İşaret İşleme
Matlab uygulamaları-H3CD3

Dr. Meriç Çetin

versiyon21920

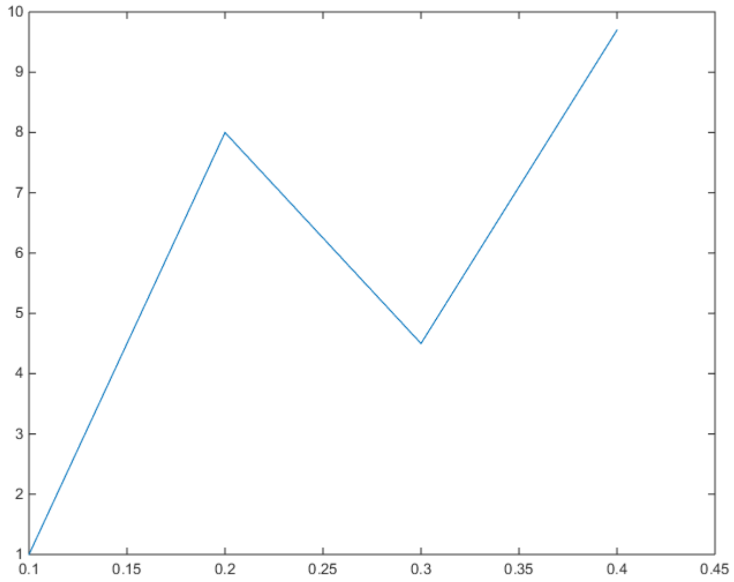
- Introduction to Signal Processing
  - https://www.youtube.com/watch?v=YmSvQe2FDKs

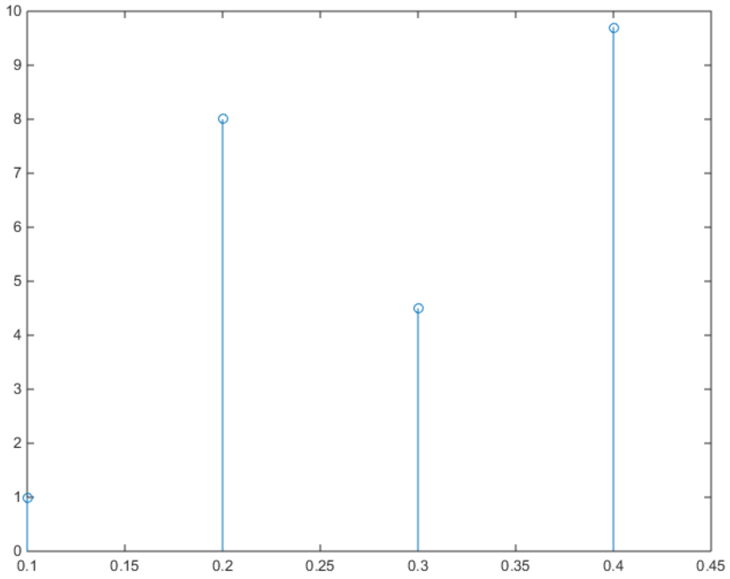
- Signal Processing with MATLAB
  - https://www.youtube.com/watch?v=sCZLJsi6-FA

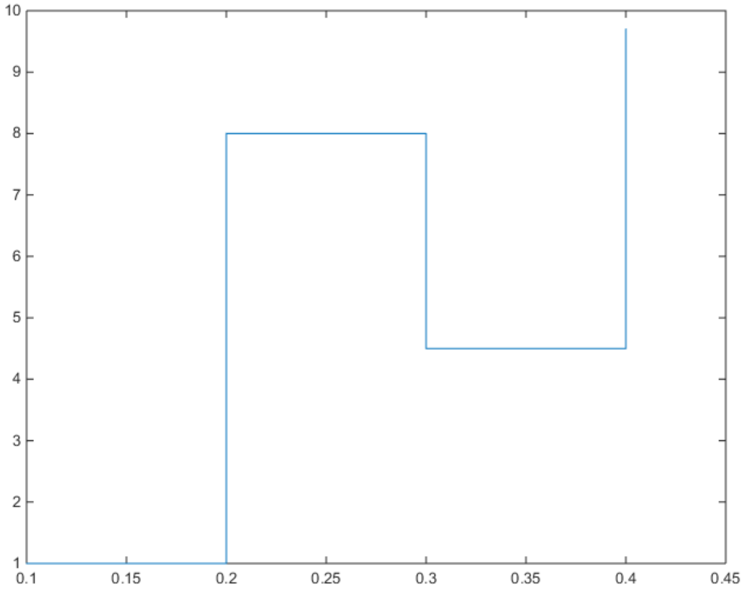
Sinyal oluşturma örnekleri

- Bu sinyalleri Matlab’da çizmeye çalışalım:
  - (x ekseni t verisi, y ekseni x verisi olmalıdır)

