

# SAYISAL ANALİZ

**Doç.Dr. Cüneyt BAYILMIŞ**



# SAYISAL ANALİZ

## 2. Hafta

### MATLAB İLE GRAFİK ÇİZİMLERİ

# İÇİNDEKİLER

1. **plot** Komutu İle Grafik Çizimi
2. **fplot** Komutu İle Grafik Çizimi
3. **ezplot** Komutu İle Grafik Çizimi
4. **Grafikler Üzerinde Düzenlemeler**
5. **subplot** Komutu ile Figür Penceresini Bölme
6. **Özel Grafikler**

# plot komutu ile grafik çizme

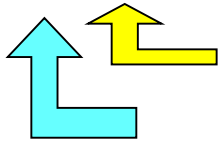
- ☐ **plot** komutunun genel kullanımı
- ☐ **xlabel** komutu ile **x** ekseninin adlandırılması
- ☐ **ylabel** komutu ile **y** ekseninin adlandırılması
- ☐ **title** komutu ile **grafiğe** isim verilmesi
- ☐ **renk**, **şekil**, **kalınlık** gibi grafiklerin özelliklerinin değiştirilmesi
- ☐ **hold on** komutu ile tek bir pencerede birden fazla grafik çizdirilmesi
- ☐ **grid** komutu ile **yatay ve dikey** bölümlendirme
- ☐ **axis** komutu ile **eksen** ölçeklendirme

# plot Komutu ile Grafik Çizimi

## genel kullanımı

❑ İki boyutlu grafik çiziminde kullanılır.

❑ `plot ( x , y )`



**y** eksenine ait vektörel ifade

**x** eksenine ait vektörel ifade

❑ **Örnek:**  $u(t) = 2\sin(\omega t)$  sinyalini **0.01** adımlarla, **0 ile 10** sn zaman dilimi için çiziniz? **Not:**  $\omega = 1$



### Komut penceresi

*% 0.01 artışlar ile 0 – 10 sn zaman diliminin tanımlanması*

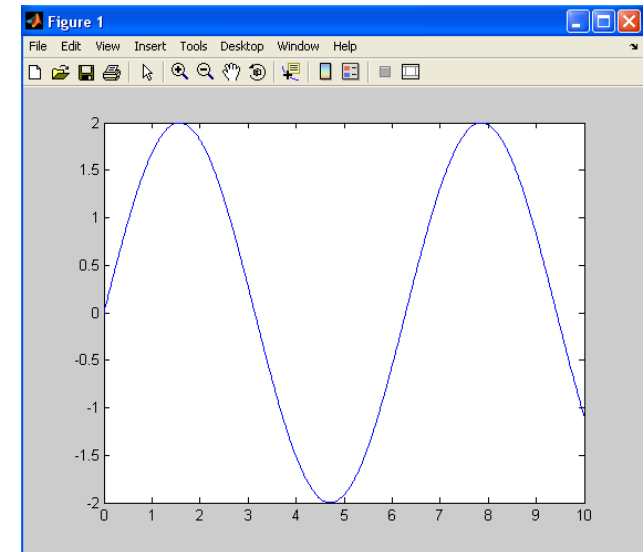
```
>> t = 0: 0.01 : 10;
```

*% Grafiğin y eksenini oluşturacak u(t) sinyalinin tanımlanması*

```
>> u = 2*sin(t);
```

*% Grafiğin çizdirilmesi*

```
>> plot(t,u)
```



# plot Komutu ile Grafik Çizimi

## grafiklere ve eksenlere isim verilmesi

- ❑ Çizdirilen grafiklerin daha anlamlı olması için, grafiklere başlık ve x ile y eksenine de isim verilmesi gerekir.
  - `title ( ' Grafiğin başlığı ' )`
  - `xlabel ( ' x ekseninin etiketi ' )`
  - `ylabel ( ' y ekseninin etiketi ' )`
- ❑ Önceki örnek çizdirilen grafik üzerinde isim verilmesi:

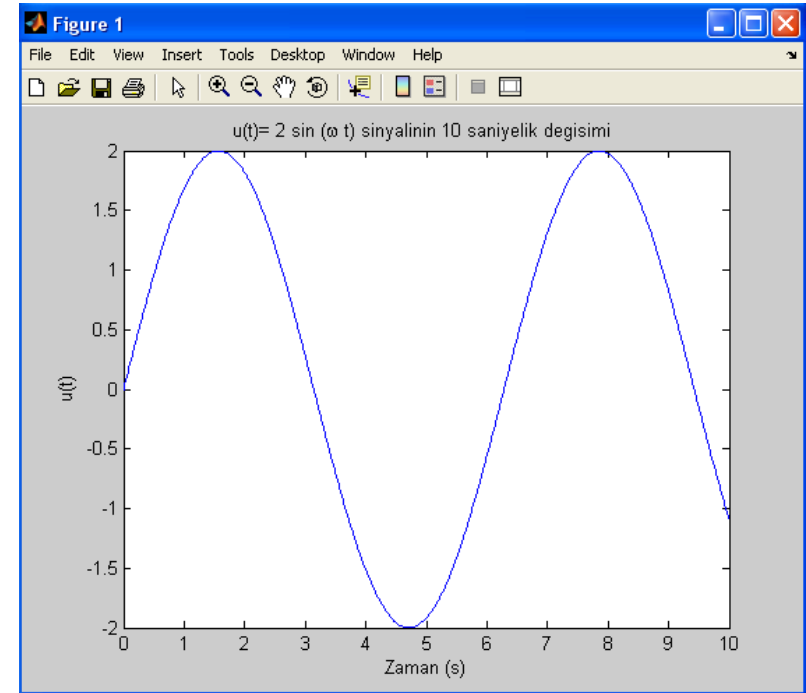


### Komut penceresi

```
% Grafik üzerinde eksen açıklamalarının yapılması
>> xlabel ( 'Zaman (s) ' )

>> ylabel ( ' u(t) ' )

% Grafiğe başlık verilmesi
>> title ( 'u(t)= 2 sin ( \omega t )
sinyalinin 10 saniyelik değişimi ' )
```



