

İşaret İşleme

Ayrık Zamanda Matlab uygulamalar- H6CD3

Dr. Meriç Çetin
versiyon191020

Ayrık Zamanda Birim Basamak ve Birim Darbe Sinyalleri

%delta function

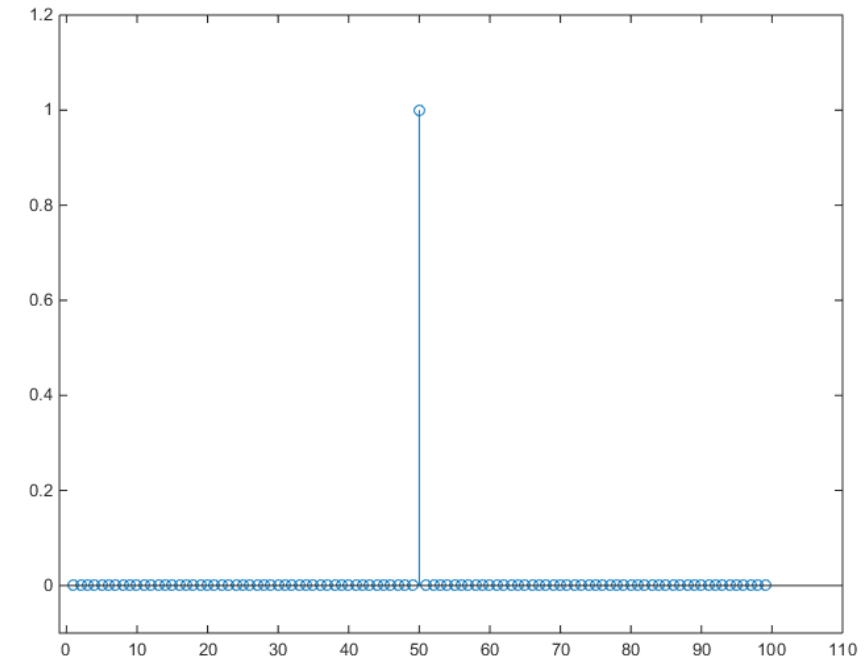
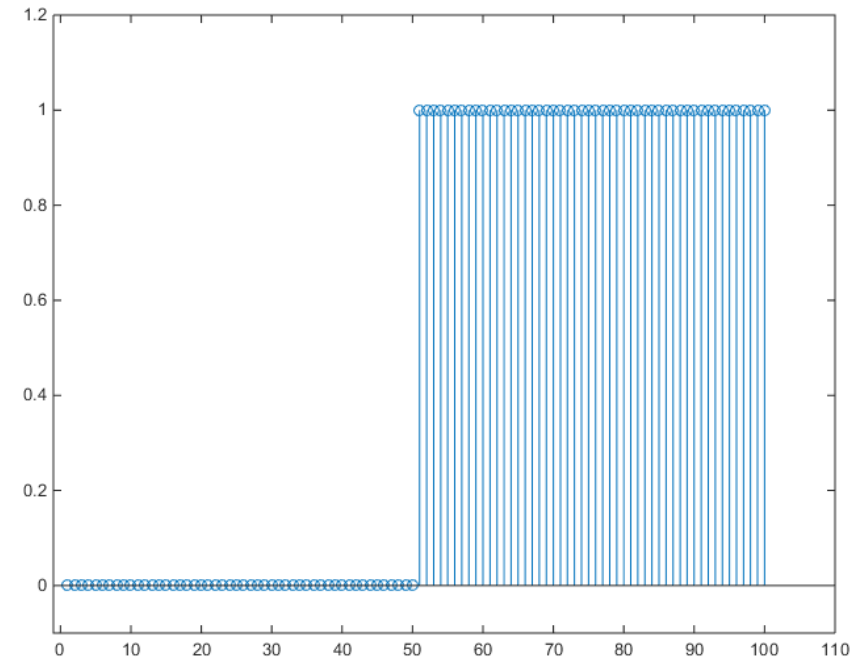
```
u = [zeros(1,50),ones(1,50)];
```

%step function

```
delta = [zeros(1,49),1, zeros(1,49)];
```

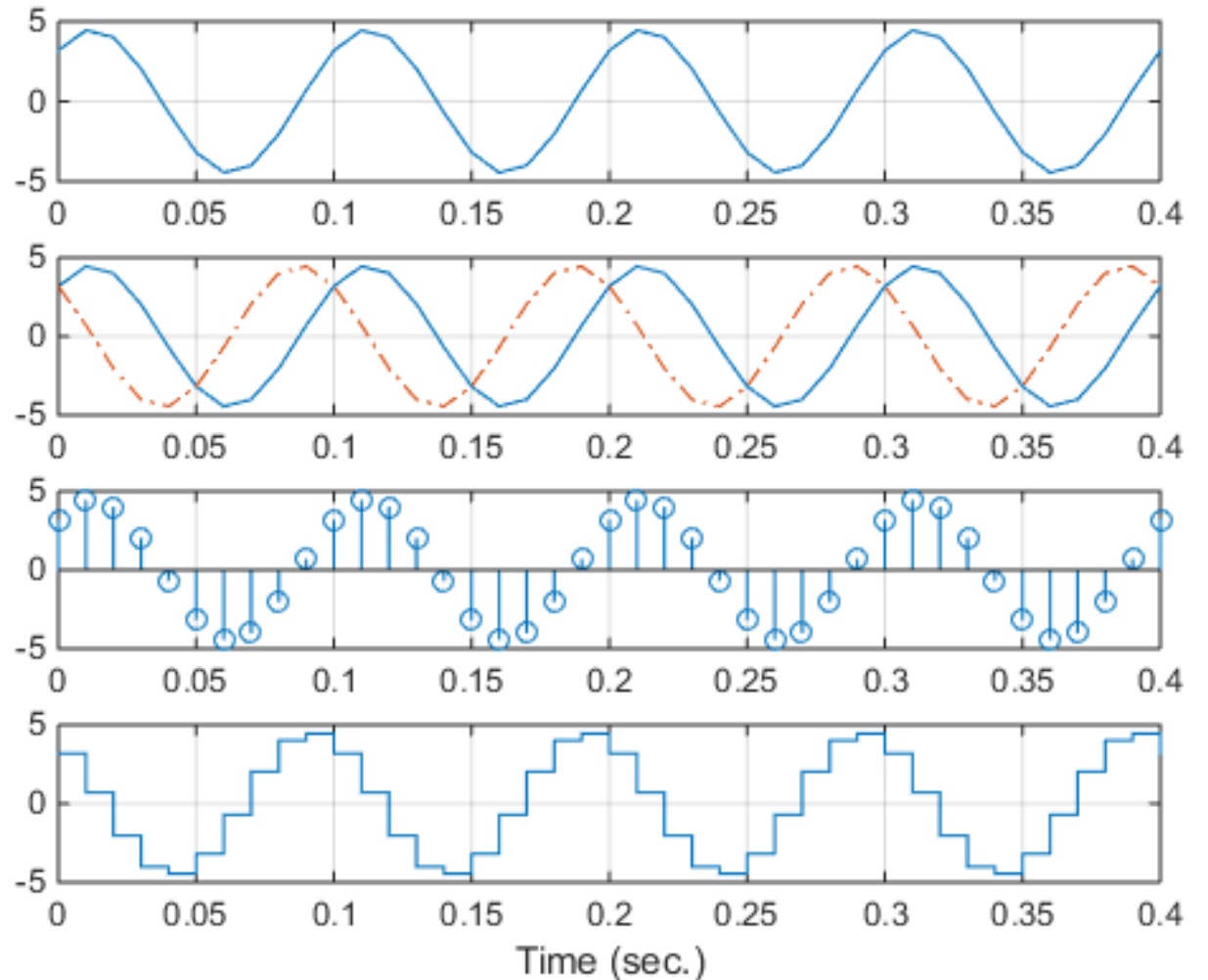
```
figure,stem(delta); axis([-1 110 -0.1 1.2])
```

```
figure,stem(u); axis([-1 110 -0.1 1.2])
```



Ayrık zamanda sinyal oluşturma örnekleri

```
t=0:0.01:0.4;  
xx = 4.5*sin(2*pi*10*t + pi/4);  
yy = 4.5*cos(2*pi*10*t + pi/4);  
subplot(4,1,1), plot(t,xx);grid  
subplot(4,1,2), plot(t,xx,t,yy,'-');grid  
subplot(4,1,3), stem(t,xx);grid  
subplot(4,1,4), stairs(t,yy);grid  
xlabel('Time (sec.)');
```



Ayrık zamanda sinyal oluşturma örnekleri

```
n=0:5;
```

```
xa = (-0.5).^n;
```

