

Sinyaller ve Sistemler Konvolüsyon Ödevi

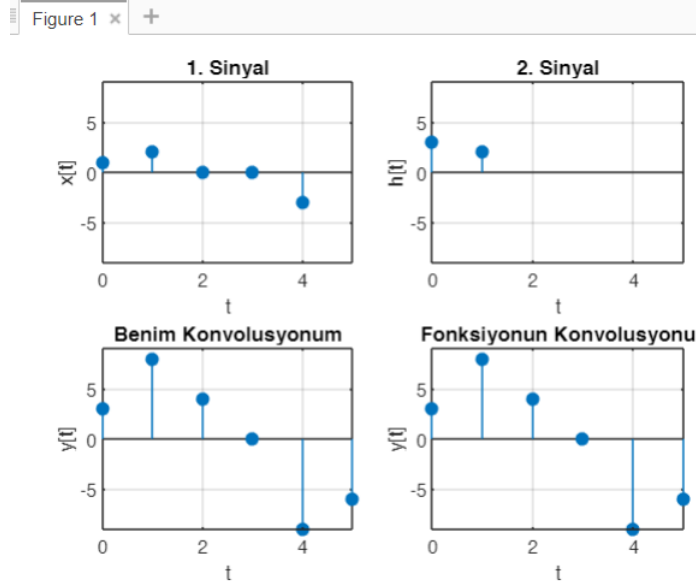
İsim: Ali Emre Nebiler | Okul No: 19011070 | Email: l1119070@std.yildiz.edu.tr

1. Deneme:

$$x[t] = \delta[n] + 2\delta[n-1] - 3\delta[n-4]$$

$$h[t] = 3\delta[n] + 2\delta[n-1]$$

x[t] baslangic noktasi: 0
x[t] = [1 2 0 0 -3]
h[t] baslangic noktasi: 0
h[t] = [3 2]



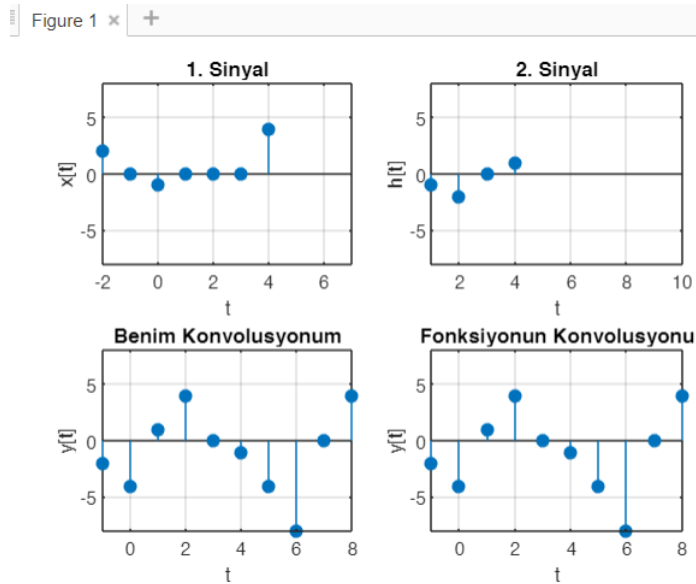
Name	Value	Size	Class
arr_all	1×19 double	1×19	double
h	[3,2]	1×2	double
h_str	0	1×1	double
i	5	1×1	double
j	2	1×1	double
l	6	1×1	double
m	5	1×1	double
max	9	1×1	double
n	2	1×1	double
x	[1,2,0,0,-3]	1×5	double
x_str	0	1×1	double
y1	[3,8,4,0,-9,-6]	1×6	double
y2	[3,8,4,0,-9,-6]	1×6	double
y_str	0	1×1	double

2. Deneme:

$$x[t] = 2\delta[n+2] - \delta[n] + 4\delta[n-4]$$

$$h[t] = -\delta[n-1] - 2\delta[n-2] + \delta[n-4]$$

x[t] baslangic noktasi: -2
x[t] = [2 0 -1 0 0 0 4]
h[t] baslangic noktasi: 1
h[t] = [-1 -2 0 1]