# plot Komutu ile Grafik Çizimi

## grafik başlık ve eksen isimlerinin boyutlandırılması

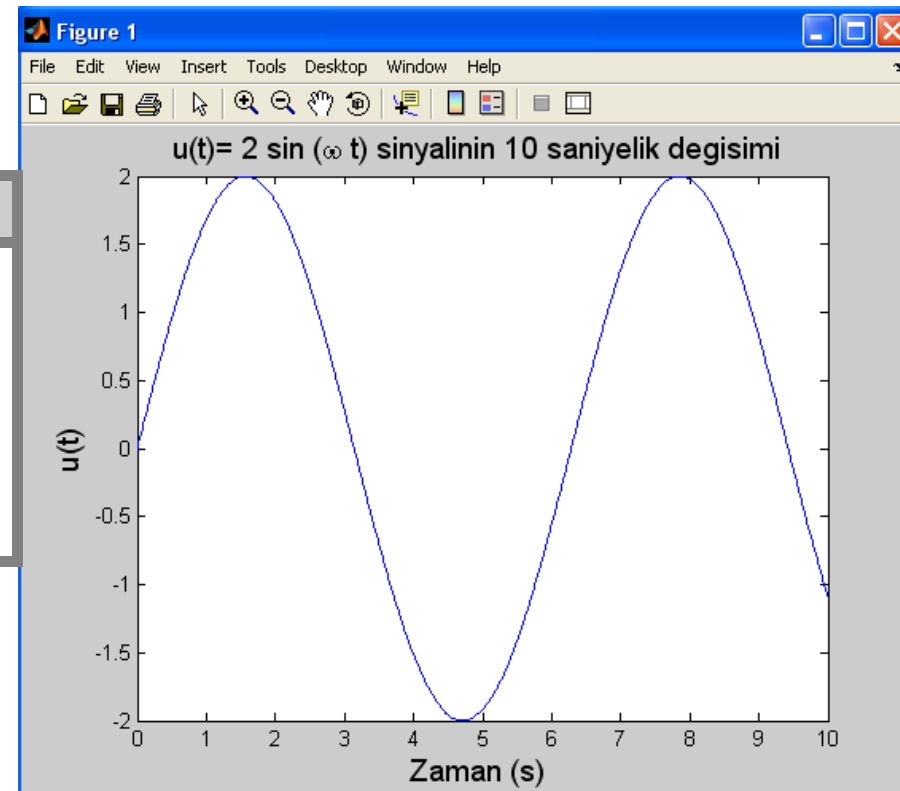
- ❑ Bazı durumlarda eksen ve başlık isimlerinin daha koyu yazdırılması istenebilir. Bu durumda yazının büyüklük (font) ayarı değiştirilmelidir.

➤ `fontsize ( ' istenen punto ' )`



### Komut penceresi

```
% Grafik üzerinde eksen ve başlık açıklamalarının 14 punto yazılması  
>> xlabel ('Zaman (s)', 'fontsize', [14])  
  
>> ylabel (' u(t) ', 'fontsize', [14])  
  
>> title ('u(t)= 2 sin (\omega t) sinyalinin 10  
saniyelik değişimi ', 'fontsize', [14])
```

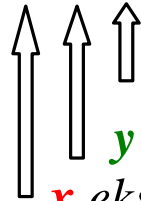


# plot Komutu ile Grafik Çizimi

## grafik çizgi-işaret stillerinin değiştirilmesi

- ❑ **plot** komutu ile grafikler düz çizgi tarzındadır.
- ❑ Farklı türde çizgi ve işarete sahip grafik çizdirmek için **plot** komutu aşağıdaki gibi kullanılmalıdır.

**plot(x,y,'c')**



*çizimde kullanılacak çizgi / renk tanımlaması*

*y eksenine ait vektörel ifade*

*x eksenine ait vektörel ifade*

### Çizgi çeşitleri

Düz çizgi	-	İki noktalı	:
Kesikli çizgi	--	Kesikli-noktalı	-.

### İşaret çeşitleri

Nokta	.	Üçgen (aşağı)	v
Artı	+	Üçgen (yukarı)	^
Yıldız	*	Üçgen (sola)	<
Daire	o	Üçgen (sağa)	>
x-işareti	x	Beş köşeli	p
Kare	s	Altı köşeli	h
Baklava şekli	d		



# plot Komutu ile Grafik Çizimi

## grafik çizgi-işaret stillerinin değiştirilmesi

- ❑ **Örnek:** `plot` komutu ile **kesik çizgili** ve **daire** işaretlerine sahip grafik çizimi.