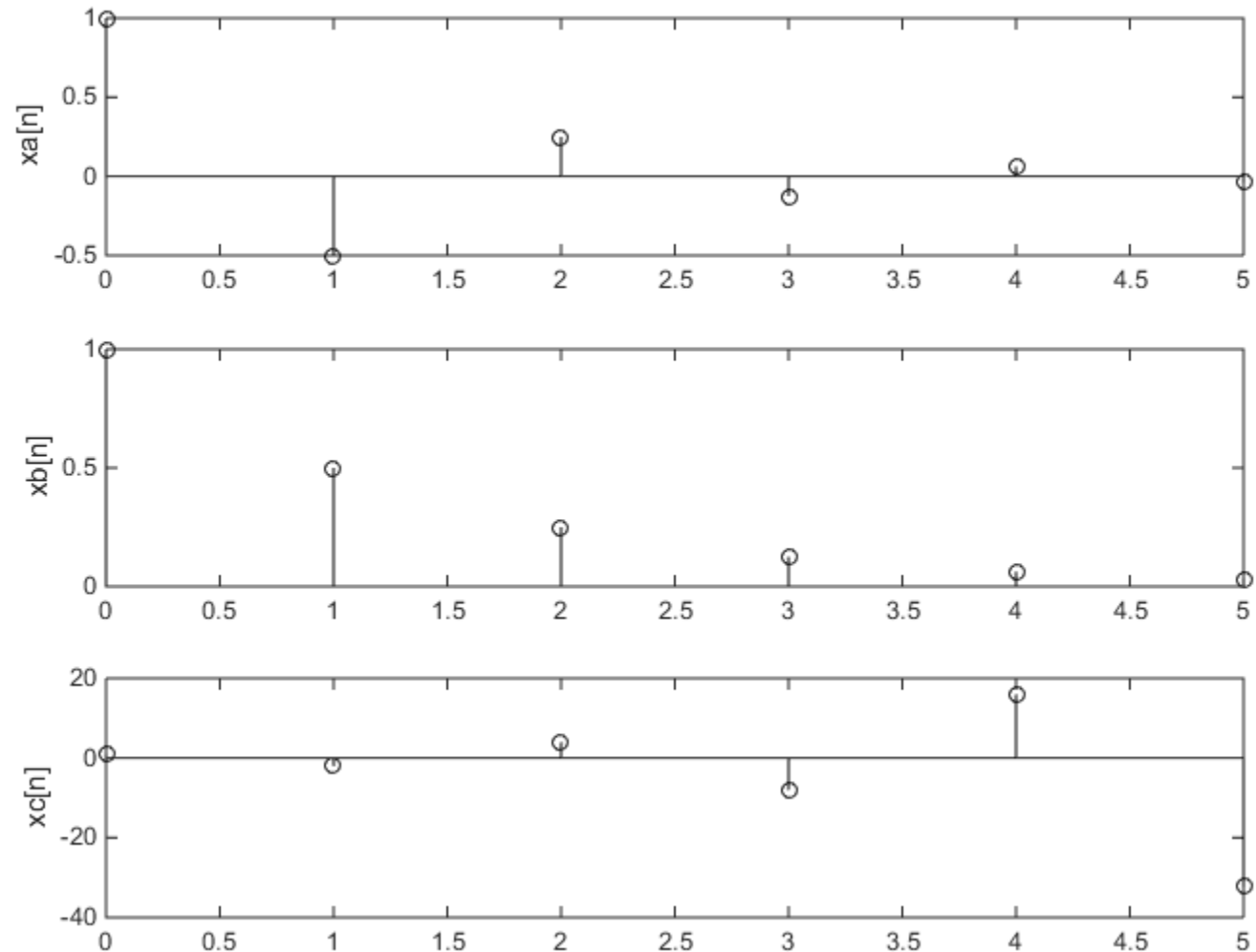
```
xb = (2).^(-n);
```

```
xc = (-2).^n;
```

```
subplot(311),stem(n,xa,'k');ylabel('xa[n]')
```

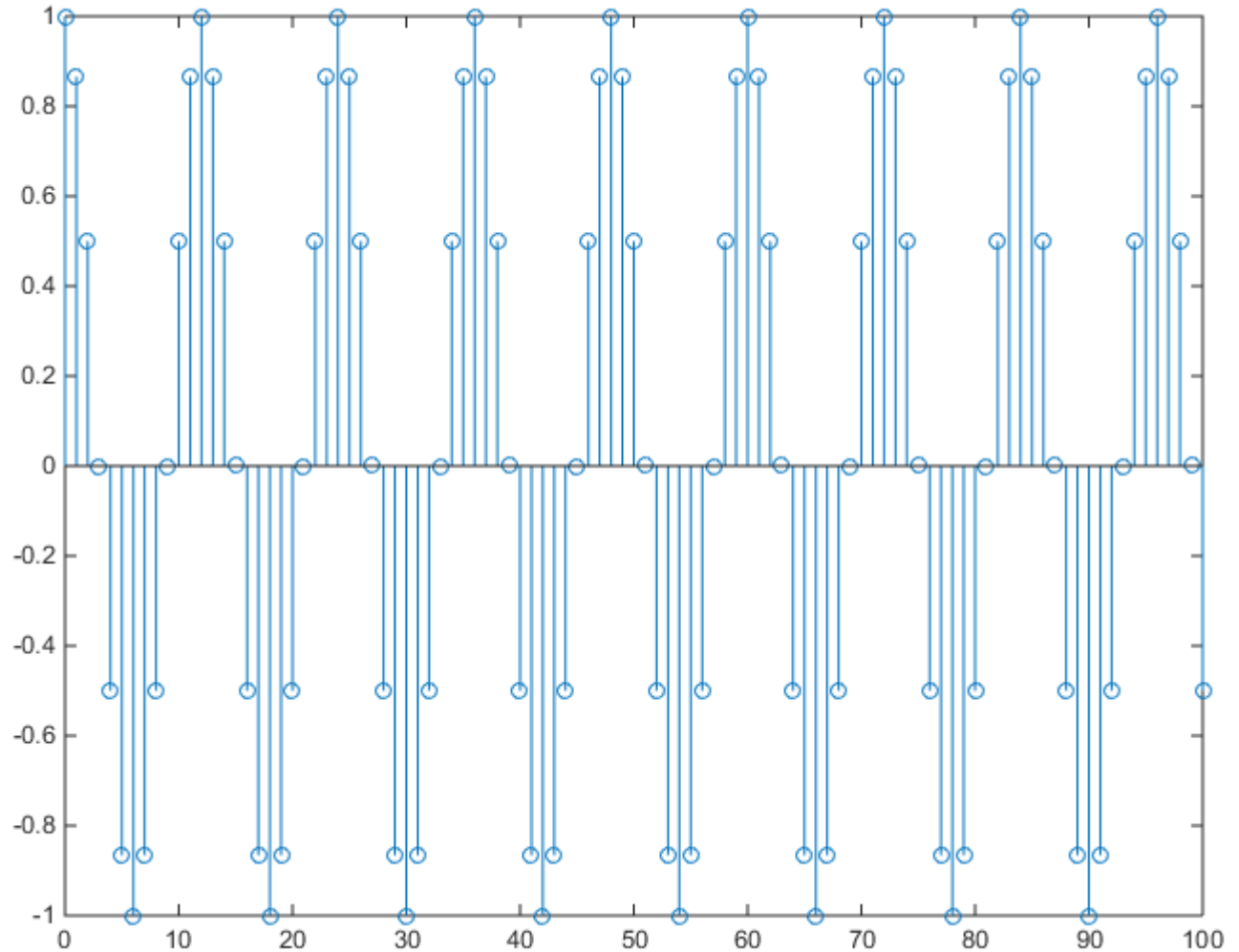
```
subplot(312),stem(n,xb,'k');ylabel('xb[n]')
```

```
subplot(313),stem(n,xc,'k');ylabel('xc[n]')
```



Ayrık zamanlı bir sinyalin enerji hesabına örnek

```
n=[0:100];  
x=exp(j*pi/6.*n);  
Ex2=sum(abs(x).^2)  
stem(n,x)
```



Ayrık zamanlı bir sistemin sıfır giriş cevabı (y_{sgc})

COMPUTER EXAMPLE C3.4

Using the initial conditions $y[-1] = 2$ and $y[-2] = 1$, find and sketch the zero-input response for the system described by

$$(E^2 - 1.56E + 0.81)y[n] = (E + 3)x[n].$$



System Stability

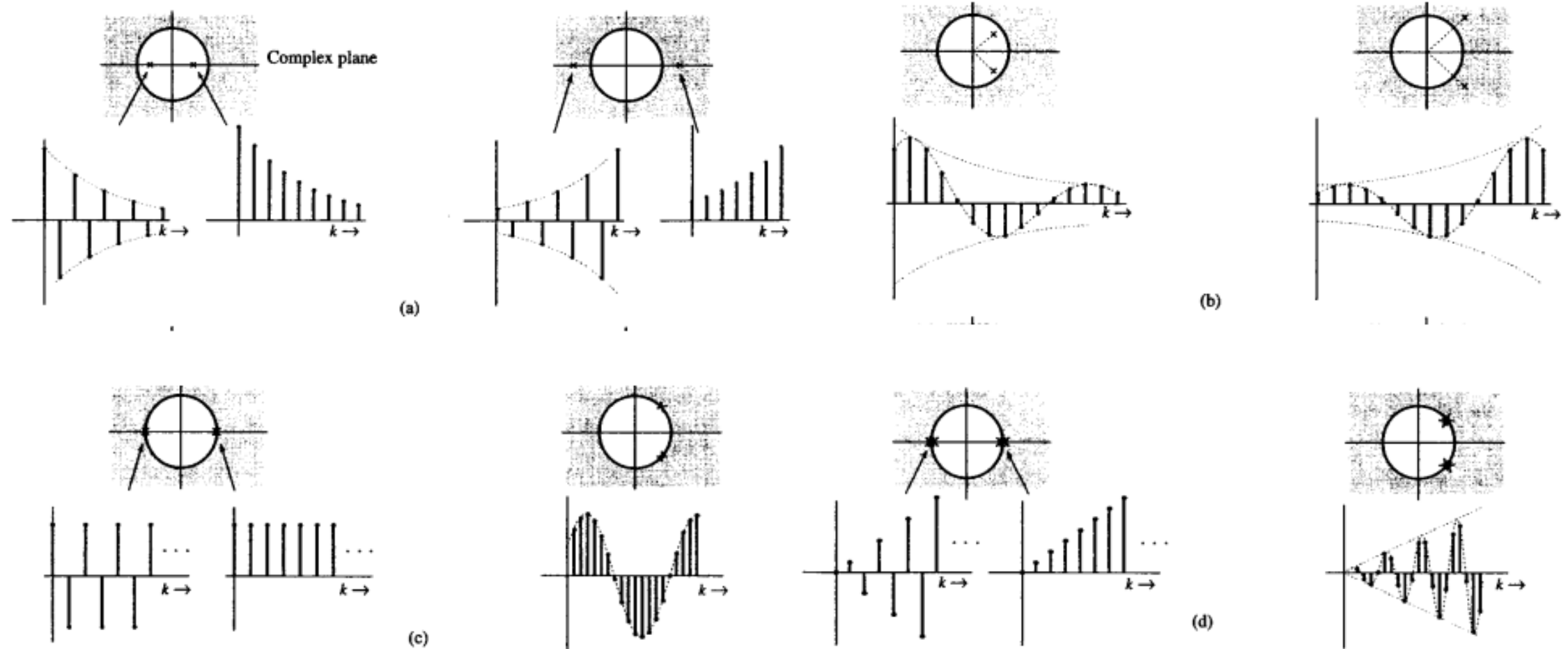


Fig. Characteristic roots location and the corresponding characteristic modes.