t	x
0.1	1
0.2	8
0.3	4.5
0.4	9.7



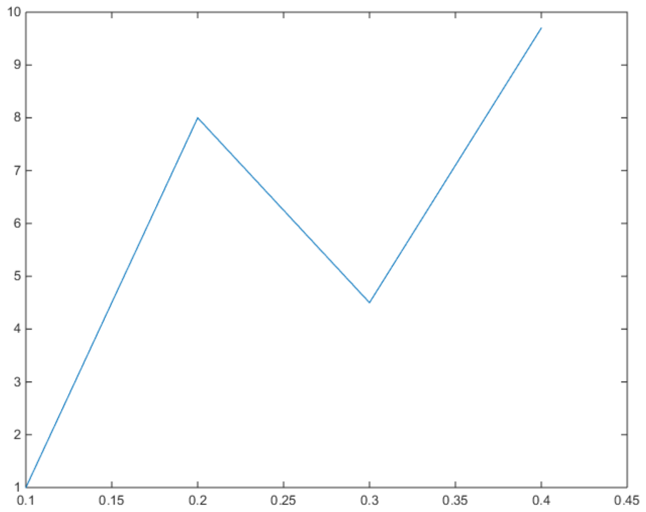


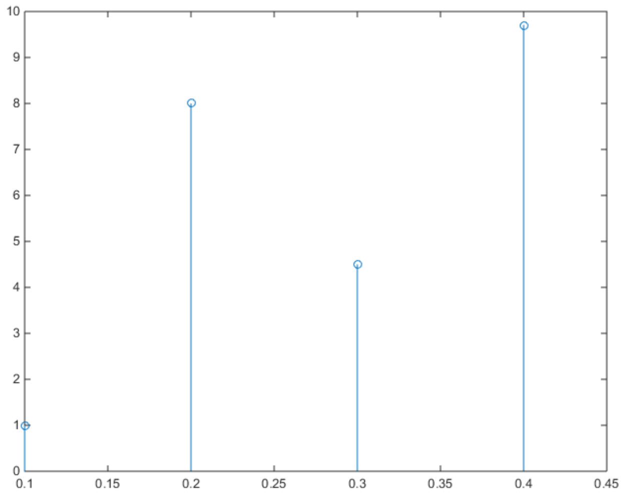


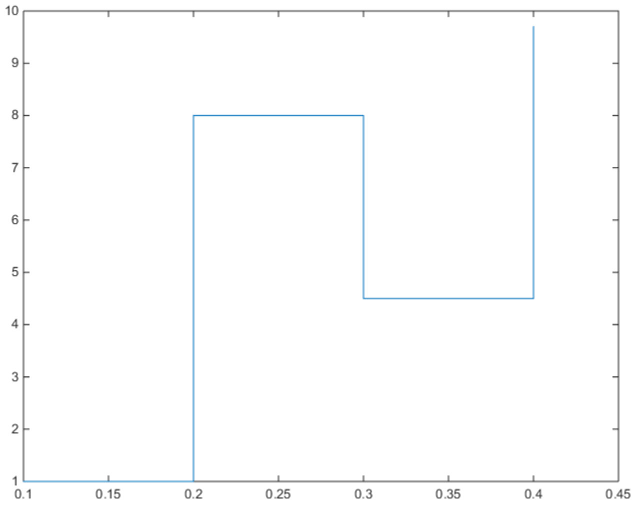
Sinyal oluşturma örnekleri

```
clc, clear all, close all
t = [0.1 0.2 0.3 0.4];
x = [1.0 8.0 4.5 9.7];
plot(t,x),
figure, stem(t,x),
figure, stairs(t,x)
```