### Komut penceresi

*% 0.1 artışlar ile 0 – 10 sn zaman diliminin tanımlanması*

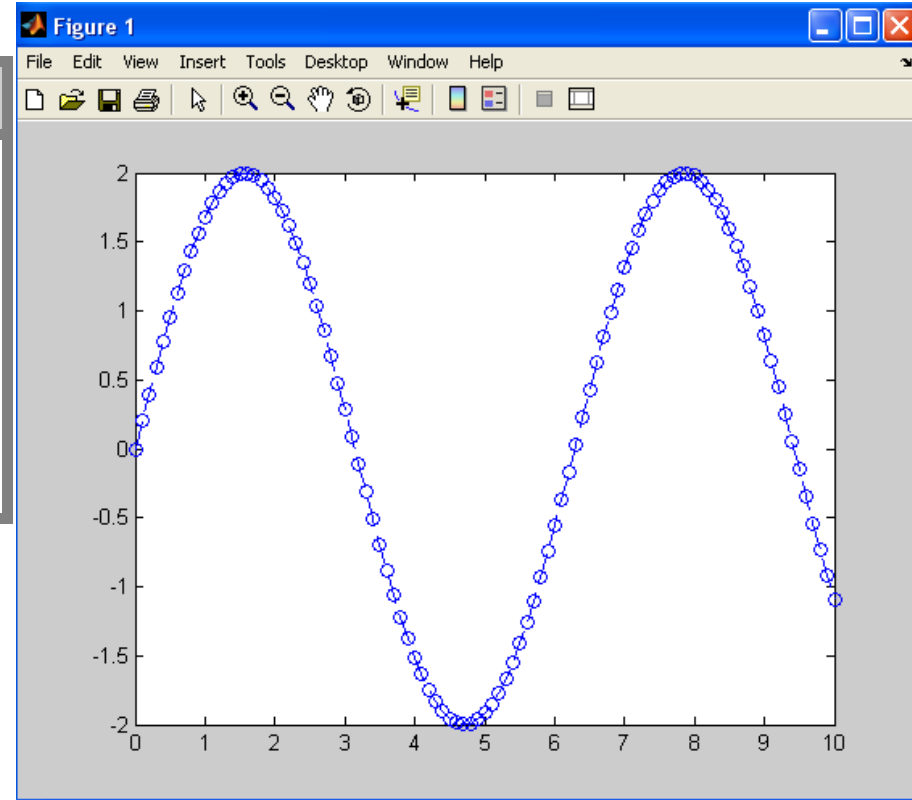
```
>> t = 0: 0.1 : 10;
```

*% Grafiğin y eksenini oluşturacak u(t) sinyalinin tanımlanması*

```
>> u = 2*sin(t);
```

*% Grafiğin kesik çizgili ve daire işaretleri ile çizdirilmesi*

```
>> plot(t,u, '-o')
```



## plot Komutu ile Grafik Çizimi

### grafik çizgi renklerinin değiştirilmesi

- Renk tanımlamalarında genel olarak, renklere ait ingilizce kelimelerin baş karakterleri kullanılmaktadır. Örneğin kırmızı için **red 'r'**

#### Renk çeşitleri

Kırmızı	r	Beyaz	w
Yeşil	g	Siyah	k
Mavi	b	Çıyan	c
Sarı	y	Maganda	m
Görünmez	i		

- Standart renk tanımlamalarının dışında [**r g b**] bir başka değişle [**kırmızı yeşil mavi**] şeklinde vektörel tanımlamada yapılabilir.

- Vektörel tanımlamadaki sayısal değerler 0 ile 1 arasında olmalıdır. 0 rengin olmayacağını, 1 oluşturulacak renk içerisindeki ana rengi gösterir. 0.4 gibi bir değer ise o renge ait katkı miktarını gösterir.

#### Vektörel olarak tanımlanmış bazı renk çeşitleri

[1 1 1] Beyaz	[1 0 0] Kırmızı	[1 1 0] Sarı
[0 0 0] Siyah	[0 1 0] Yeşil	[1 0 1] Pembe
	[0 0 1] Mavi	[0 1 1] Açık mavi

# plot Komutu ile Grafik Çizimi

## grafik çizgi-işaret ve renk stillerinin değiştirilmesi

- ❑ **Örnek:** `plot` komutu ile **iki noktalı çizgili** ve **kare** işaretlerine sahip **kırmızı** renkli grafik çizimi.



### Komut penceresi

*% 0.1 artışlar ile 0 – 10 sn zaman diliminin tanımlanması*

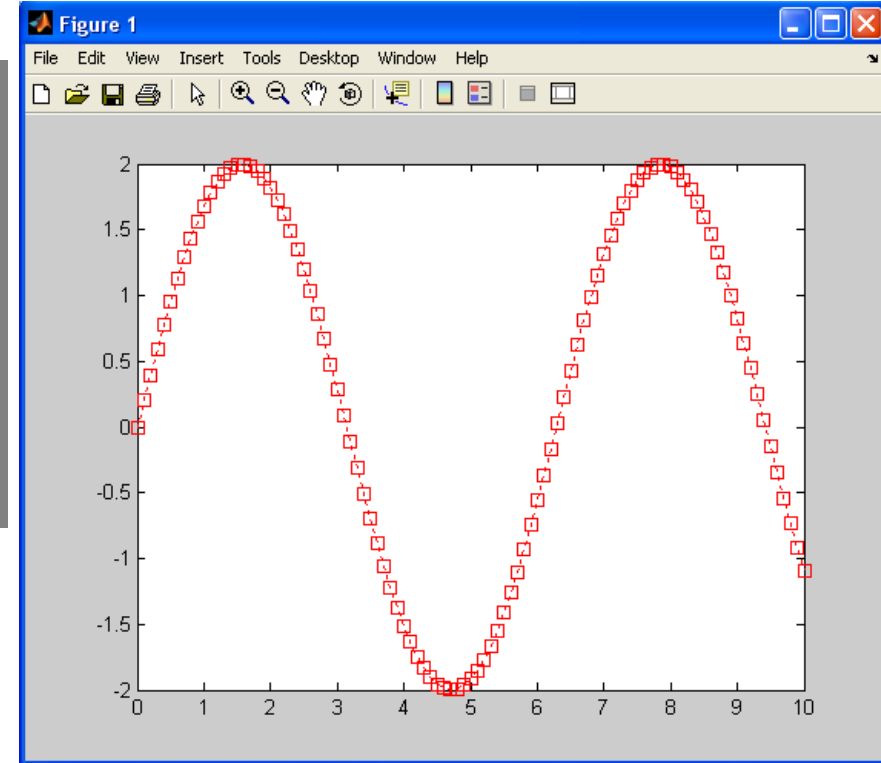
```
>> t = 0: 0.1 : 10;
```

*% Grafiğin y eksenini oluşturacak u(t) sinyalinin tanımlanması*

```
>> u = 2*sin(t);
```

*% iki noktalı, kare işaretli ve kırmızı renkte grafik*

```
>> plot(t,u, 's r')
```