Ayrık zamanda y_{SGC}

$$(E^2 - 1.56E + 0.81)y[n] = (E + 3)x[n].$$

initial conditions $y[-1] = 2$ and $y[-2] = 1$,

```
clc, clear all, close all
```

```
n = (-2:20)';
```

```
y = [1;2;zeros(length(n)-2,1)];
```

```
for k = 1:length(n)-2
```

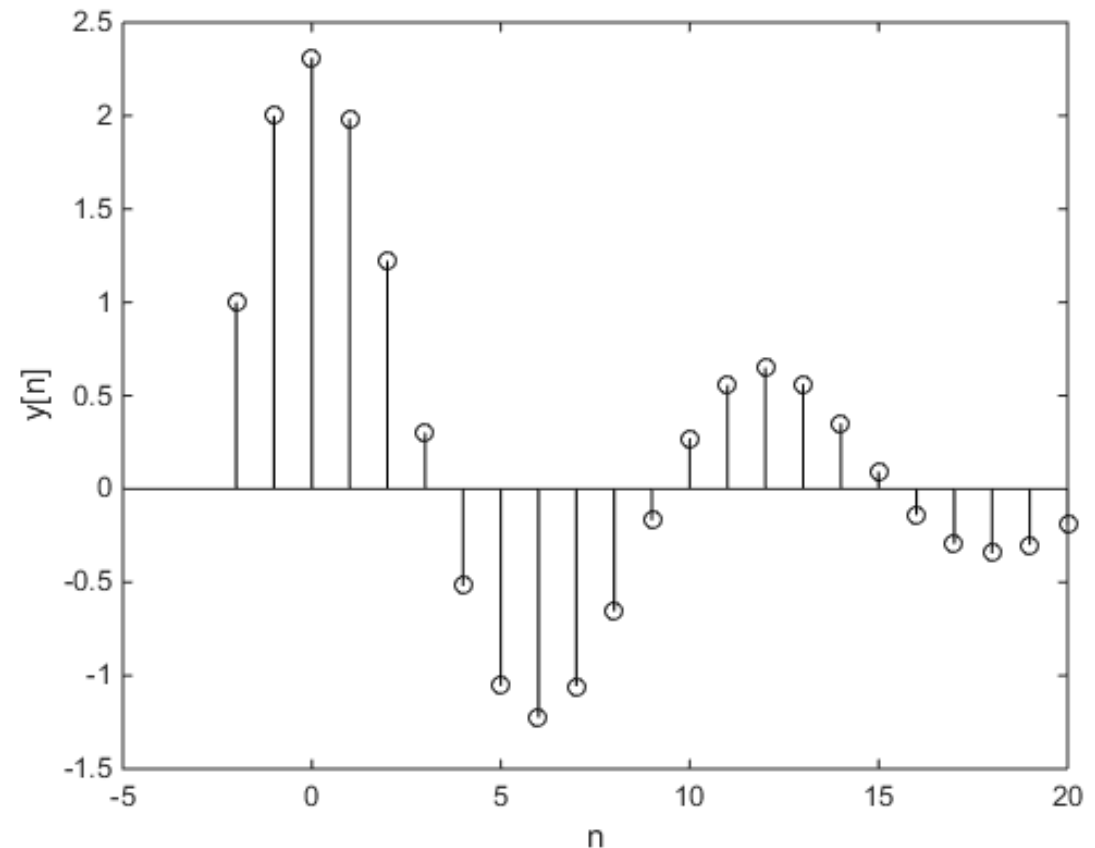
```
    y(k+2)=1.56*y(k+1)-0.81*y(k);
```

```
end
```

```
stem(n,y,'k');xlabel('n');ylabel('y[n]')
```

```
disp('n      y');
```

```
disp([num2str([n,y]))]);
```



filter()

A routine called `filter` is available to solve difference equations numerically, given the input and the difference equation coefficients. In its simplest form this routine is invoked by

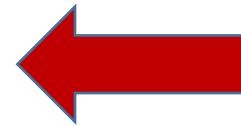
```
y = filter(b,a,x)
```

where

```
b = [b0, b1, ..., bM]; a = [a0, a1, ..., aN];
```

Given the following difference equation $y(n] - y[n - 1] + 0.9y[n - 2] = x[n]; \quad \forall n$

- Calculate and plot the impulse response $h[n]$ at $n = -20, \dots, 100$.
- Calculate and plot the unit step response $s[n]$ at $n = -20, \dots, 100$.

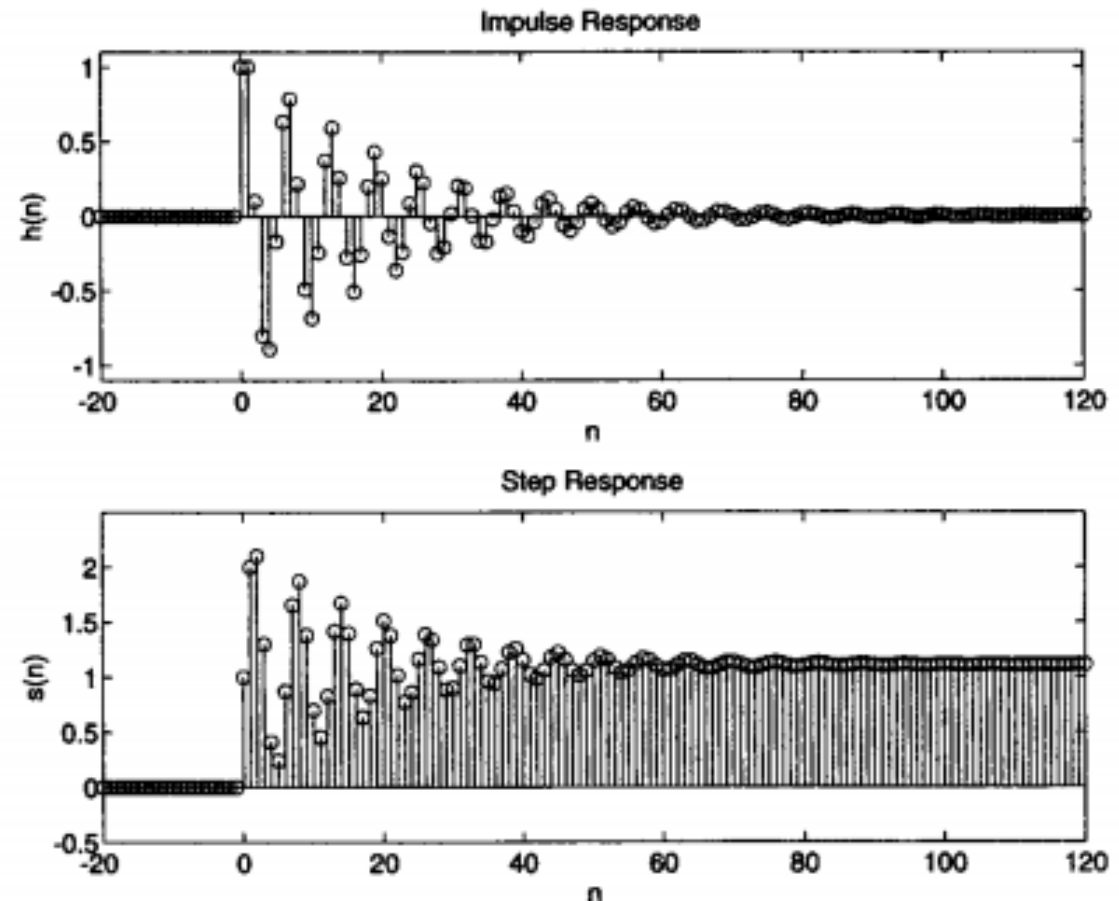


filter()

```
clc,clear all,  
num=[1]; den=[1 -1 0.9];n=0:120;  
x=impseq(0,0,120);  
h=filter(num,den,x);  
subplot(211),stem(n,h), title('unit impulse response');
```

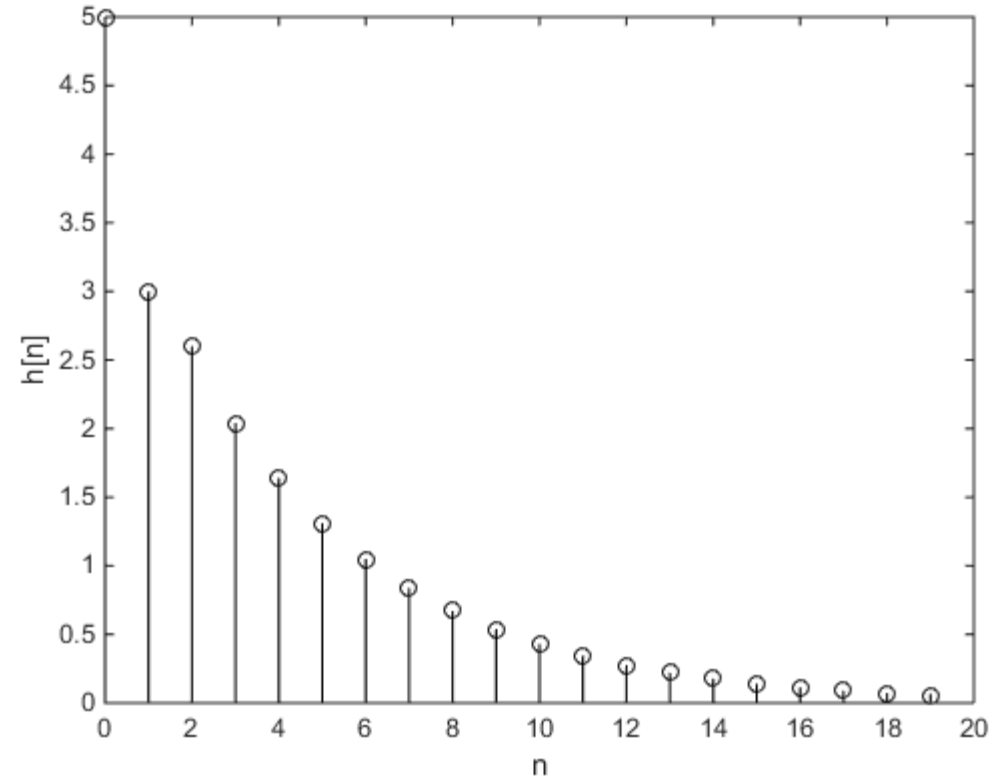
```
num=[1]; den=[1 -1 0.9];n=0:120;  
z=stepseq(0,0,120);  
h=filter(num,den,z);  
subplot(212),stem(n,h), title('unit step response');
```

```
function [x,n]=stepseq(n0,n1,n2)  
n = [n1:n2];  
x = [(n-n0)>=0];  
function [x,n]=impseq(n0,n1,n2)  
n = [n1:n2];  
x = [(n-n0)==0];
```