t	x
0.1	1
0.2	8
0.3	4.5
0.4	9.7

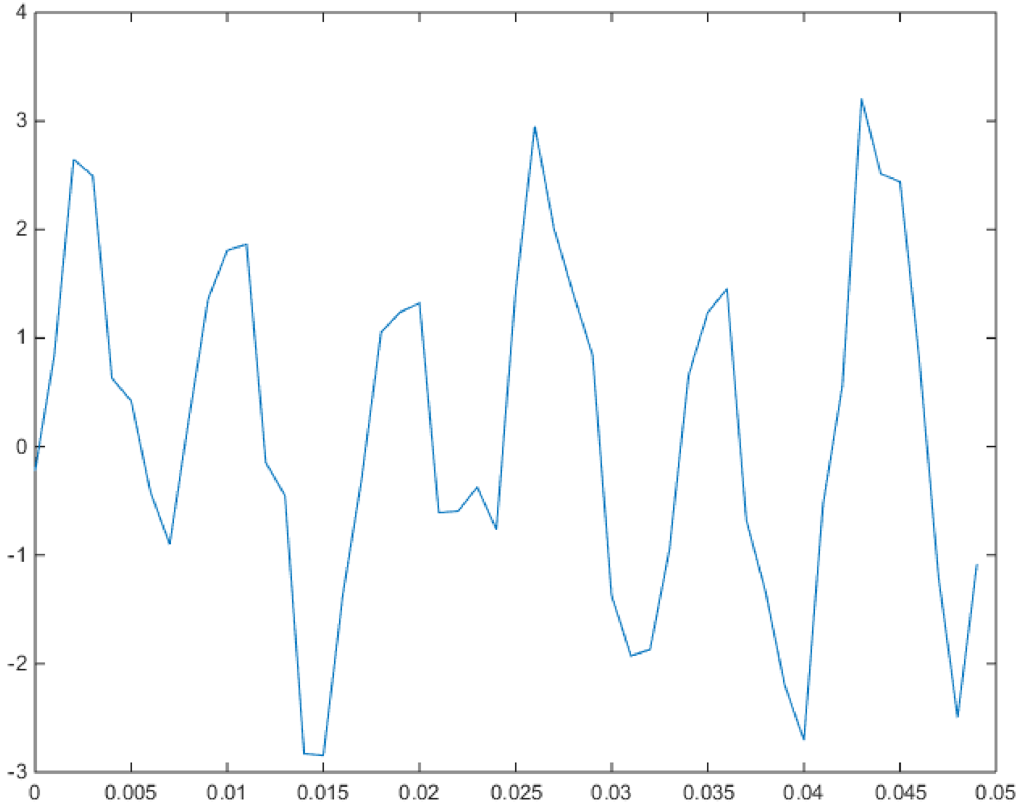






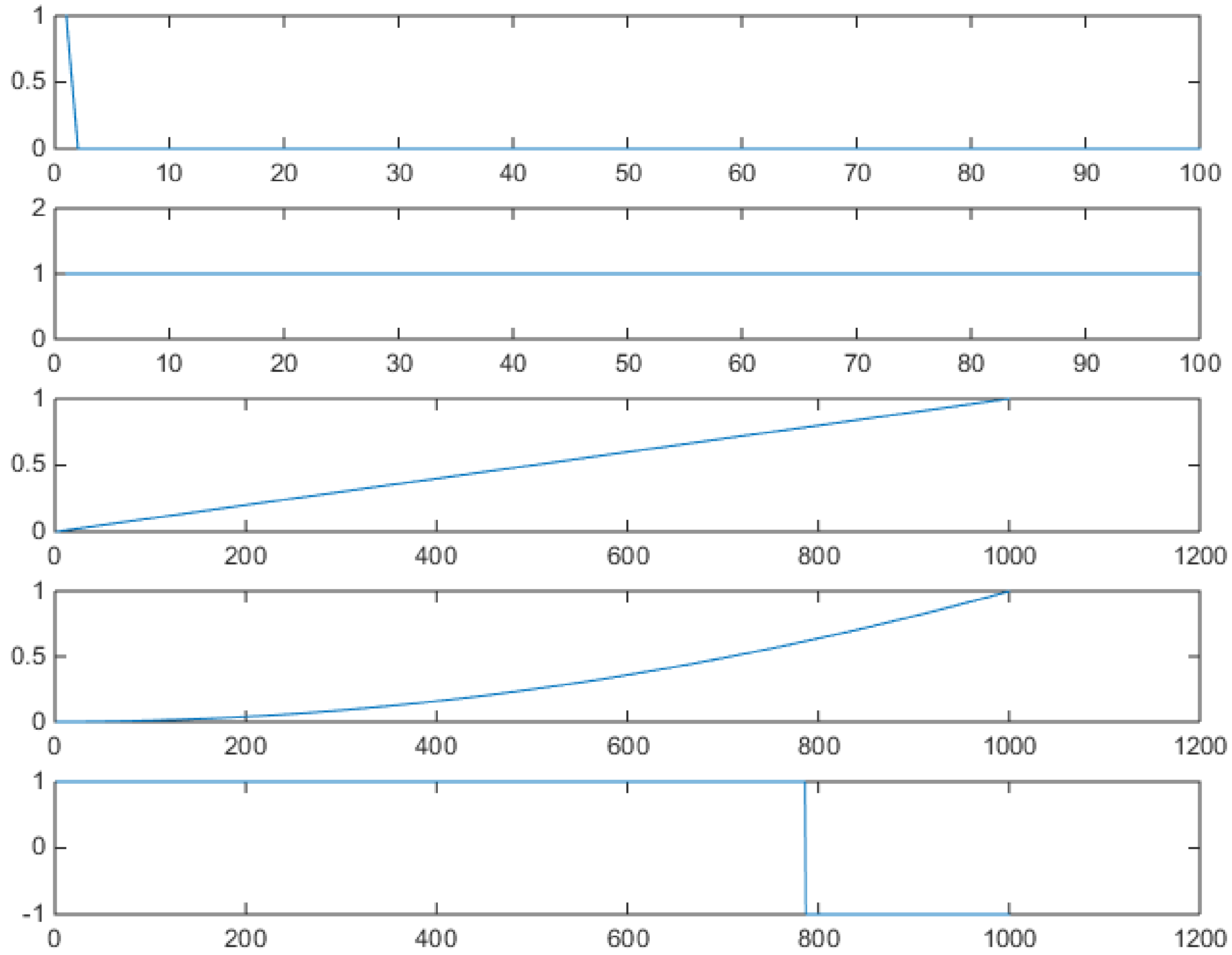
Sinyal oluşturma örnekleri

```
clear all
t = (0:0.001:1)’;
y = sin(2*pi*50*t) + 2*sin(2*pi*120*t);
randn('state',0);
yn = y + 0.5*randn(size(t));
plot(t(1:50),yn(1:50))
```