# plot Komutu ile Grafik Çizimi

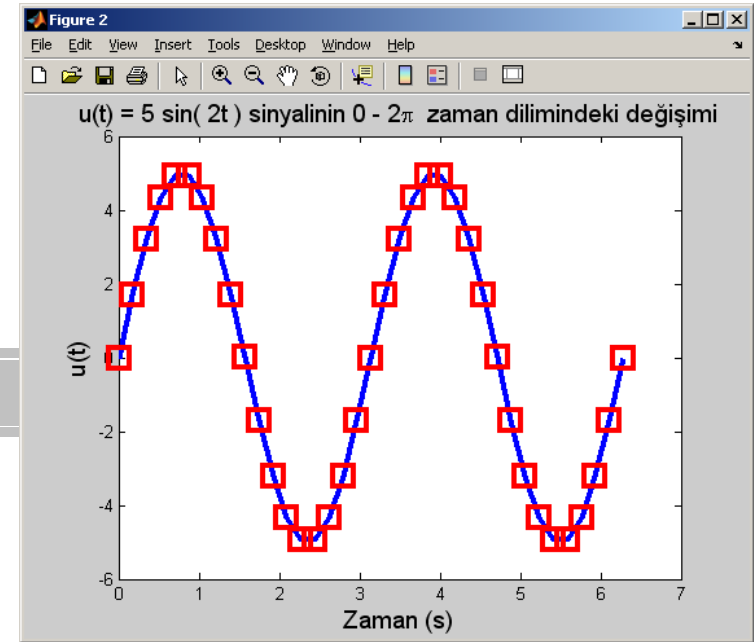
## plot komutunun diğer özellikleri

`plot(x,y,'LineStyle','--',...  
'Color','r',...  
'LineWidth',2,...  
'Marker','square',...  
'MarkerEdgeColor','k',...  
'MarkerFaceColor','g',...  
'MarkerSize',10)`



PROGRAM 3.3

```
1 % Grafik çiziminde yer alan u(t) sinyalinin tanımlanması
2 - t = 0:pi/18:2*pi;
3 - u = 5*sin(2*t);
4
5 % Grafik çiziminin plot komutu ile gerçekleştirilmesi
6 - plot(t,u,'-bs','linewidth',3,'MarkerEdgeColor','[1 0 0]', ...
7     'MarkerSize',16)
8
9 % Grafik üzerinde eksen açıklamalarının yapılması
10 - xlabel('Zaman (s)','fontsize',14)
11 - ylabel('u(t)','fontsize',14)
12 - title('u(t) = 5 sin( 2t ) sinyalinin 0 - 2\pi zaman dilimindeki ...
    değişimi','fontsize',14)
```



# plot Komutu ile Grafik Çizimi

## tek bir figürde **birden fazla grafik çizimi**

- ❑ Tek bir figure içerisinde **farklı özelliklere sahip** birden fazla grafik çizdirilmesi istenirse,

`plot(x1,y1,'c1',x2,y2,'c2', ... , xn,yn,'cn')`

*birinci grafiğe ait çizgi ve renk çeşidi*  
*birinci grafiğe ait vektörel ifadeler*

*ikinci grafiğe ait çizgi ve renk çeşidi*  
*ikinci grafiğe ait vektörel ifadeler*

*n\'inci grafiğe ait çizgi ve renk çeşidi*  
*n\'inci grafiğe ait vektörel ifadeler*

# plot Komutu ile Grafik Çizimi

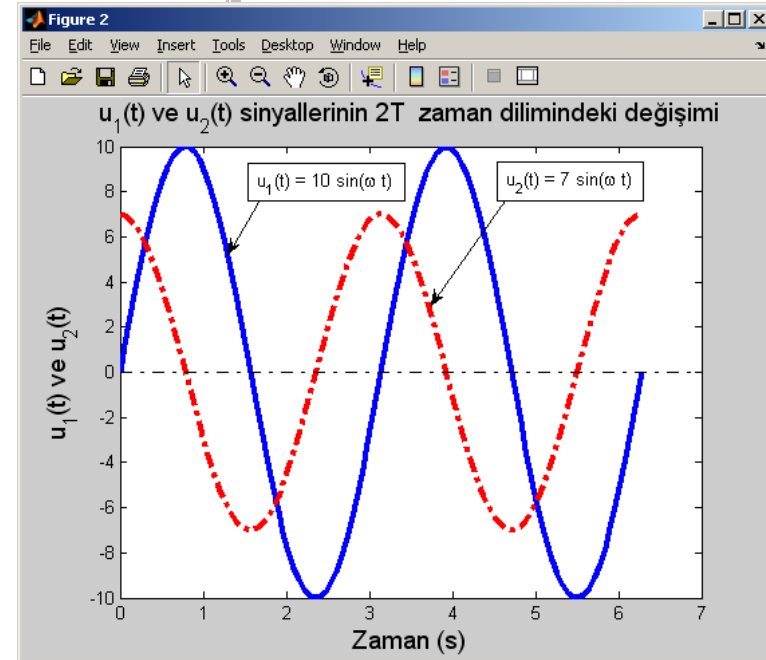
## tek bir figürde birden fazla grafik çizimi

- ❑ **Örnek:** Aşağıda belirtilen işlemleri bir **m.file** içerisinde yapınız.
- $u_1(t) = 10\sin(\omega t)$  ve  $u_2(t) = 7\cos(\omega t)$  iki ayrı sinyali tanımlayınız.  $\omega = 2$  rad/sn
  - Sinyallerin iki (2) periyotluk değişimlerini **tek bir grafik üzerinde** karşılaştırınız.