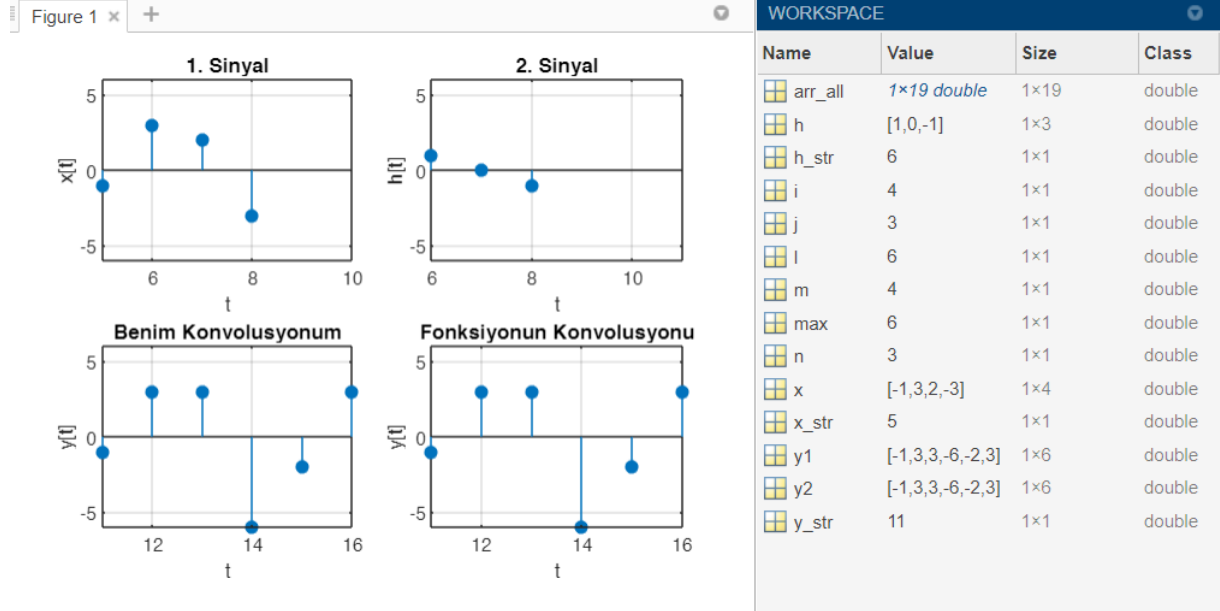
Name	Value	Size
arr_all	1×31 double	1×31
h	[-1,-2,0,1]	1×4
h_str	1	1×1
i	7	1×1
j	4	1×1
l	10	1×1
m	7	1×1
max	8	1×1
n	4	1×1
x	[2,0,-1,0,0,0,4]	1×7
x_str	-2	1×1
y1	[-2,-4,1,4,0,-1,-4,-8,0,4]	1×10
y2	[-2,-4,1,4,0,-1,-4,-8,0,4]	1×10
y_str	-1	1×1

3. Deneme:

$$x[t] = -\delta[n-5] + 3\delta[n-6] + 2\delta[n-7] - 3\delta[n-8]$$

$$h[t] = +\delta[n-6] - \delta[n-8]$$

$x[t]$ baslangic noktasi: 5
 $x[t] = [-1 \ 3 \ 2 \ -3]$
 $h[t]$ baslangic noktasi: 6
 $h[t] = [1 \ 0 \ -1]$



Yazdığım MATLAB kodu bir sonraki sayfadadır.

Algoritmada temel olarak yaptığım işlem, t anındaki $x[t]$ değeri ile tüm muhtemel $h[]$ değerlerini çarpmak ve gerekli $y[]$ 'nin üstüne eklemek. Bu şekilde tüm $y[]$ değerlerini bulabiliyorum.

```

clc
clear all
close all

disp('- KONVOLUSYON HESABI -');
disp('(!) Baslangic noktalarini tam sayi olarak giriniz. ');
disp('(!) Sinyalleri [. . .] seklinde giriniz. ');
x_str = input('x[t] için t baslangic noktasi: ');
x = input('x[t] = '); % x[t]
h_str = input('h[t] için t baslangic noktasi: ');
h = input('h[t] = '); % h[t]

m = length(x); % x[t] değer sayısı
n = length(h); % h[t] değer sayısı
l = m+n-1; % y[t] değer sayısı
y_str = x_str + h_str; % y[t] başlangıç

% benim konvolüsyon kodum
y1 = zeros(1,l); % benim y[t] sinyali
for i = 1:m % x[t]'nin değeri
    for j = 1:n % h[t]'nin iterasyonu
        %{
            her iterasyonda bir y[t] değerinin bir kısmı
            örneğin y[1]:
            x[2] h[1] iken y[1]'in bir kısmı ...
            x[1] h[2] iken y[1]'in diğer bir kısmı ...
            ve hepsinin üst üste eklenmesi.
        %}
        y1(i+j-1) = y1(i+j-1) + h(j)*x(i);
    end
end
% hazır konvolüsyon fonksiyonu
y2 = conv(x,h); % fonksiyonun y[t] sinyali

% sonuçları yazdırma
arr_all = [x h y1 y2];
max = max(abs(arr_all)); % grafik y değeri için maximum
subplot(221) % x[t] sinyali
stem(x_str:x_str+m-1, x, 'filled'),
set(gca,'xlim',[x_str x_str+l-1]), % x düzlemi aralığı
set(gca,'ylim',[-max max]), % y düzlemi aralığı
ylabel('x[t]'), xlabel('t'), grid on,
title('1. Sinyal');
subplot(222) % h[t] sinyali
stem(h_str:h_str+n-1, h, 'filled'),
set(gca,'xlim',[h_str h_str+l-1]),
set(gca,'ylim',[-max max]),
ylabel('h[t]'), xlabel('t'), grid on,
title('2. Sinyal');
subplot(223) % benim hesapladığım y[t] sinyali
stem(y_str:y_str+l-1, y1, 'filled'),
set(gca,'xlim',[y_str y_str+l-1]),
set(gca,'ylim',[-max max]),
ylabel('y[t]'), xlabel('t'), grid on,
title('Benim Konvolusyonum');
subplot(224) % fonksiyonun y[t] sinyali
stem(y_str:y_str+l-1, y2, 'filled'),
set(gca,'xlim',[y_str y_str+l-1]),
set(gca,'ylim',[-max max]),
ylabel('y[t]'), xlabel('t'), grid on,
title('Fonksiyonun Konvolusyonu');

```