Sinyal oluşturma örnekleri

```
clear all
t = (0:0.001:1)';
y1 = [1; zeros(99,1)]; % impulse
y2 = ones(100,1); % step (filter assumes 0 initial
y3 = t; % ramp
y4 = t.^2;
y5 = square(4*t);
figure, subplot(511),plot(y1)
subplot(512),plot(y2)
subplot(513),plot(y3)
subplot(514),plot(y4)
subplot(515),plot(y5)
```



Periyodik ve aperiodik dalga formları

```
fs = 10000;
t = 0:1/fs:1.5;
x = sawtooth(2*pi*50*t);
plot(t,x), axis([0 0.2 -1 1])
```

```
t = 0:1/1000:2;
y = chirp(t,0,1,150);
specgram(y,256,1000,256,250)
```

```
x = linspace(-5,5);
y = sinc(x);
plot(x,y)
```

Bir sinyalin sesini duyabilir miyiz?

Aç-Kapat

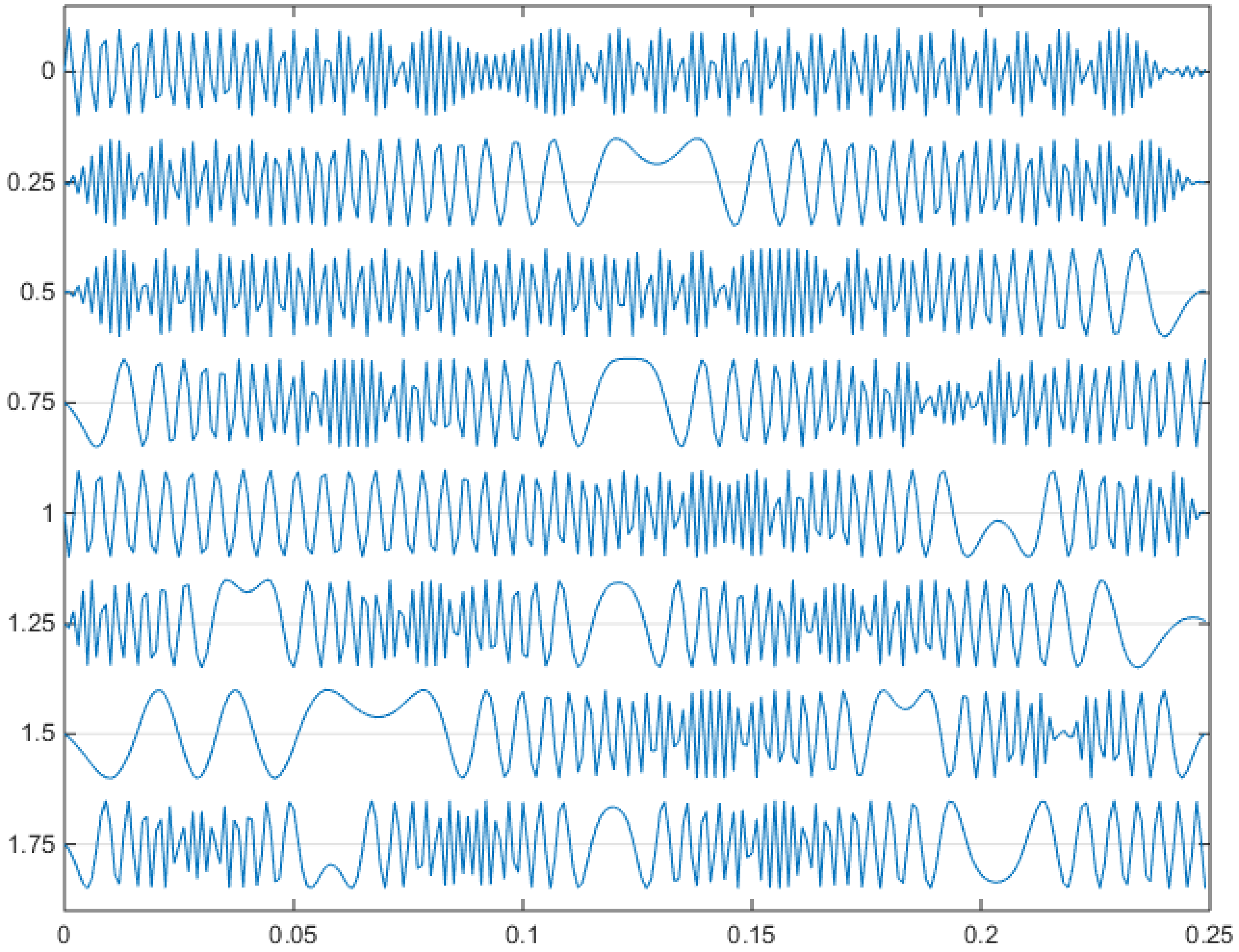
Frekans değişimini duyabildiğimiz bir sinyal örneği

```
clc,clear all
fs=1000;
ts=0:1/fs:2;

f=250+240*sin(2*pi*ts);

x=0.1*sin(2*pi*f.*ts);

strips(x,0.25,fs)
sound(x,fs)
pause(2);
soundsc(x,fs)
```