### PROGRAM

```
1 % Grafik çiziminde kullanılacak u1(t) ve u2(t) sinyallerinin 2*T'ye göre tanımlanması
2 - w = 2;
3 - T = 2*pi/w;
4 - t = linspace(0,2*T);
5 - u1 = 10*sin(w*t);
6 - u2 = 7*cos(w*t);
7
8 % Grafik çiziminin tek plot komutu ile gerçekleştirilmesi
9 - plot(t,u1,'-b',t,u2, '-.r', 'linewidth',3)
10
11 % Grafik üzerinde eksen açıklamalarının yapılması
12 - xlabel('Zaman (s)','fontsize',14)
13 - ylabel('u_1(t) ve u_2(t)','fontsize',14)
14 - title('u_1(t) ve u_2(t) sinyallerinin 2T zaman dilimindeki ...
15         değişimi','fontsize',14)
```



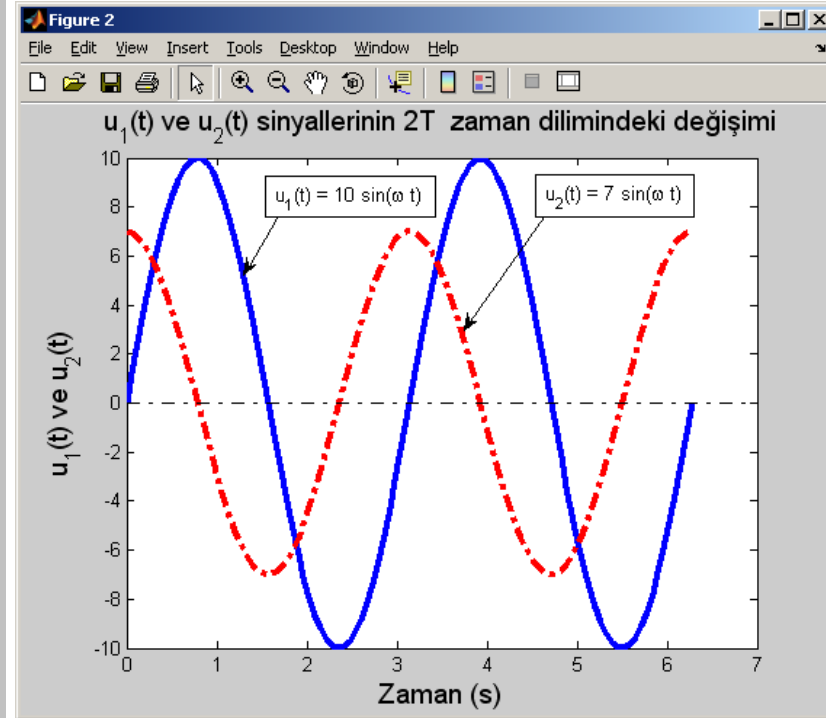
## hold on komutu ile tek bir figürde birden fazla grafik çizimi

- ❑ Önceki örnekte elde edilen çizimi sıra ile elde ederek tek bir grafikte gösterelim.
- ❑ İlk önce  $u_1(t)$  sinyali çizdirilir.
- ❑ **hold on** komutu çizdirilmiş grafiğin figür penceresinde tutulmasını sağlar.
- ❑ **hold on** komutu kullanıldıktan sonra çizdirilen grafik aynı figüre eklenir.
- ❑ **hold on** komutunu iptal etmek için **hold off** kullanılır.



### PROGRAM

```
1      % u1(t)'e ait grafiğin çizimi
2      - plot(t,u1,'-b' 'linewidth',3)
3
4      % birinci grafiğin figür penceresinde tutulması
5      - hold on
6
7      % u2(t)'e ait grafiğin çizimi
8      - plot(t,u2,'-.r' 'linewidth',3)
9
10     % elde edilen grafik üzerine eksen çizgisinin eklenmesi
11     - plot([0 7],[0 0],'-k')
12
```



# plot Komutu ile Grafik Çizimi

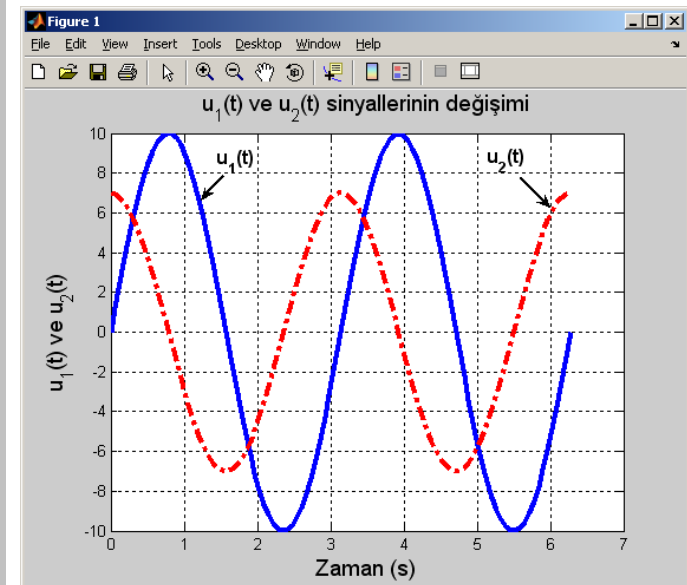
## grid komutu ile grafiği yatay ve dikey bölümlendirme

- ❑ Grafiklerin daha rahat okunabilmesi için yatay ve dikey çizgiler ile bölüm oluşturur.
- ❑ **grid on** çizgileri ekler.
- ❑ **grid off** çizgileri kaldırır.



### PROGRAM

```
1 % Grafik çiziminde kullanılacak u1(t) ve u2(t) sinyallerinin 2*T'ye göre çizimi
2 - w = 2;
3 - T = 2*pi/w;
4 - t = linspace(0,2*T);
5 - u1 = 10*sin(w*t);
6 - u2 = 7*cos(w*t);
7 - plot(t,u1,'-b',t,u2, '-.r', 'linewidth',3)
8
9 % grid on komutu ile ızgaralamanın oluşturulması
10 - grid on
```