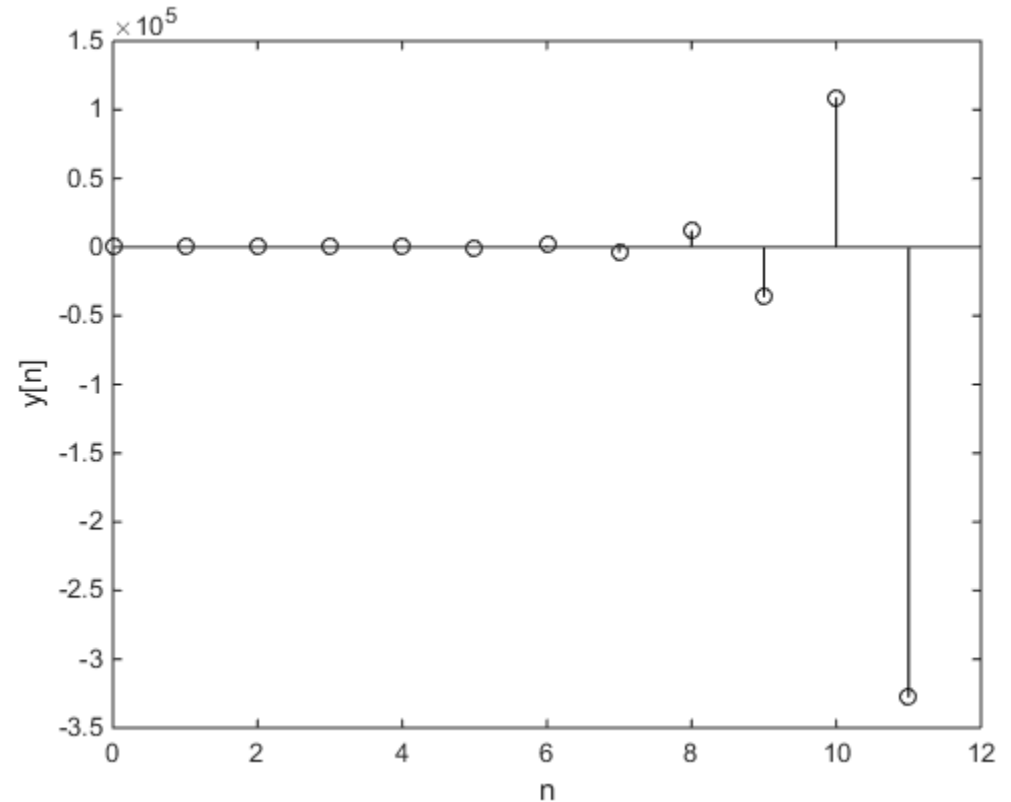
Filter() komutu ile ayrik zamanda birim darbe cevabi

```
n = (0:19);  
x = inline('n==0');  
a=[1 -0.6 -0.16];  
b=[5 0 0];  
h=filter(b,a,x(n));  
  
stem(n,h,'k');  
xlabel('n');ylabel('h[n]')  
disp('n          h');  
disp([num2str([n',h'])]);
```



Ayrık zamanlı bir sistemin sıfır durum cevabı (y_{SDC})

```
n = (0:11);  
x = inline('(4.^(-n)).*(n>=0)');  
a=[1 6 9];  
b=[2 6 0];  
h=filter(b,a,x(n));  
stem(n,h,'k');xlabel('n');ylabel('y[n]')
```

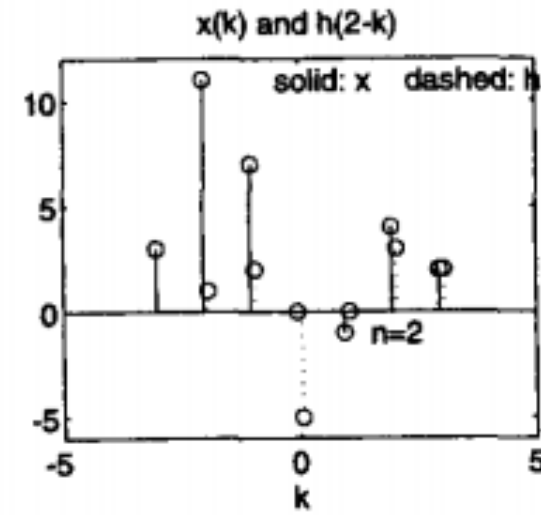
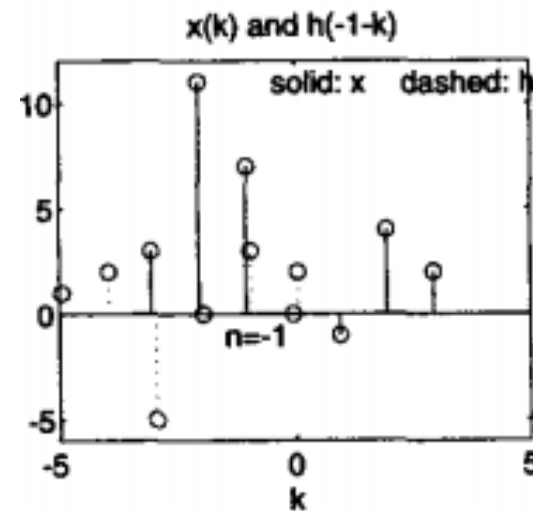
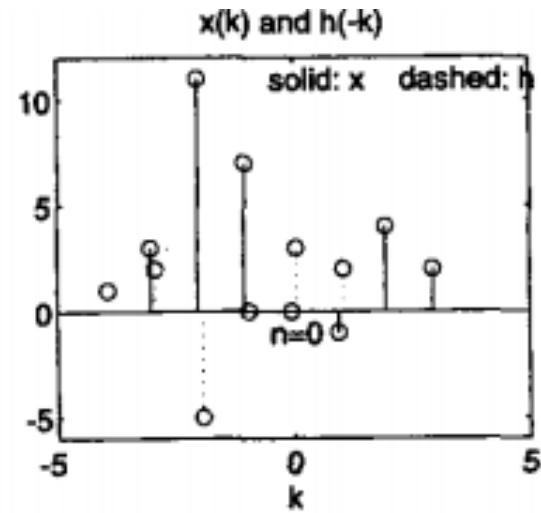
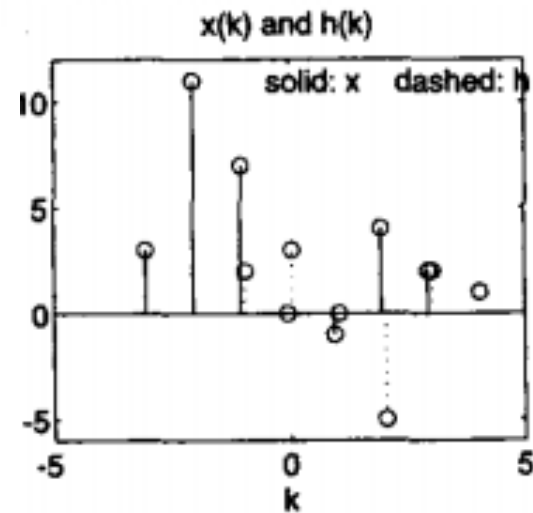


Ayrık Zamanda Konvolüsyon

$$x(n) = \begin{bmatrix} 3, 11, 7, 0, -1, 4, 2 \end{bmatrix}, \quad -3 \leq n \leq 3; \quad h(n) = \begin{bmatrix} 2, 3, 0, -5, 2, 1 \end{bmatrix}, \quad -1 \leq n \leq 4$$

\uparrow \uparrow

determine the convolution $y(n) = x(n) * h(n)$.



of $h(k)$. The bottom-left plot shows $x(k)$ and $h(-1-k)$, the folded-and-shifted-by-1 version of $h(k)$. Then

$$\sum_k x(k)h(-1-k) = 3 \times (-5) + 11 \times 0 + 7 \times 3 + 0 \times 2 = 6 = y(-1)$$

The bottom-right plot shows $x(k)$ and $h(2-k)$, the folded-and-shifted-by-2 version of $h(k)$, which gives

$$\sum_k x(k)h(2-k) = 11 \times 1 + 7 \times 2 + 0 \times (-5) + (-1) \times 0 + 4 \times 3 + 2 \times 2 = 41 = y(2)$$