Bu komutlara bir bakın

- [y, f_s] = audioread('…');

- sound();

- [y] = audioread('…', x, f_s);

- spectrogram();

- Biraz daha ayrıntı için;
- https://www.youtube.com/watch?v=SJRHv5vvlnU

Bir ses işleme örneği

```
clc, clear all,close all
f=0.8;  n=5;
a=fir1(n,f,'high');
b=fir1(n,f,'low');

[y1, fs1] = audioread('whale1.mp3'); %[y2, fs2] = audioread('pia60.wav');
% sound(y1,fs1);

o=filter(a,1,y1);
p=filter(b,1,o);
fvtool(p,1);

subplot(2,1,1),plot(y1);  subplot(2,1,2),plot(p);
```

Bir ses işleme örneği daha

```
% Loading of test signal train
clear all
load train
whos
```

```
% Listening to/plotting train Signal
sound(y,Fs)
t=0:1/Fs:(length(y)-1)/Fs;
figure; plot(t,y'); grid
ylabel('y[n]'); xlabel('n')
```

```
% Using stem to plot 200 samples of train
figure
n=100:299;
stem(n,y(100:299)); xlabel('n');ylabel('y[n]')
title('Segment of train signal')
axis([100 299 -0.5 0.5] )
```

```
% Creating a WAV file from scratch and reading it back
clear all
Fs=5000; % sampling rate
t=0:1/Fs: 5; % time parameter
y=0.1*cos(2*pi*2000*t)-0.8*cos(2*pi*2000*t.^2); % sinusoid and chirp
%% writing chirp.wav file
wavwrite(y, Fs,'chirp.wav')
%% reading chirp.wav back into MATLAB as y1 and listening to it
[y1, Fs, nbits, readinfo] = wavread('chirp.wav');
sound(y1, Fs) % sound generated
figure
plot(t(1:1000), y1(1:1000))
```

Bir ses işleme örneği daha

Bir görüntü aynı zamanda unutulmayacak bir sinyaldir.

3 boyutlu bir sinyal

orjinal goruntu

sadece R bandina iliskin goruntu

sadece G bandina iliskin goruntu

sadece B bandina iliskin goruntu

3 boyutlu bir sinyali inceleyelim

```
clear all, clc,close all
I=imread('peppers.png');

R=I(:,:,1);
G=I(:,:,2);
B=I(:,:,3);
figure,imshow(I),title('orjinal goruntu'),pause

Ig = R; figure,imshow(Ig),title('sadece R bandina iliskin goruntu'),
Ig = G; figure,imshow(Ig),title('sadece G bandina iliskin goruntu'),
Ig = B; figure,imshow(Ig),title('sadece B bandina iliskin goruntu')
```

inline() fonksiyonu ile sinüzoid ile zorlanan bir üstel sinyal tanımlayalım;

$$f(t) = e^{-t} \cos(2\pi t).$$

```
f = inline('exp(-t).*cos(2*pi*t)','t')
t = 0;
f(t)
```

inline() fonksiyonu ile sinüzoid ile zorlanan bir üstel sinyal tanımlayalım;

$(-2 \leq t \leq 2)$ **aralığında bu f(t) sinyalini çizdirelim.**

```
f(t);
t = (-2:2);
plot(t,f(t));
xlabel('t'); ylabel('f(t)'); grid;
```

Örnekleme zamanını değiştirirsek neler değişir?

```
t = (-2:0.01:2);
figure, plot (t,f(t));
xlabel('t'); ylabel('f(t)'); grid;
```

inline() fonksiyonu ile birim basamak sinyalini çizdirelim;

```
t = (-2:2)
```

```
clc,clear all
u = inline('(t>=0)','t')
t = (-2:2);
plot (t,u(t));
xlabel('t'); ylabel('u(t)');
```

Örnekleme zamanını değiştirirsek neler değişir?

```
t = (-2:0.01:2);
figure, plot (t,u(t));
xlabel('t'); ylabel('u(t)');
axis ([-2 2 -0.1 1.1]);
```

inline() fonksiyonu ile belli bir aralıkta sınırlı bir birim basamak sinyalini çizdirelim;

```
clc,clear all,close all
p = inline('(t>=0) & (t<1)','t');
t = (-1:0.01:2);
plot(t,p(t));
xlabel('t');
ylabel('p(t) = u(t)-u(t-1)');
axis ([-1 2 -.1 1.1]);
```

inline() fonksiyonu ile sinüzoid ile zorlanan bir üstel sinyalde bağımsız değişken dönüşümleri

$$g(t) = f(t)u(t) = e^{-t}\cos(2\pi t)u(t)$$

$g(2t + 1)$ fonksiyonunu $(-2 \leq t \leq 2)$ aralığı için çizdiriniz.

