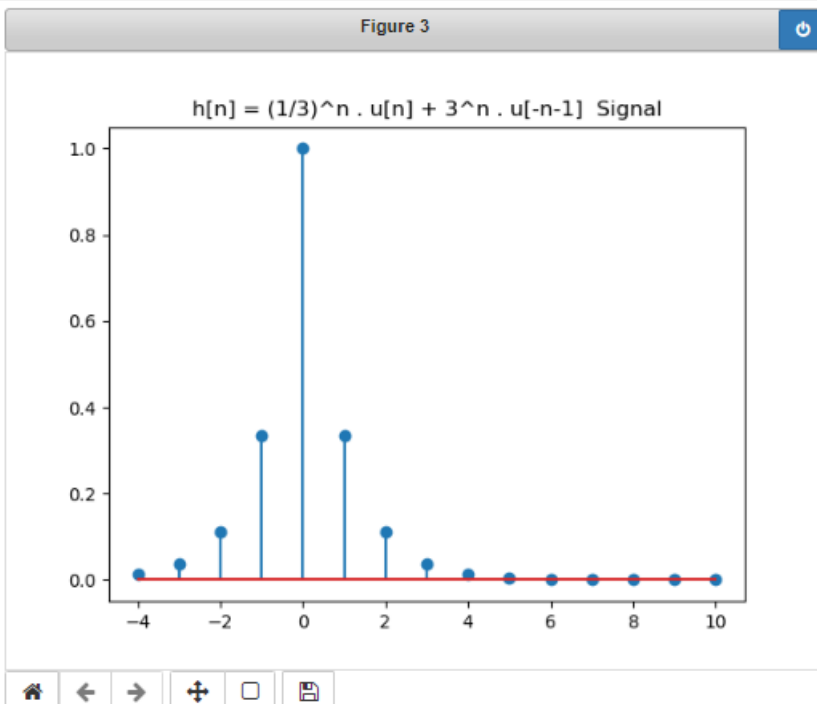
```
In [4]: # -4'den 11'e kadar 1'er değ er ile artan bir dizi oluřturulması
n = np.arange(-4,11,1)
h = ((1/2)**abs(n))
# Ayrık zamanlı h iřaretinin yazdırılması
plt.figure()
plt.stem(n,h)
plt.title('h[n] = (1/2)^|n|  Signal')
plt.show()
```



```
In [5]: # iřaret sıfırdan  nce değ er aldıđı i in bu iřaret nedensel deđildir
# iřaret simetrik bir řekilde oluřmaktadır
```

2.18 e)

```
In [6]: # -4'den 11'e kadar 1'er değ er ile artan bir dizi oluřturulması
n = np.arange(-4,11,1)
h = ((1/3)**n) * np.heaviside(n, 1) + (3**n) * np.heaviside(-n-1,1)
# Ayrık zamanlı h iřaretinin yazdırılması
plt.figure()
plt.stem(n,h)
plt.title('h[n] = (1/3)^n . u[n] + 3^n . u[-n-1]  Signal')
plt.show()
```

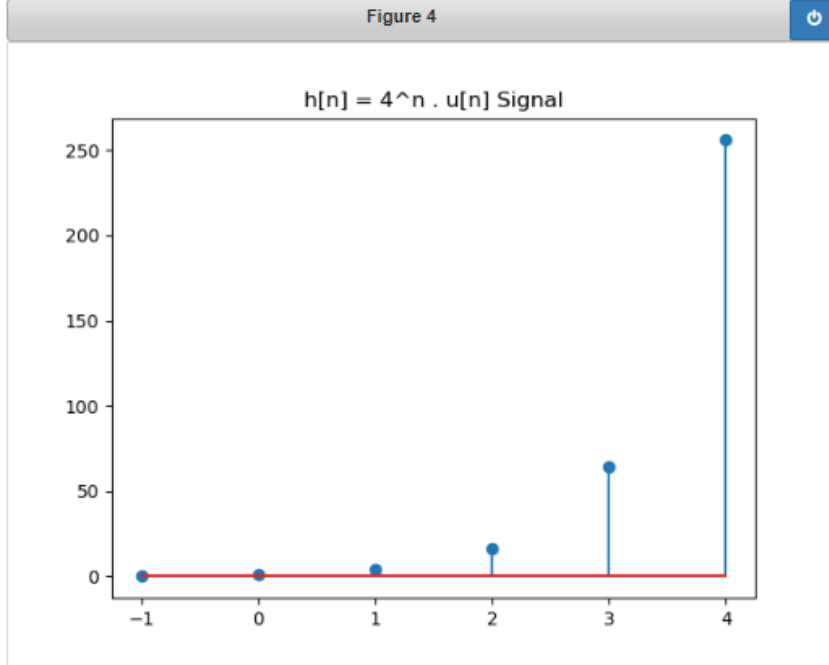


```
In [7]: # iřaret sıfırdan  nce değ er aldıđı i in bu iřaret nedensel deđildir
```