Thus we have obtained two values of $y(n)$. Similar graphical calculations can be done for other remaining values of $y(n)$. Note that the beginning point (first nonzero sample) of $y(n)$ is given by $n = -3 + (-1) = -4$, while the end point (the last nonzero sample) is given by $n = 3 + 4 = 7$. The complete output is given by

$$y(n) = \left\{ 6, 31, 47, 6, \underset{\uparrow}{-51}, -5, 41, 18, -22, -3, 8, 2 \right\}$$



clear all

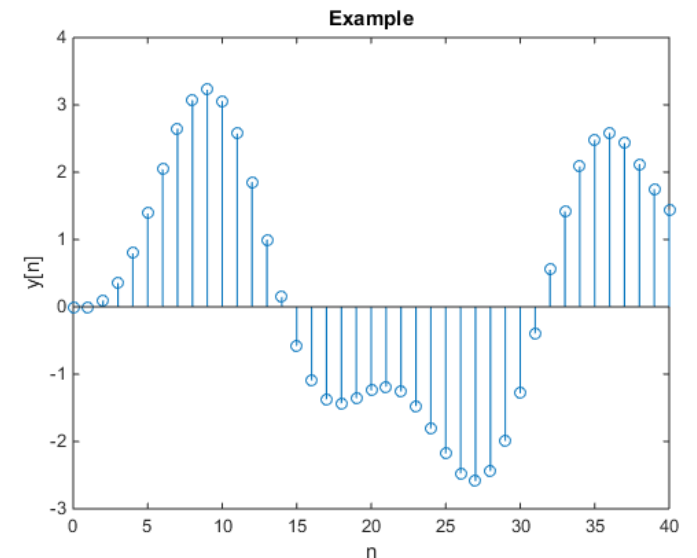
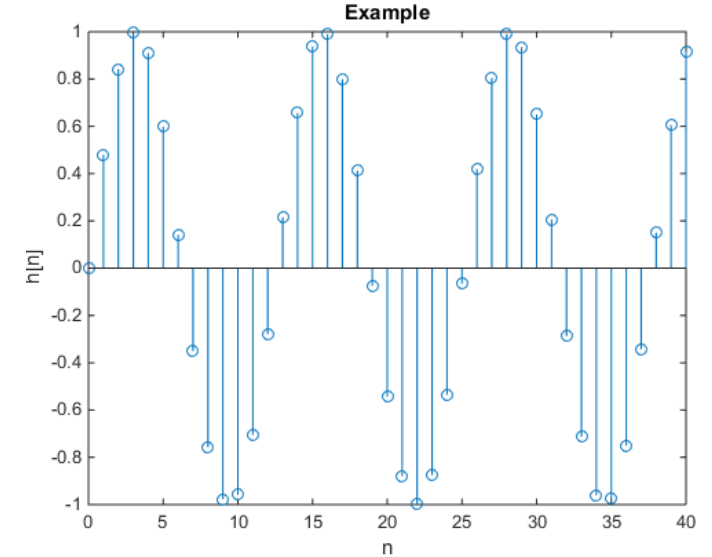
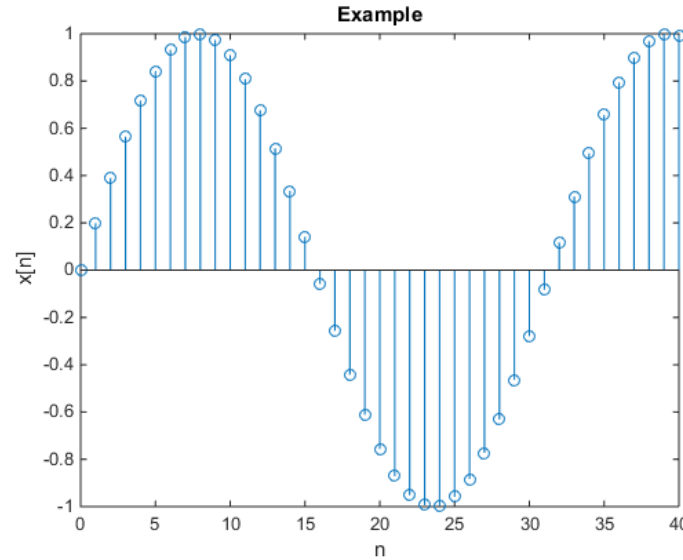
`x=[3 11 7 0 -1 4 2];`

`h=[2 3 0 -5 2 1];`

`y=conv(x,h)`

Ayrık Zamanda Konvolüsyon

```
n=0:40;  
x = sin(0.2*n);  
h = sin(0.5*n);  
y = conv(x,h);  
figure,stem(n,h);  
title('Example'),  
xlabel('n'),ylabel('h[n]')  
pause,figure,stem(n,x)  
title('Example'),xlabel('n'),ylabel('x[n]')  
pause,figure,stem(n,y(1:length(n)))  
title('Example'),xlabel('n'),ylabel('y[n]')
```



$x[n] * h[n] \rightarrow$
Konvolüsyon sonucu

Alıştırma

The conv Command

Use `help conv` to find out how to use the `conv` command.

Let

$$f(n) = u(n) - u(n - 4)$$

$$g(n) = n \cdot u(n) - 2(n - 4) \cdot u(n - 4) + (n - 8) \cdot u(n - 8).$$

Make stem plots of the following convolutions. Use the MATLAB `conv` command to compute the convolutions.

(a) $f(n) * f(n)$

(b) $f(n) * f(n) * f(n)$

(c) $f(n) * g(n)$

(d) $g(n) * \delta(n)$

(e) $g(n) * g(n)$

Comment on your observations: Do you see any relationship between $f(n) * f(n)$ and $g(n)$?

Compare $f(n)$ with $f(n) * f(n)$ and with $f(n) * f(n) * f(n)$. What happens as you repeatedly convolve this signal with itself?

Gürültü gidermek için basit
bir yol

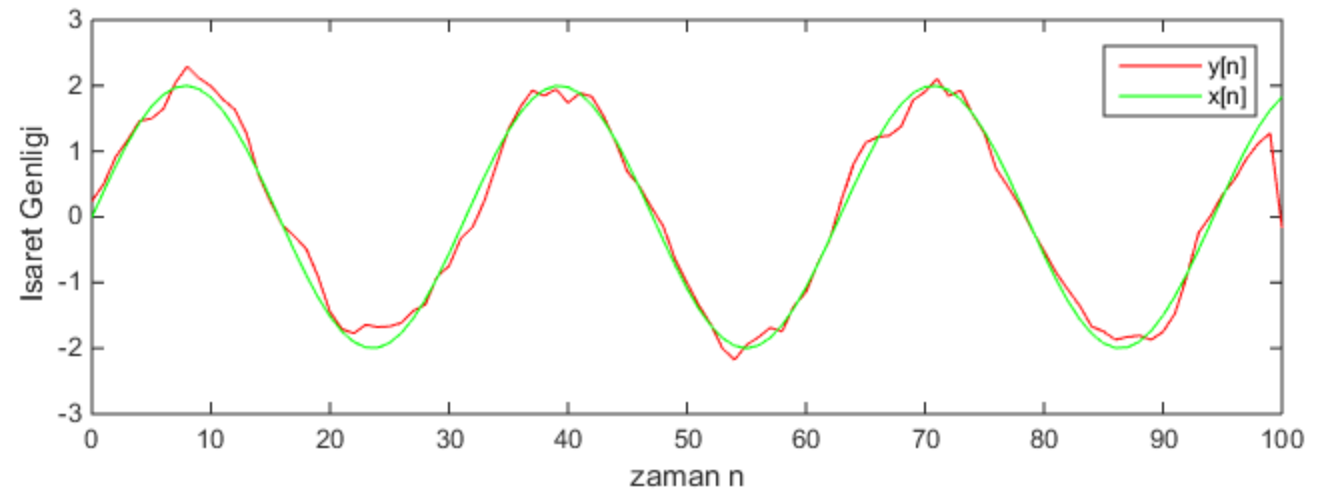
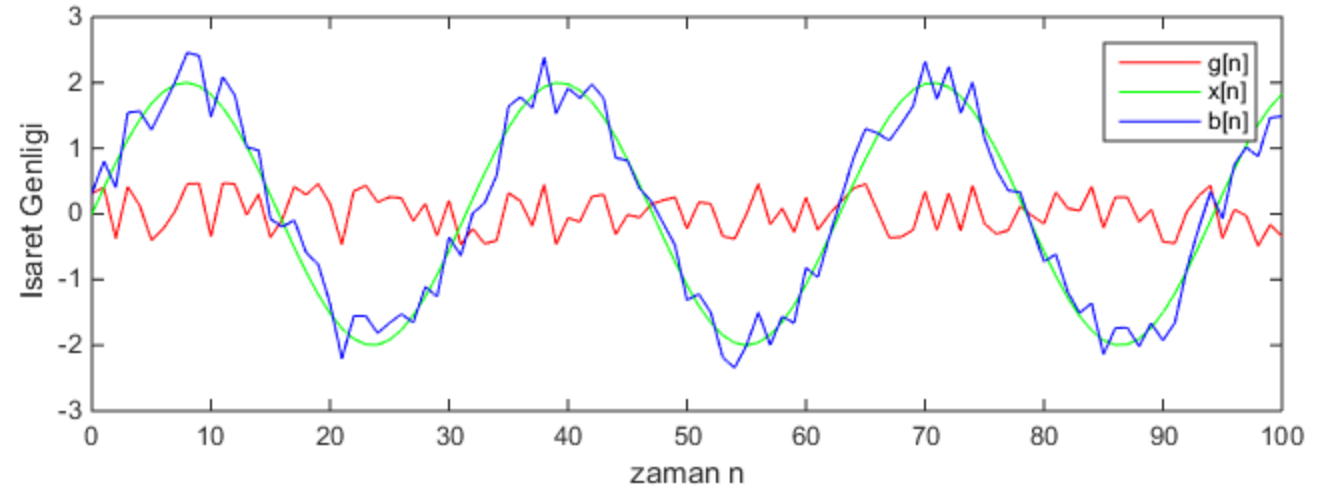
Kayan Ortalama ile Gürültü Giderme

Sayısal işaret işlemenin tipik bir uygulama alanı toplamsal gürültüyle bozulmuş bir işaretin gürültüden arındırılmasıdır. $x(n)$ gürültü eklenerek bozulan bilgi işareti, $g(n)$ bozucu gürültü işareti, $b(n) = x(n) + g(n)$ ise bozulmuş işaretini belirtsin. Amacımız gürültü bileşenini mümkün olduğunca yok edip, $x(n)$ için bir yaklaşık kestirim olan $y(n)$ işaretini oluşturmaktadır. Bu amaçla kullanılacak basit bir yöntem, her n anı için yakın $b(m)$ değerlerinin bir ortalamasını alıp gürültü bileşenini yok etmeye çalışmaktır. Buna örnek olacak üç-noktalı bir kayan-ortalama algoritması şu şekilde verilir.