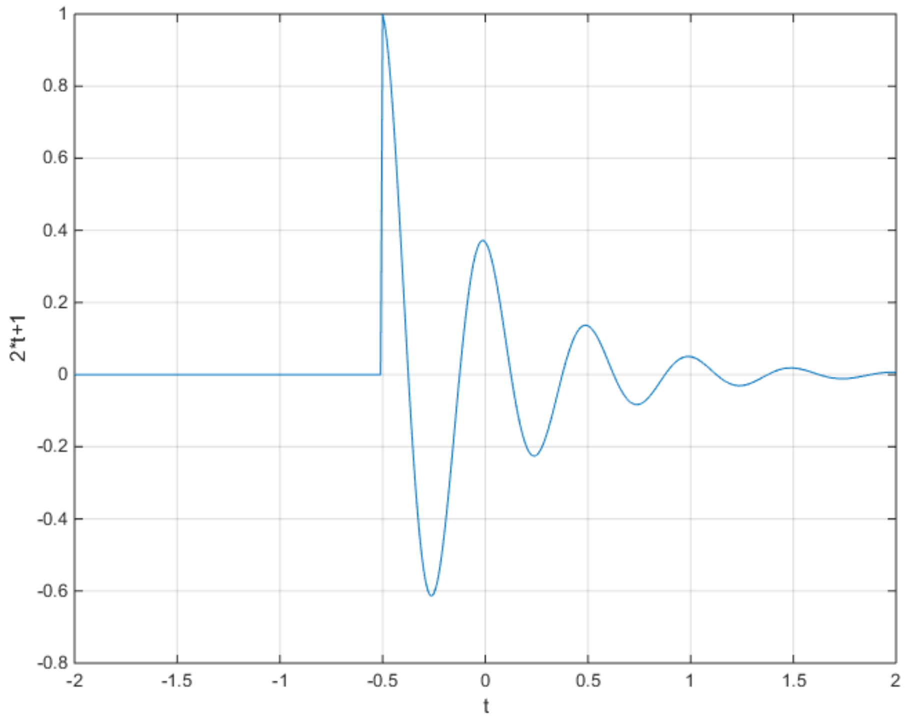
g(2t+1) ---> hangi değişken dönüşümleri kullanılır?

inline() fonksiyonu ile sinüzoid ile zorlanan bir üstel sinyalde bağımsız değişken dönüşümleri

$$g(t) = f(t)u(t) = e^{-t}\cos(2\pi t)u(t)$$

g(2t+1) ---> hangi değişken dönüşümleri kullanılır?

```
t = (-2:0.01:2);
g = inline('exp(-t).*cos(2*pi*t).*(t>=0)', 't');
figure, plot (t,g(2*t+1));
xlabel('t'); ylabel('2*t+1'); grid
```



inline() fonksiyonu ile sinüzoid ile zorlanan bir üstel sinyalde bağımsız değişken dönüşümleri

$$g(t) = f(t)u(t) = e^{-t}\cos(2\pi t)u(t)$$

g(-t+1) fonksiyonunu $(-2 \le t \le 2)$ aralığı için çizdiriniz.

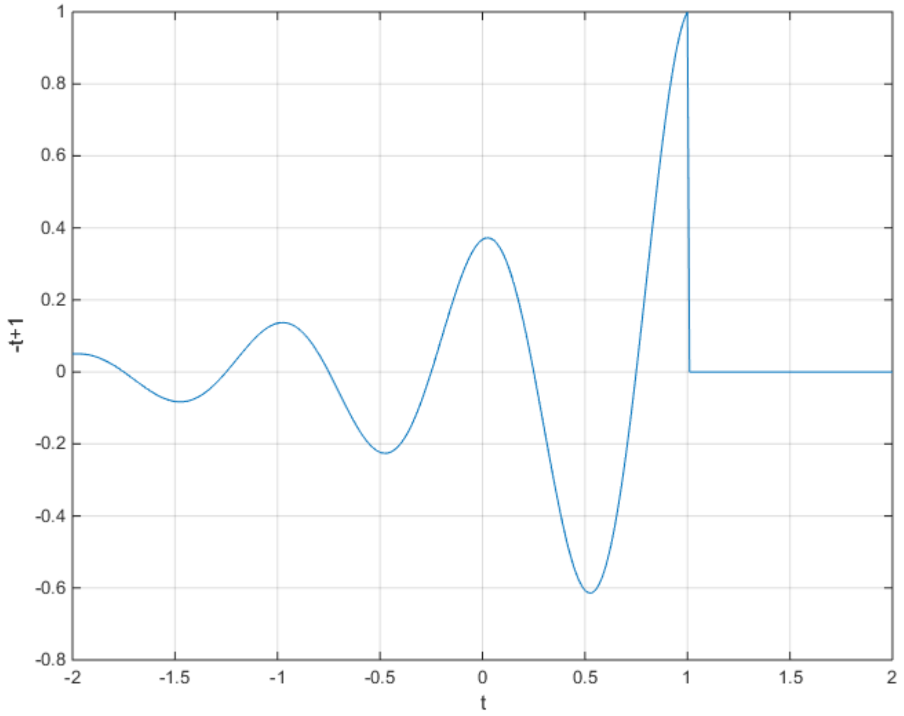
g (-t+1) ---> hangi değişken dönüşümleri kullanılır?