Soru 7

2.19 a)

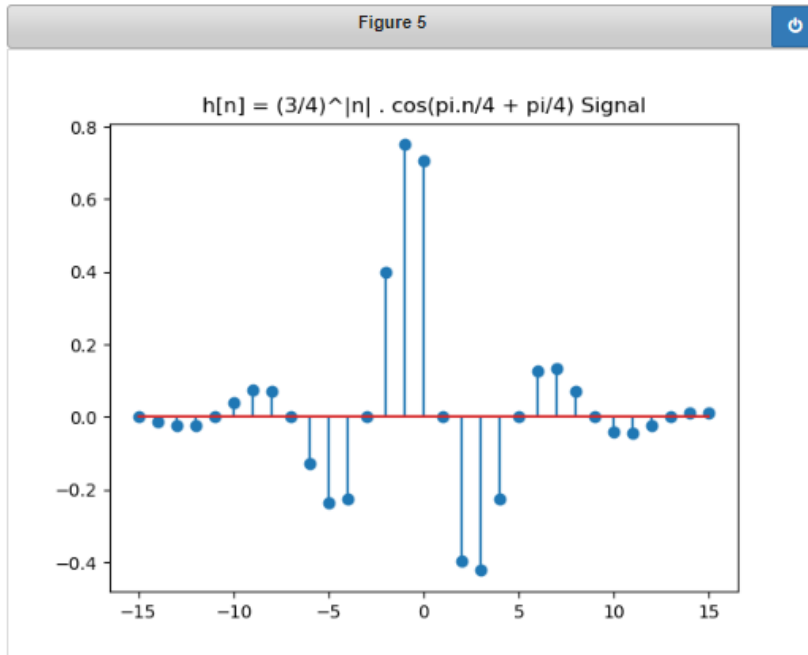
```
In [8]: # -1'den 5'e kadar 1'er değ er ile artan bir dizi oluřturulması
n = np.arange(-1,5,1)
h = (4.**n) * np.heaviside(n, 1)
# Ayrık zamanlı h iřaretinin yazdırılması
plt.figure()
plt.stem(n,h)
plt.title('h[n] = 4^n . u[n] Signal')
plt.show()
```



```
In [9]: # İřaret s rekli  stel bir artıř g sterdiđi i in stabil deđildir
```

2.19 e)

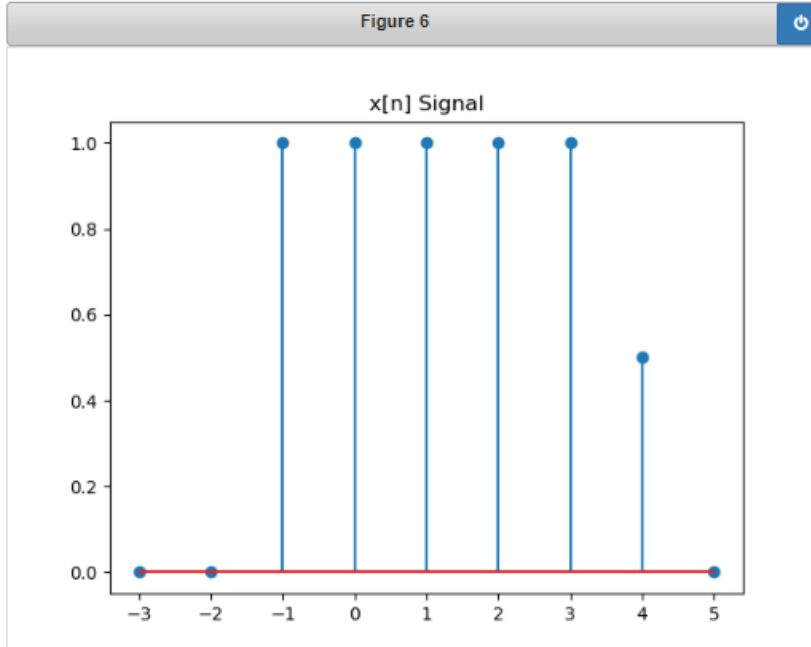
```
In [10]: # -15'den 16'ya kadar 1'er değ er ile artan bir dizi oluřturulması
n = np.arange(-15,16,1)
h = ((3/4)**abs(n)) * np.cos((np.pi*n/4) + (np.pi/4))
plt.figure()
# Ayrık zamanlı h iřaretinin yazdırılması
plt.stem(n,h)
plt.title('h[n] = (3/4)^|n| . cos(pi.n/4 + pi/4) Signal')
plt.show()
```



```
In [11]: # İşaret bir süre sonra stabilleşmektedir
# Sistem stabildir
```