$$y(n) = \frac{1}{3} \left(b(n-1) + b(n) + b(n+1) \right)$$

Kayan Ortalama ile Gürültü Giderme

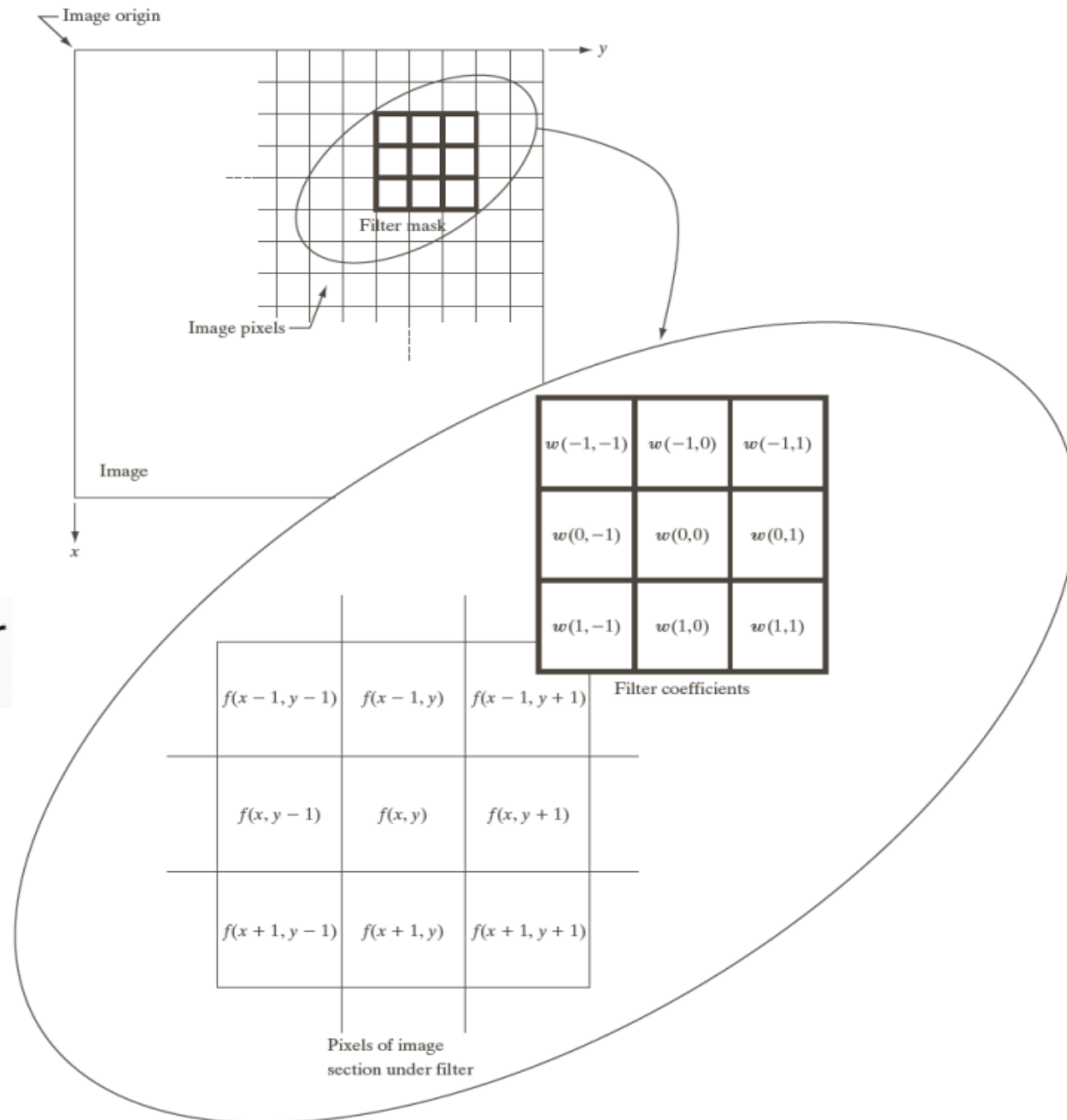
$$y(n) = \frac{1}{3}(b(n-1) + b(n) + b(n+1))$$



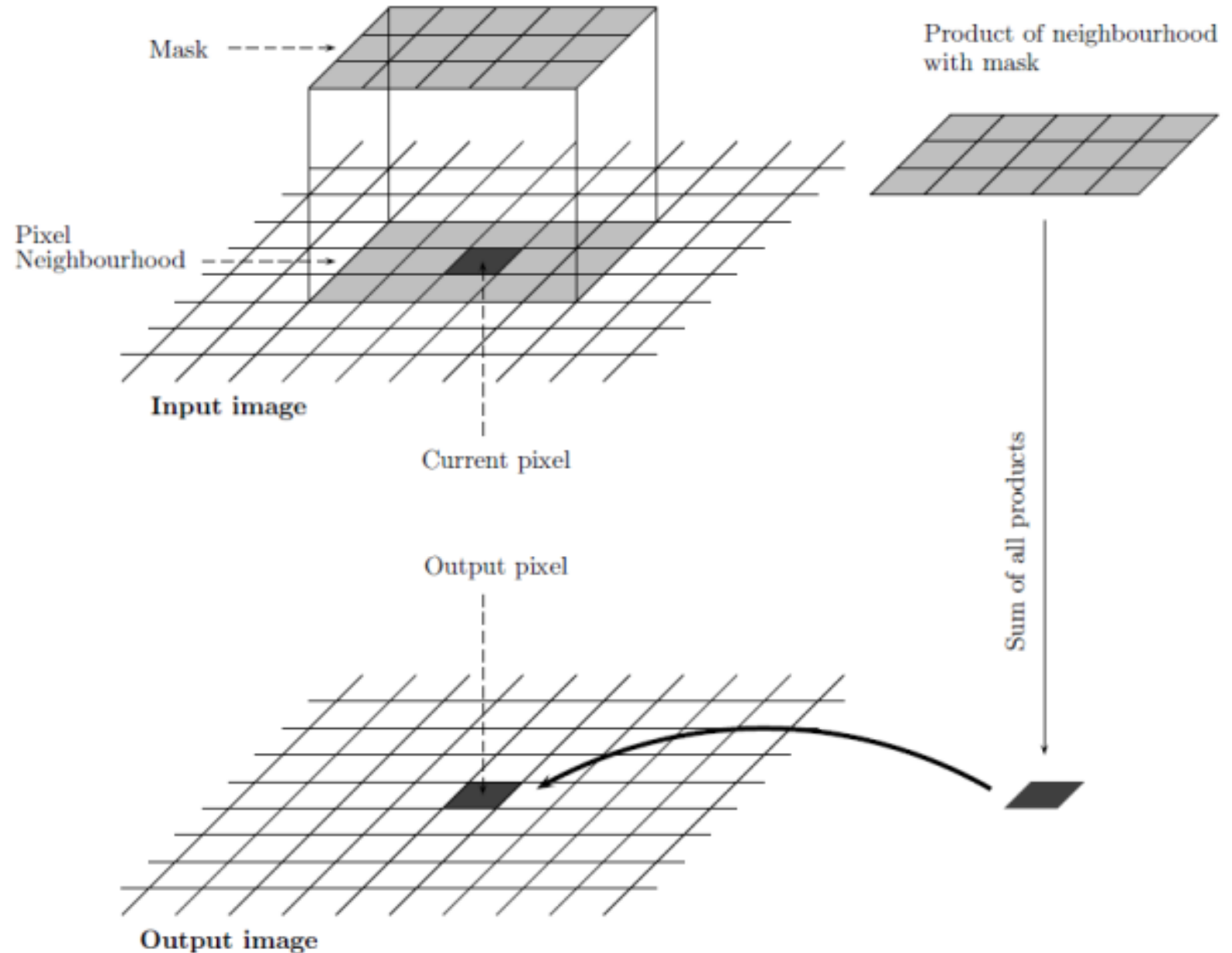
Çok boyutlu sinyaller için konvolüsyon

$$g(x, y) = \sum_{i=-a}^a \sum_{j=-b}^b w(i, j) f(x + i, y + j)$$

Linear spatial filtering of an image of size MxN with a filter of size mxn is given by the expression