inline() fonksiyonu ile sinüzoid ile zorlanan bir üstel sinyalde bağımsız değişken dönüşümleri

$$g(t) = f(t)u(t) = e^{-t}\cos(2\pi t)u(t)$$

g(-t+1) ---> hangi değişken dönüşümleri kullanılır?

```
t = (-2:0.01:2);
g = inline('exp(-t).*cos(2*pi*t).*(t>=0)', 't');
figure, plot (t,g(-t+1));
xlabel('t'); ylabel('-t+1'); grid
```



Genel Karmaşık Üstel Sinyaller

Hatırlatma: H1CD3

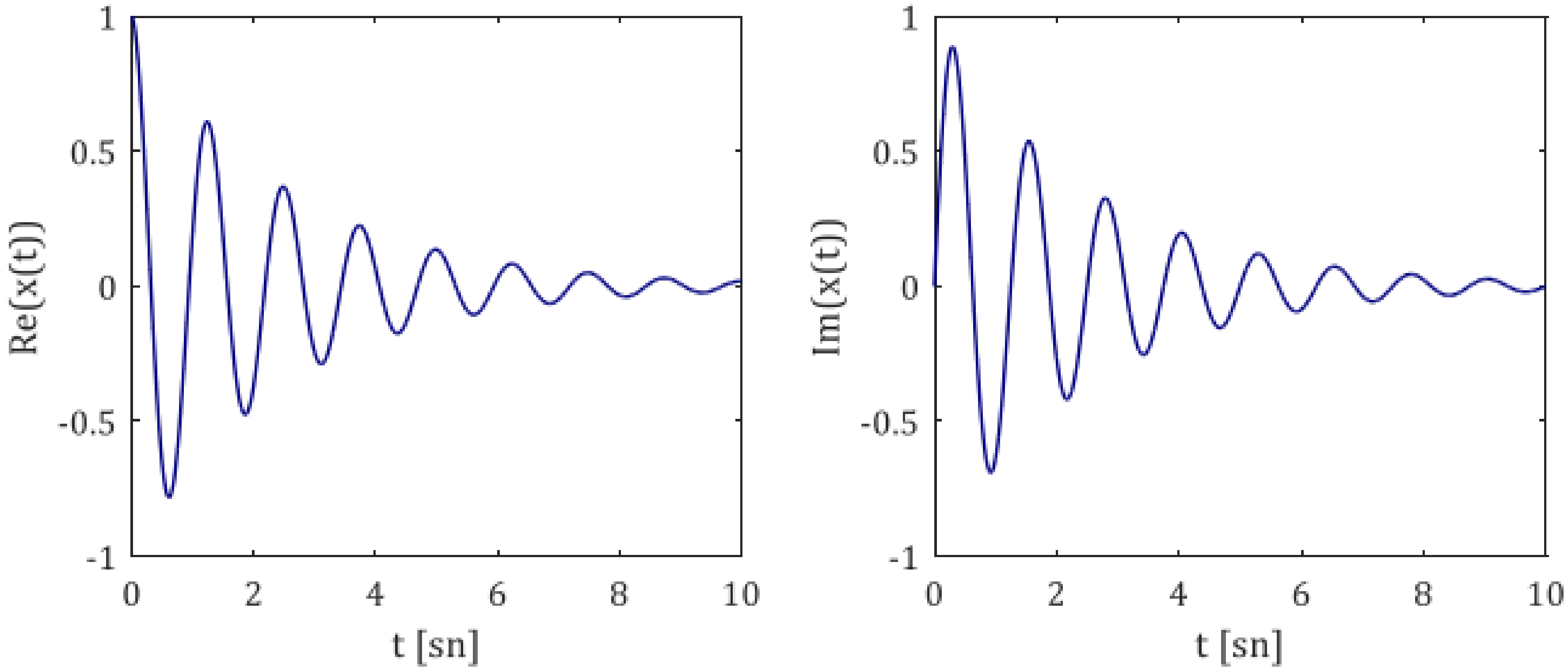
Bu sinyaller şu şekildedir: $x(t) = e^{(\sigma + j\omega_0)t}.$