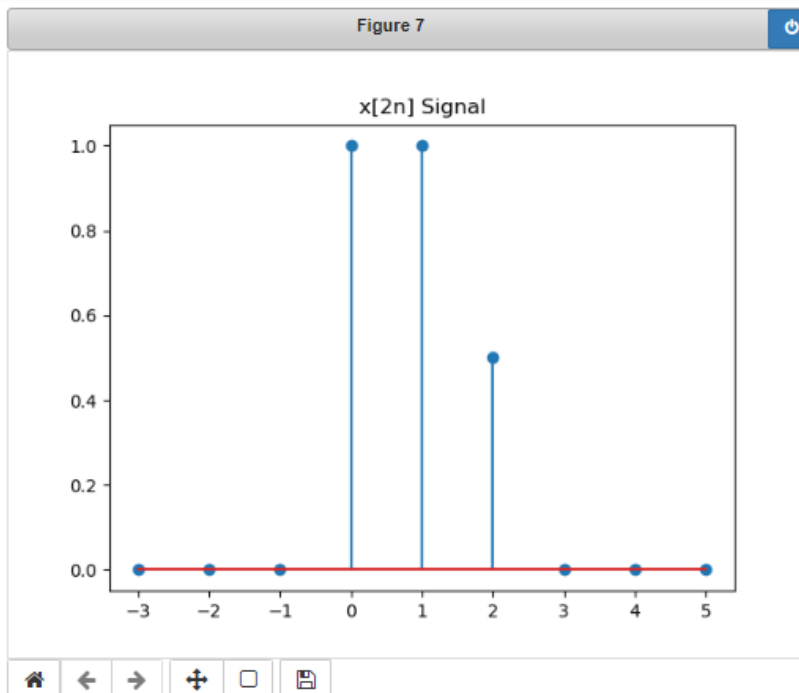
Soru 8

```
In [12]: # x[n] işaretinin kendisi
n = np.arange(-3,6,1)
x = [0,0,1,1,1,1,1,1/2,0]
plt.figure()
plt.stem(n,x)
plt.title('x[n] Signal')
plt.show()
```



2.21 c)

```
In [13]: # x[2n] işaretinin
n = np.arange(-3,6,1)
x = [0,0,0,1,1,1/2,0,0,0]
plt.figure()
plt.stem(n,x)
plt.title('x[2n] Signal')
plt.show()
```