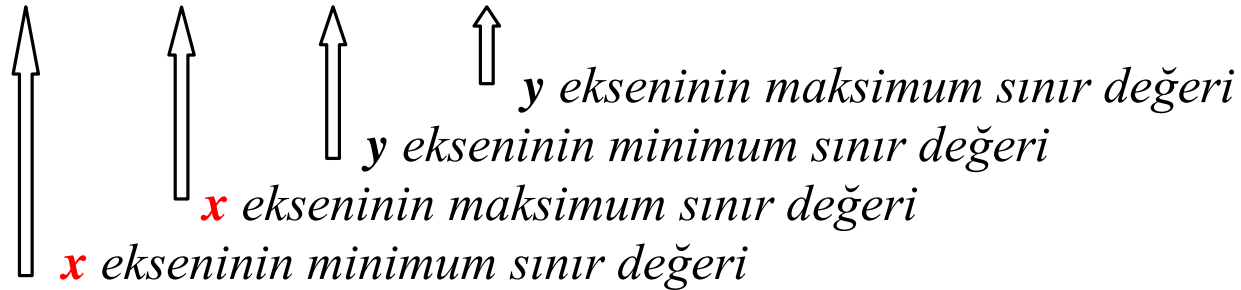


## plot Komutu ile Grafik Çizimi

### axis komutu ile eksen ölçeklendirme

- ❑ Grafiğe ait eksen ölçeklendirmesini **istenilen değerlere** göre yeniden düzenler.

**axis ( [xmin xmax ymin ymax] )**

  
x ekseninin minimum sınır değeri  
x ekseninin maksimum sınır değeri  
y ekseninin minimum sınır değeri  
y ekseninin maksimum sınır değeri

# plot Komutu ile Grafik Çizimi

## axis komutu ile eksen ölçeklendirme

- ❑ Örnek:  $u(t) = 2\sin(\omega t)$  sinyalini 0.01 adımlarla, 0 ile 10 sn zaman dilimi için çiziniz? Not:  $\omega = 1$
- ❑ Ardından grafiğin x eksenini 0 - 12, y eksenini ise -3 ile +3 olarak yeniden ölçeklendiriniz.



### Komut penceresi

*% 0.01 artışlar ile 0 – 10 sn zaman diliminin tanımlanması*

```
>> t = 0: 0.01 : 10;
```

*% Grafiğin y eksenini oluşturacak u(t) sinyalinin tanımlanması*

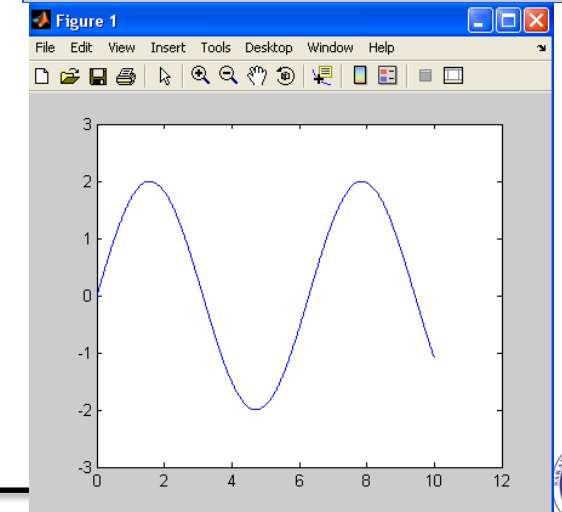
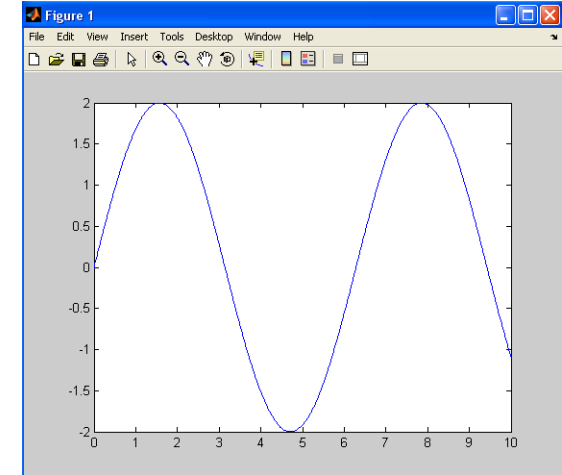
```
>> u = 2*sin(t);
```

*% Grafiğin çizdirilmesi*

```
>> plot(t,u)
```

*% Grafiğin eksenlerinin yeniden ölçeklendirilmesi*

```
>> axis([ 0 12 -3 3 ])
```



# fplot komutu ile grafik çizme

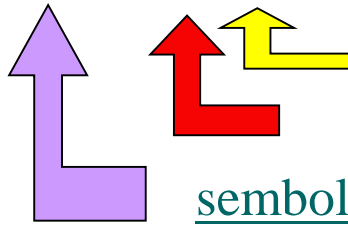
- ❑ **fplot** komutunun genel kullanımı
- ❑ **renk, şekil, kalınlık, çizgi çeşidi** gibi grafiklerin özelliklerinin değiştirilmesi

# fplot Komutu ile Grafik Çizimi

## genel kullanımı

- ❑ Bir fonksiyona ait grafiğin tanımlanan sınır değerlerine göre ( $x_1$  ve  $x_2$  aralığında) çizimini yapar.

- ❑ `fplot('F', [x1 x2])`



$x$  ekseninde istenen son sınır değeri

$x$  ekseninde istenen ilk sınır değeri

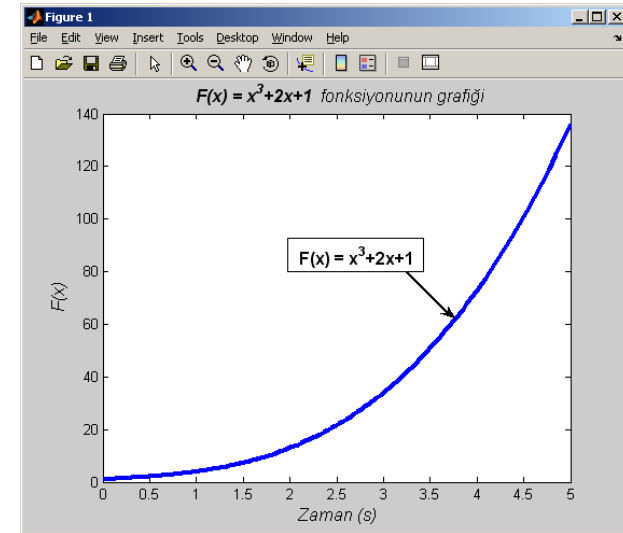
sembolik olarak fonksiyonun tanımlanması

- ❑ **Örnek:**  $F(x) = x^3 + 2x + 1$  fonksiyonuna ait 0 - 5 sn aralığındaki değişimini çizdiren programı yazınız?