Çok boyutlu sinyaller için konvolüsyon

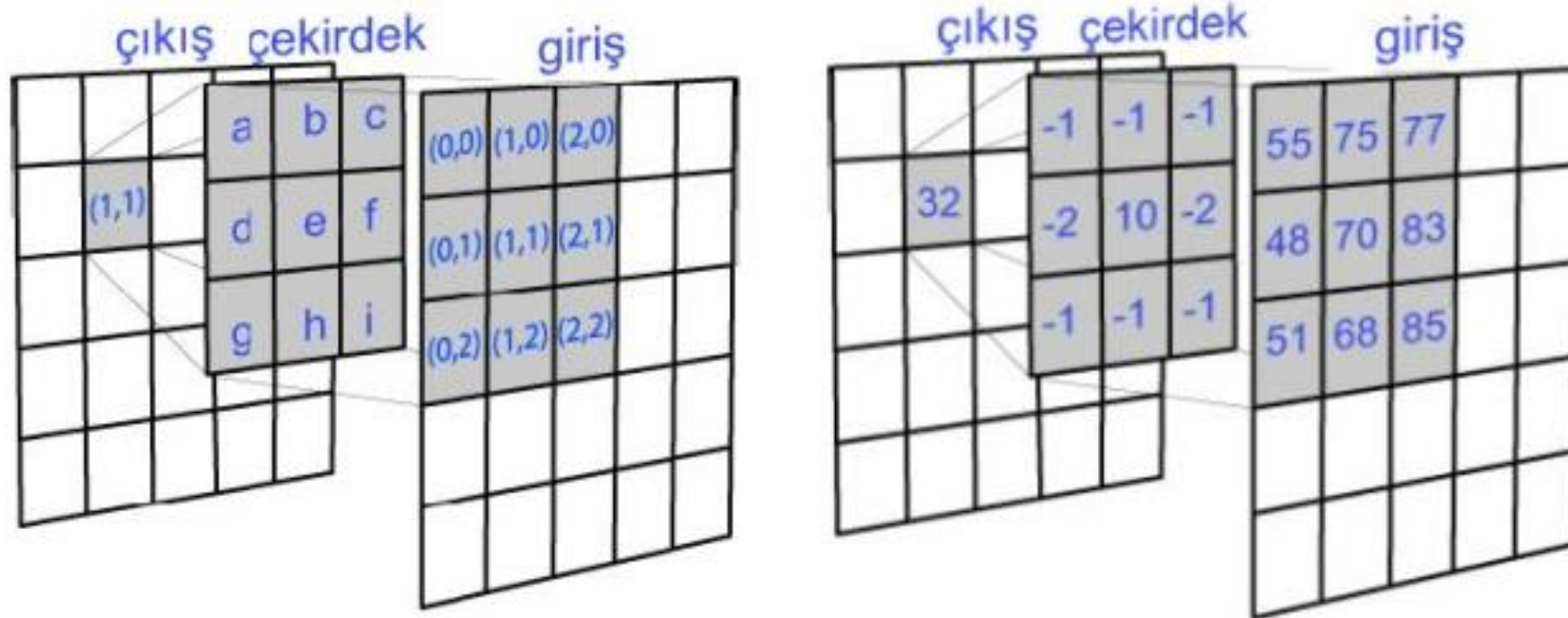


- 2 boyutlu görüntülerde **konvolüsyon yaklaşımına göre;**
 - (i) Süzgeç kalıbının merkezi, ilgilenilen görüntü pikseli üzerine gelecek şekilde görüntü üzerine yerleştirilir,
 - (ii) Kalıp içerisinde kalan piksellere ilişkin gri seviye değerleri ile aynı konumdaki süzgeç kalıp katsayıları birbiriyle çarpılarak elde edilen sonuçlar toplanır,
 - (iii) Toplam sonucu ilgilen pikselin **yeni gri seviye değeri** olarak belirlenir.
- Konvolüsyon işlemi, süzgeç kalıbının görüntünün bütünü üzerinde kaydırılmasıyla tüm pikseller için gerçekleştirilir.
- Konvolüsyon; **yumuşatma, keskinleştirme, kenar belirleme** gibi görüntü işleme fonksiyonlarını gerçekleştirmede çok sık kullanılmaktadır.
- Problemin durumuna göre gerçekleştirilecek süzgeçleme işleminde (alçak geçiren, yüksek geçiren, bant geçiren süzgeçleme) farklı katsayılarla sahip süzgeç kalıpları kullanılır.

Konvolüsyon örneği

$$image = \begin{bmatrix} i(0,0) & i(0,1) & i(0,2) \\ i(1,0) & i(1,1) & i(1,2) \\ i(2,0) & i(2,1) & i(2,2) \end{bmatrix} = \begin{bmatrix} 55 & 75 & 77 \\ 48 & 70 & 83 \\ 51 & 68 & 85 \end{bmatrix} \quad çekirdek = \begin{bmatrix} -1 & -1 & -1 \\ -2 & 10 & -2 \\ -1 & -1 & -1 \end{bmatrix}$$

$$i(1,1) = (-1 \times 55) + (-1 \times 75) + (-1 \times 77) + (-2 \times 48) + (10 \times 70) + (-2 \times 83) + (-1 \times 51) + (-1 \times 68) + (-1 \times 85) = 32 \quad (3.4)$$

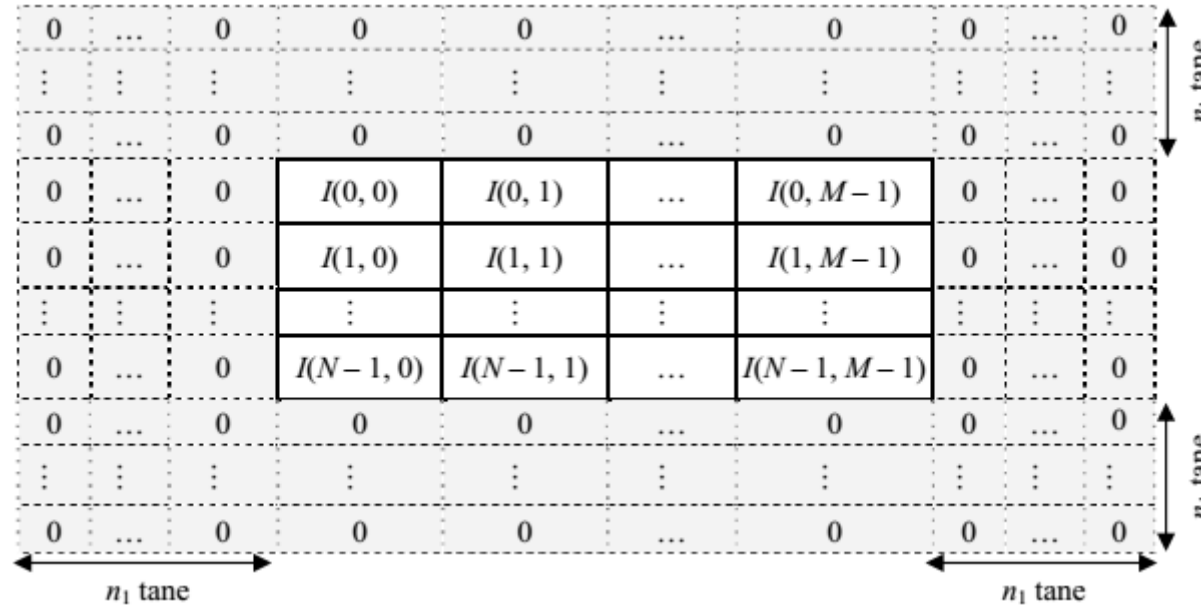


- Uzay bölgesinde süzgeçleme işlemi için yukarıda ifade edilen temel yaklaşım matematiksel olarak

$$O(i, j) = \sum_{k=-n_1}^{n_1} \sum_{l=-n_2}^{n_2} h(k, l) I(i + k, j + l)$$

- Süzgeç kalıbı için genelde $n_1 = n_2$ seçilir.

Konvolüsyon işleminde dikkat edilmesi gereken nokta; giriş görüntüsünün kenar pikselleri üzerinde konvolüsyon işlemi gerçekleştirilirken **herhangi bir hatanın oluşmaması için süzgeç kalıbının boyutuna göre giriş görüntüsünün kenarlarına sıfır eklenmesi ile ilgilidir.**



Görüntü işleme uygulamalarındaki konvolüsyon örnekleri

Alçak Geçiren Süzgeçleme

- Birbirine komşu pikseller arasındaki yüksek korelasyondan dolayı, tipik bir görüntünün enerjisinin büyük bir bölümü (yaklaşık %95'i) esas olarak alçak frekanslı bileşenlerinde toplanmıştır.
- Diğer taraftan, gürültü çok geniş bir frekans bandına sahiptir. Başka bir deyişle, gürültünün yüksek frekanslı bileşenleri daha fazladır.
- Toplamsal bir rasgele gürültü ile bozulmaya uğramış görüntünün enerjisi, gürültüden dolayı geniş bir frekans alanına yayılmış olacaktır.
- Görüntü içerisindeki alçak frekanslı bileşenler korunurken yüksek frekanslı bileşenler zayıflatılırsa (bastırılırsa), görüntüdeki gürültünün etkisi büyük oranda azaltılmış olur.
- Alçak geçiren süzgeçleme işlemi, çok az da olsa bir bilgi kaybına sebep olma pahasına görüntüdeki gürültünün büyük bir kısmını bastırır.

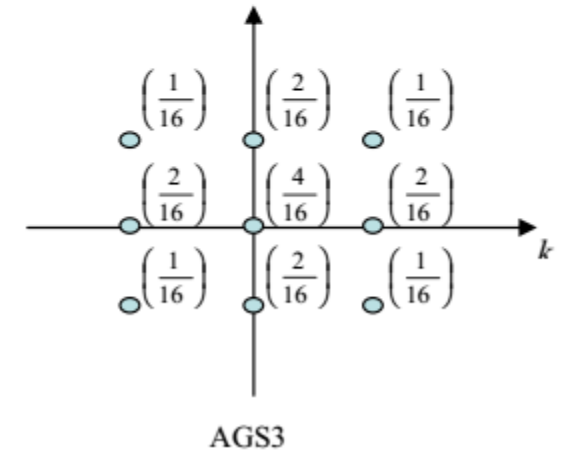
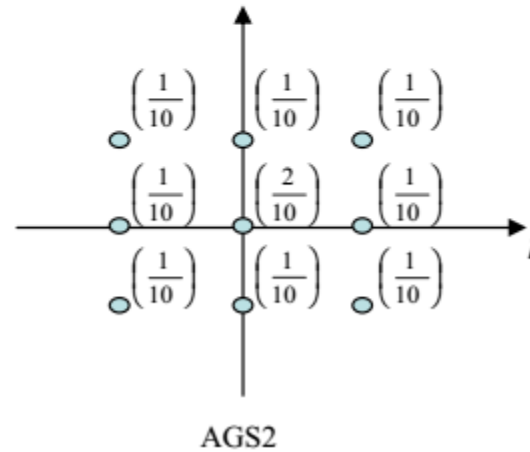
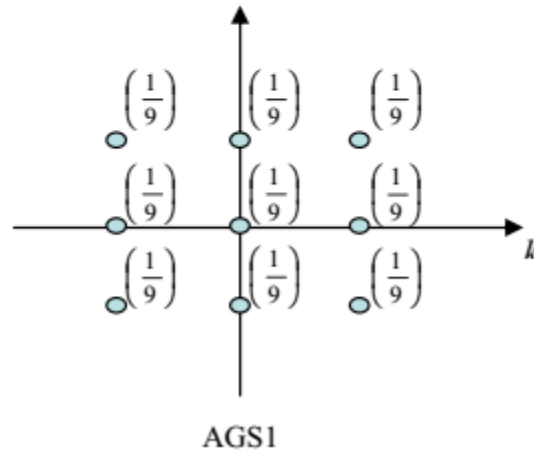
- Süzgeçleme sonucu bilgi kaybından kastedilen; görüntünün bulanıklaşması, ayrıntıların bastırılması ve keskinliğin yumuşatılması olarak ifade edilir.
- Görüntüyü oluşturan nesnelere ilişkin kenar çizgileri ve ayrıntılar yüksek frekanslı bileşenler içerdiğinden dolayı böyle bir durumun oluşması normaldir.
- Bu işlem, gürültülü görüntünün AGS kalıpları ile konvolüsyonu sonucu gerçekleştirilir.

$$\left(\frac{1}{9}\right) \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\left(\frac{1}{10}\right) \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\left(\frac{1}{16}\right) \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

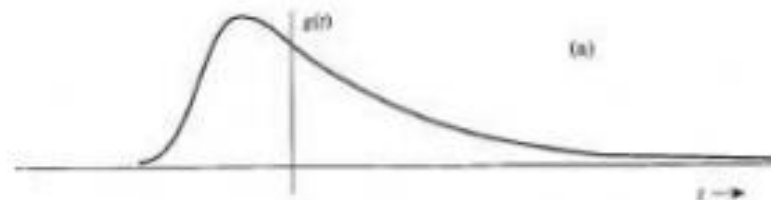
**Hangi
filtrenin
yumuşatma
gücü daha
fazladır?**



SNR=?