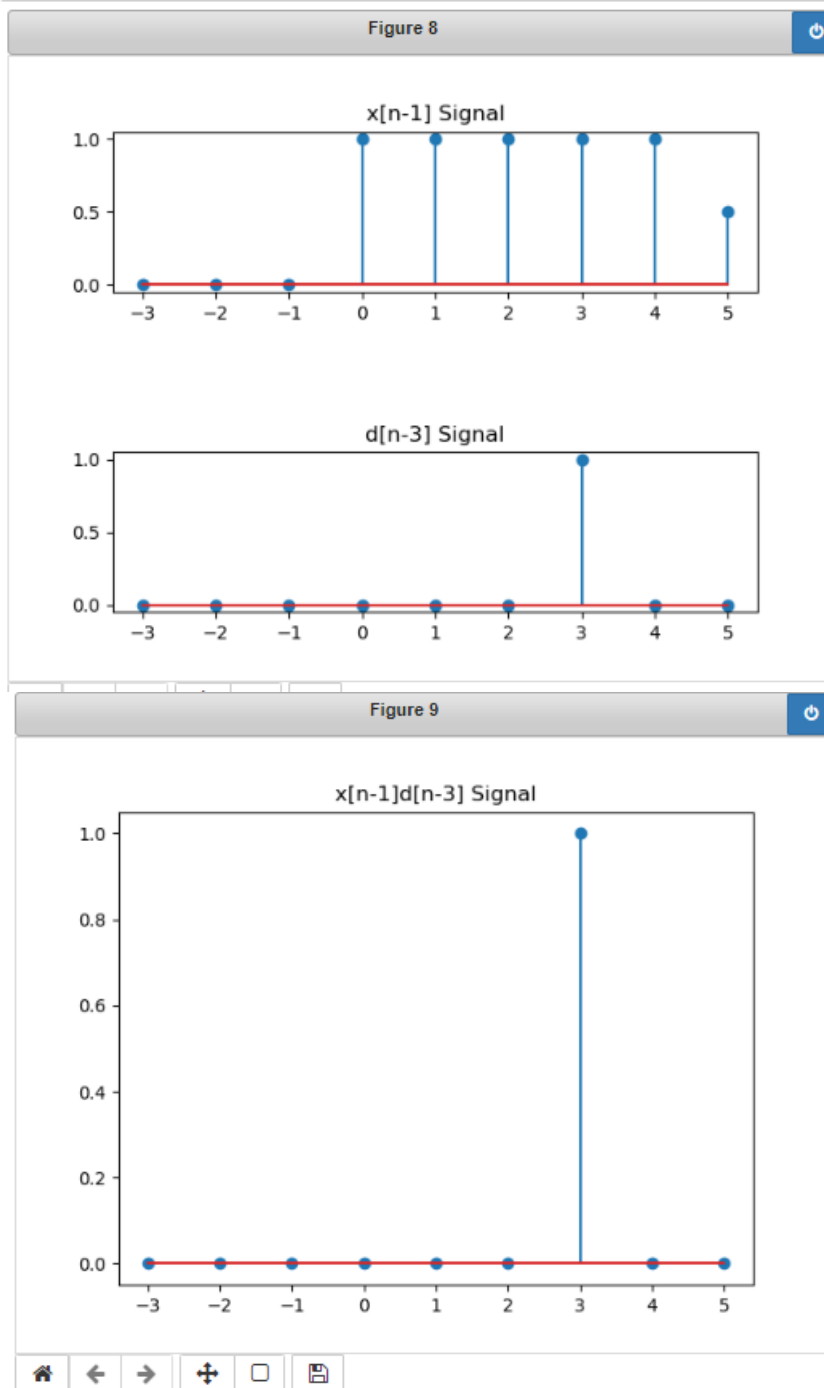
```
In [14]: # İşaretin orantılanması değişmiştir
# İşaret kısalmıştır
```

2.21 e)

```
In [15]: n = np.arange(-3,6,1)
# x[n-1] işaretinin kendisi
x = np.array([0,0,0,1,1,1,1,1,1/2])
# dirac[n-3] işareti
d = np.array([0,0,0,0,0,0,0,1,0,0])
# x[n-1]d[n-3] işareti
t = x * d

fig = plt.figure()
# Subplotlar arası mesafe eklenildi
plt.subplots_adjust(hspace=1)
ax1=fig.add_subplot(211)
ax2=fig.add_subplot(212)
# Subplotlara yazıldı
ax1.stem(n,x)
ax1.title.set_text('x[n-1] Signal')
ax2.stem(n,d)
ax2.title.set_text('d[n-3] Signal')

plt.figure()
plt.stem(n,t)
plt.title('x[n-1]d[n-3] Signal')
plt.show()
```



```
In [16]: # Ana işaret öncelikle ötelenmiştir
# Ardından dirac işareti ötelenmiş ve bu iki işaret birbiri ile çarpılmıştır
# Adımlar sırasıyla gösterilmiştir
```