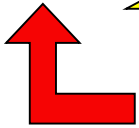
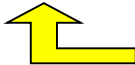
### PROGRAM

```
1 % F(x) fonksiyonunun tanımlanarak 0-5 saniye arasındaki grafiğinin çizimi
2 - fplot('x^3+2*x+1',[0 5])
3
4 - xlabel('\it Zaman (s)','fontsize',12)
5 - ylabel('\it F(x)','fontsize',12)
6 - title('{\bf \it F(x) = x^3+2x+1} \it fonksiyonunun...
7      grafiği','fontsize',12)
```



## fplot Komutu ile Grafik Çizimi

### çizgi çeşidi ve renginin değiştirilmesi

- ❑ Plot komutunda olduğu gibi fplot komutunda çizgi çeşidi ve rengi değiştirilebilir.
- ❑ Ayrıca tanımlanan  $x_1$  ve  $x_2$  aralığındaki örnekleme adedinin tanımlanacak bir tolerans değeri ile değiştirilmesine imkan tanır.
- ❑ `fplot ( 'F' , [  $x_1$   $x_2$  ] , tol , 'çri' )`
  -  çizgi çeşidi, rengi ve işaretleme çeşidi
  -  tanımlanan tolerans değeri

# fplot Komutu ile Grafik Çizimi

## çizgi çeşidi ve renginin değiştirilmesi

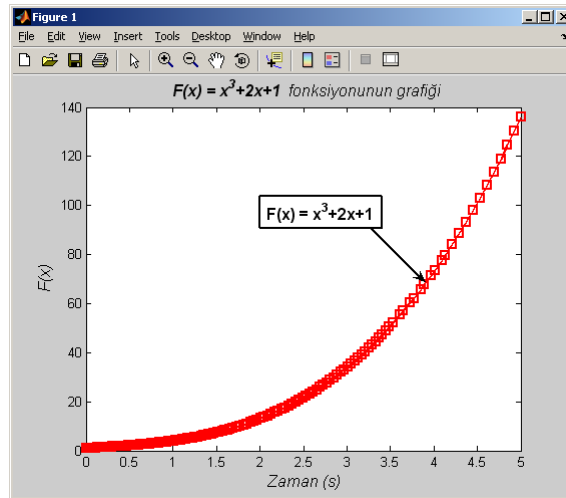
- ❑ **Örnek:**  $F(x) = x^3 + 2x + 1$  fonksiyonuna ait 0 - 5 sn aralığındaki değişimini düz çizgili, kırmızı renkli ve kare işaretli olarak çizdiren programı yazınız?



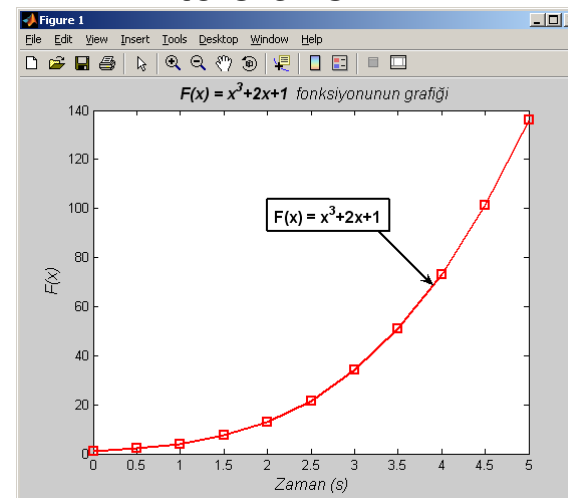
### PROGRAM

```
1 % F(x) fonksiyonunun tolerans değeri tanımlanmadan elde edilen 0-5 saniye aralığındaki
2 % grafiğinin çizimi (düz çizgi, kırmızı renkli ve kare kutularla işaretleme şekli)
3 - fplot('x^3+2*x+1',[0 5], '-rs')
4
5 % F(x) fonksiyonunun tolerans değeri 0,1 tanımlandığında elde edilen 0-5 saniye
6 % aralığındaki grafiğinin çizimi (düz çizgi, kırmızı renkli ve kare kutularla işaretleme şekli)
7 - fplot('x^3+2*x+1',[0 5],0.1, '-rs')
```

toleranssız



toleranslı



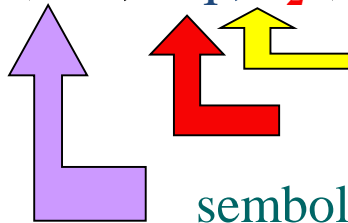
# ezplot komutu ile grafik çizme

- ❑ **ezplot** komutunun genel kullanımı
- ❑ Bir örnek uygulama

## genel kullanımı

- ❑ **fplot** komutu gibi bir fonksiyona ait grafiğin tanımlanan sınır değerlerine göre ( $x_1$  ve  $x_2$  aralığında) çizimini yapar.

- ❑ **ezplot** ( 'F' ,  $x_1$  ,  $x_2$  )



$x$  ekseninde istenen son sınır değeri

$x$  ekseninde istenen ilk sınır değeri

sembolik olarak fonksiyonun tanımlanması

- ❑ **fplot**'un kullanımından farklı olarak fonksiyonda kullanılan sembol **syms** komutu ile önceden tanımlanabilir.