- Energy signals have finite energy
 - Every signal in real life is an energy signal

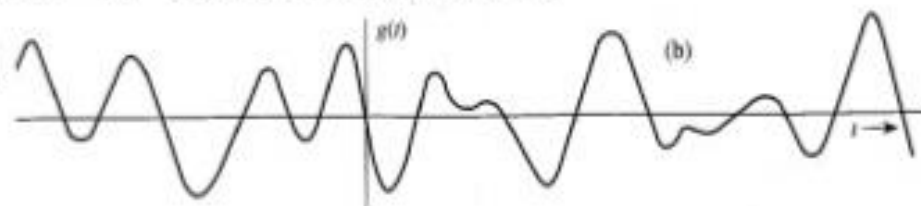
$$E_g = \int_{-\infty}^{\infty} |g(t)|^2 dt$$



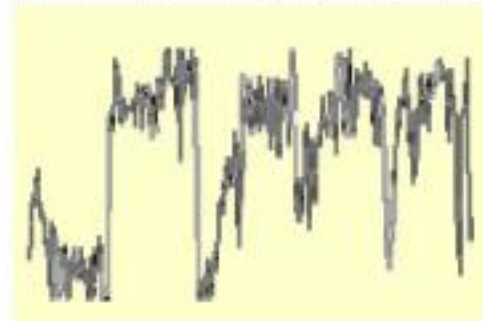
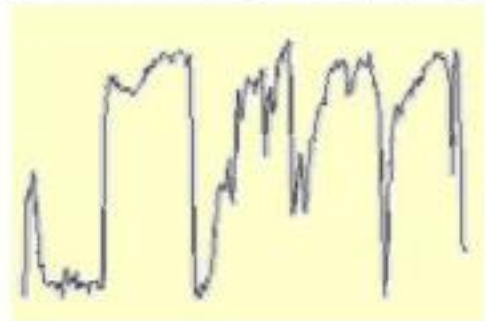
- Power signal have finite and nonzero power.

- Power signal is of infinite duration

$$P_g = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{T/2} |g(t)|^2 dt$$



Gaussian noise



$$f(x, y) = \underbrace{\hat{f}(x, y)}_{\text{Ideal Image}} + \underbrace{\eta(x, y)}_{\text{Noise process}}$$

Gaussian i.i.d. ("white") noise:
 $\eta(x, y) \sim \mathcal{N}(\mu, \sigma)$

```
>> noise = randn(size(im)).*sigma;  
>> output = im + noise;
```

What is the impact of the sigma?

Slide credit: M. Hebert

Toplamal Gauss gürültüsü ile bozulmuş görüntü, SNR = 10dB



AGS1 ile elde edilen görüntü



AGS2 ile elde edilen görüntü

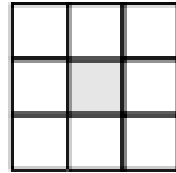


AGS3 ile elde edilen görüntü

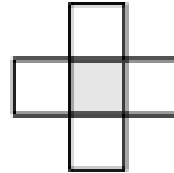


Medyan Süzgeçleme

- Gürültü ile bozulmuş görüntüdeki gürültüyü bastırmak için değişik bir yaklaşım *medyan süzgeç* kullanmaktır.
- Alçak geçiren süzgeçlemeden farklı olarak medyan süzgeçlemede; görüntüdeki her bir pikselin sahip olduğu gri seviye değeri, bu pikselin komşuluğundaki piksellerin gri seviye değerlerinin ortalaması yerine **ortancası** ile yer değiştirmektedir.
- Medyan süzgeçleme, görüntünün kenar bilgisine fazla zarar vermeden özellikle **tuz-biber gürültüsü** (salt and pepper noise) ve **benek gürültüsü** (impulsive noise or speckle noise)'nün bastırılmasında etkilidir.
- Diğer taraftan, rasgele Gauss gürültüsü için gürültü süzme başarımları kötü olup görüntüyü bulanıklaştırır ve kenar özelliklerinin kaybolmasına neden olur.



MS1



MS2



MS3

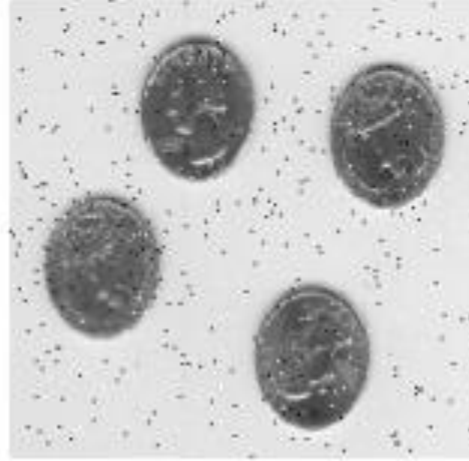


MS4

Asıl görüntü



Tuz-biber gürültülü görüntü



MS1 ile süzgeçlenmiş görüntü



MS2 ile süzgeçlenmiş görüntü



MS3 ile süzgeçlenmiş görüntü



MS4 ile süzgeçlenmiş görüntü



Asıl görüntü piksellerinin yaklaşık %2'sini etkileyecek şekilde tuz-biber gürültüsü ile bozulmuş bir görüntünün MS1, MS2, MS3 ve MS4 süzgeç kalıpları ile medyan süzgeçlenmesi sonucu elde edilen sonuçlar.

Keskinleştirme Süzgeçleri

- **Keskinleştirme** işleminin esas amacı, görüntüdeki ayrıntıları kuvvetlendirmek veya görüntünün elde edilmesi esnasında kullanılan yöntemin ya doğal etkisi ya da yapılan bir hata sonucu görüntü içerisinde **bulanıklaşan ve sonuçta kaybolan ayrıntıları iyileştirmektir.**
- Görüntü keskinleştirme; elektronik baskı ve medikal görüntülemeye endüstriyel muayene ve güçlü silahların otomatik hedef tayinine kadar geniş bir yelpazede kullanıma sahiptir.
- Komşuluktaki piksellerin ortalamasını alarak uzaysal bölgede görüntü bulanıklaştırma (yumuşatma) işlemi gerçekleştirilebilir.
- Ortalama alma işlemi integral almaya eşdeğer olduğundan dolayı keskinleştirme işleminin **türev alma** ile gerçekleştirilebileceğini düşünmek mantıksız olmaz.
- Temel olarak bir türev işlemcisinin etkisi, uygulandığı noktadaki görüntünün süreksizlik derecesi ile orantılıdır.
- Görüntünün türevini alma, kenarlar ve gürültü gibi diğer süreksizlikleri kuvvetlendirirken yavaş değişim gösteren gri seviye değerlerine sahip bölgeleri bastırır.

- x ve y yönündeki türev ifadelerini birleştirirsek **$f(x, y)$ görüntü fonksiyonu için Laplasyen (a)**

$$\nabla^2 f = [f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1)] - 4f(x, y)$$

- Bu maske 90°'lik artırımlarla gerçekleştirilen döndürmelerde yönbağımsız sonuçlar üretir. Yukarıdaki eşitliğe terimler ekleyerek köşegen yönleri Laplasyen'e dahil edilebilir **(b)**, **(c)** ve **(d)** ile verilen diğer maske biçimleri de uygulamada yaygın olarak kullanılır ki Laplasyenin negatifini alarak oluşturulabilir.

0	1	0
1	-4	1
0	1	0

(a)

1	1	1
1	-8	1
1	1	1

(b)

0	-1	0
-1	4	-1
0	-1	0

(c)

-1	-1	-1
-1	8	-1
-1	-1	-1

(d)

Sayısal Laplasyen'i gerçekleştirmede kullanılan süzgeç maskeleri.



- Bu türev işlemini gerçekleştiren maskeler

M_x	<table><tr><td>-1</td></tr><tr><td>1</td></tr></table>	-1	1	<table><tr><td>-1</td></tr><tr><td>0</td></tr><tr><td>1</td></tr></table>	-1	0	1	<table><tr><td>-1</td><td>0</td></tr><tr><td>0</td><td>1</td></tr></table>	-1	0	0	1	<table><tr><td>-1</td><td>0</td><td>1</td></tr><tr><td>-1</td><td>0</td><td>1</td></tr><tr><td>-1</td><td>0</td><td>1</td></tr></table>	-1	0	1	-1	0	1	-1	0	1	<table><tr><td>-1</td><td>0</td><td>1</td></tr><tr><td>-2</td><td>0</td><td>2</td></tr><tr><td>-1</td><td>0</td><td>1</td></tr></table>	-1	0	1	-2	0	2	-1	0	1
-1																																
1																																
-1																																
0																																
1																																
-1	0																															
0	1																															
-1	0	1																														
-1	0	1																														
-1	0	1																														
-1	0	1																														
-2	0	2																														
-1	0	1																														
M_y	<table><tr><td>-1</td><td>1</td></tr></table>	-1	1	<table><tr><td>-1</td><td>0</td><td>1</td></tr></table>	-1	0	1	<table><tr><td>0</td><td>-1</td></tr><tr><td>1</td><td>0</td></tr></table>	0	-1	1	0	<table><tr><td>-1</td><td>-1</td><td>-1</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	-1	-1	-1	0	0	0	1	1	1	<table><tr><td>-1</td><td>-2</td><td>-1</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>2</td><td>1</td></tr></table>	-1	-2	-1	0	0	0	1	2	1
-1	1																															
-1	0	1																														
0	-1																															
1	0																															
-1	-1	-1																														
0	0	0																														
1	1	1																														
-1	-2	-1																														
0	0	0																														
1	2	1																														
	(b)	(c)	(d)	(e)	(f)																											

b) maske1, **c)** maske2, **d)** *Roberts* maskesi, **e)** *Prewitt* maskesi, **f)** *Sobel* maskesi.



DIP-XE

DIP-XE

DIP-XE