Euler bağıntısı ➡ $x(t) = e^{(\sigma + j\omega_0)t} = e^{\sigma t} \cos \omega_0 t + j e^{\sigma t} \sin \omega_0 t.$

Genel Karmaşık Üstel Sinyaller

Hatırlatma: H1CD3

$x(t) = e^{(-0.4+j1.6\pi)t}$ sinyalinin reel ve sanal kısımlarının grafikleri



$$x(n) = e^{(-0.1+j0.3)n}, \quad -10 \le n \le 10$$

- Verilen işaretin reel, sanal kısımlarını çizdirelim, aynı zamanda genlik ve faz açısını gözlemleyelim:

```
n=-10:1:10; alpha= -0.1+0.3*j;
x=exp(alpha*n);
subplot(221);stem(n,real(x));title('real part');xlabel('n')
subplot(222);stem(n,imag(x));title('imaginary part');xlabel('n')
subplot(223);stem(n,abs(x));title('magnitude part');xlabel('n')
subplot(224);stem(n,(180/pi)*angle(x));title('phase part');xlabel('n')
```

Verilen bir sisteme ait sıfır giriş cevabının bulunması

SGC için genel çözüm

Therefore, equation (3.1): $(D^N + a_1 D^{N-1} + \cdots + a_{N-1} D + a_N) y_0(t) = 0$

has **N possible solutions**: $c_1 e^{\lambda_1 t}, c_2 e^{\lambda_2 t}, \ldots, c_N e^{\lambda_N t}$

where $c_1, c_2, \ldots, c_N$ are arbitrary constants.

It can be shown that the **general solution** is the sum of all these terms:

$$y_0(t) = c_1 e^{\lambda_1 t} + c_2 e^{\lambda_2 t} + \cdots + c_N e^{\lambda_N t}$$

Katsız kök olduğunda

In order to determine the N arbitrary constants, we need to have **N constraints** (i.e. initial or boundary or auxiliary conditions).

- $(D^2+4D+k)y(t) = (3D+5)x(t)$
  - sistemi için aşağıda belirtilen k değerleri ve verilen başlangıç şartları için sistemin sıfır giriş cevabını bulunuz.

Find the roots λ_1 and λ_2 of the polynomial $\lambda^2 + 4\lambda + k$ for three values of k :

a. $k = 3$

b. $k = 4$

c. $k = 40$

Using initial conditions $y_0(0) = 3$ and $\dot{y}_0(0) = -7$

İlk olarak k = 3 olduğunda sistemin kökleri ?

```
y_0 = dsolve('D2y + 4*Dy + 3*y = 0','y(0) = 3','Dy(0) = -7','t');
disp(['(a) k = 3; y_0 = ',char(y_0)])
t=[-2:0.01:2];
ysgc=exp(-t) + 2*exp(-3*t);
figure, plot(t,ysgc);
xlabel('t'); ylabel('ysgc');
```

- $(D^2+4D+k)y(t) = (3D+5)x(t)$
  - sistemi için aşağıda belirtilen k değerleri ve verilen başlangıç şartları için sistemin sıfır giriş cevabını bulunuz.

Find the roots λ_1 and λ_2 of the polynomial $\lambda^2 + 4\lambda + k$ for three values of k :

a. $k = 3$

b. $k = 4$

c. $k = 40$

Using initial conditions $y_0(0) = 3$ and $\dot{y}_0(0) = -7$

k = 4 olduğunda sistemin kökleri ?

```
y_0 = dsolve('D2y + 4*Dy + 4*y = 0','y(0) = 3','Dy(0) = -7','t');
disp(['(a) k = 4; y_0 = ',char(y_0)])
t=[-2:0.01:2];
ysgc=3*exp(-2*t) – t.*exp(-2*t);
figure, plot(t,ysgc);
xlabel('t'); ylabel('ysgc');
```

- $(D^2+4D+k)y(t) = (3D+5)x(t)$
  - sistemi için aşağıda belirtilen k değerleri ve verilen başlangıç şartları için sistemin sıfır giriş cevabını bulunuz.

Find the roots λ_1 and λ_2 of the polynomial $\lambda^2 + 4\lambda + k$ for three values of k :

a. $k = 3$

b. $k = 4$

c. $k = 40$

Using initial conditions $y_0(0) = 3$ and $\dot{y}_0(0) = -7$

k = 40 olduğunda sistemin kökleri ?

```
y_0 = dsolve('D2y + 4*Dy + 40*y = 0','y(0) = 3','Dy(0) = -7','t');
disp(['(a) k = 40; y_0 = ',char(y_0)])
t=[-2:0.01:2];
ysgc=3*cos(6*t).*exp(-2*t) - (sin(6*t).*exp(-2*t))/6;
figure, plot(t,ysgc);
xlabel('t'); ylabel('ysgc');
```