✓ **syms** fonksiyondaki değişkene ait sembolik ifade

✓ **ezplot** ( 'F' ,  $x_1$  ,  $x_2$  )

- ❑ veya

✓ **ezplot** ( 'F' , [  $x_1$   $x_2$  ] )

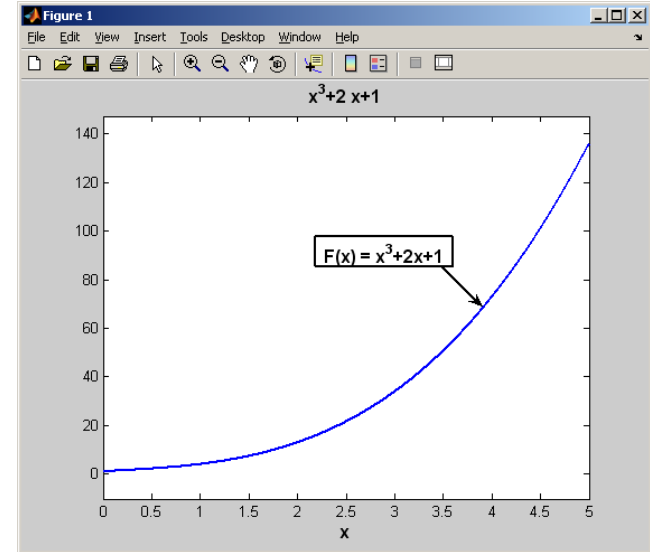


- ❑ **Örnek:**  $F(x) = x^3 + 2x + 1$  fonksiyonuna ait 0 - 5 sn aralığındaki değişimini çizdiren programı yazınız?



## PROGRAM

```
1 % F(x) fonksiyonunun tanımlanarak 0-5 saniye arasındaki grafiğin çizimi
2 - ezplot('x^3+2*x+1', 0, 5)
3
4 % veya
5
6 % F(x) fonksiyonunda yer alan x değişkeninin sembolik olarak
7 % tanımlanıp 0-5 saniye aralığındaki grafiğinin çizimi
8 - syms x
9 - ezplot(x^3+2*x+1, 0, 5)
10
11 % veya
12
13 - ezplot(x^3+2*x+1, [0, 5])
```



otomatik olarak

- ❖ x eksenini açıklaması
- fonksiyondaki değişken,
- ❖ grafik başlığı ise
- tanımlanan fonksiyondur.

# grafikler üzerinde düzenlemeler

- ❑ `gtext` komut ile açıklama ekleme
- ❑ `\rm` `\bf` `\it` ile açıklamaları düzenleme
- ❑ `legend` komutu ile açıklama yazma
- ❑ `ginput` komutu ile değer okuma
- ❑ `semilogx`, `semilogy` ve `loglog` komutları ile logaritmik grafik

# Grafikler Üzerinde Düzenlemeler

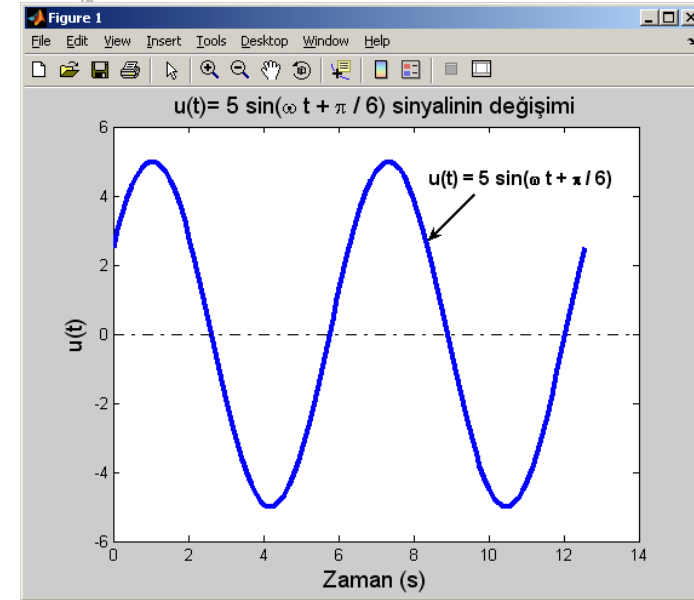
## gtext komutu ile açıklama yazma

- ❑ Grafik penceresi üzerinde açıklama yazılmasını sağlar.
- ❑ `gtext ( ' açıklama metni ' )`



### PROGRAM

```
1 % u(t) sinyalinin tanımlanması
2 - w = 1;
3 - T = 2*pi/w;
4 - t = linspace(0,2*T);
5 - u = 5*sin(1*w*t+pi/6);
6
7 % Grafik çiziminin plot komutu ile gerçekleştirilmesi
8 - plot(t,u,'-b','linewidth',3)
9 - hold on
10 - axis([0 14 -6 6])
11 - plot([0 14],[0 0],'-.k')
12 - xlabel('Zaman (s)','fontsize',14)
13 - ylabel('u(t)','fontsize',14)
14 - title('u(t)= 5 sin(\omega t + \pi / 6) sinyalinin değişimi', ...
15         'fontsize',14)
16 % Grafik üzerine u(t) ifadesinin yazdırılması
17 - gtext('u(t) = 5 sin ( \omega t + \pi /6) ')
18
```



### `\rm` `\bf` `\it` ile açıklama ve eksen/başlık yazılarının düzenlenmesi

- ☐ `\rm` normal ölçüsüne geri dönüşüm yapar (**restore normal format**)
- ☐ `\bf` yazı tipini kalınlaştırır (**boldface**)
- ☐ `\it` yazının şeklini italik yapar (**italics**)
- ☐ Bu komutlar değişiklik yapılacak **açıklamanın önünde** kullanılır.
- ☐ Bir açıklama içerisinde bu özelliklerden aynı anda birden fazla kullanmak gerekiyorsa ilgili bölümler {...} içerisinde kullanılması daha uygun olur.
- ☐ {...} kullanım sadece kendi içerisindeki değişikliği içerir, dışındakileri etkilemez.
- ☐ Aynı zamanda bir açıklamanın **hem kalın hem de italik** olması istenirse `\bf` ve `\it` ard arda kullanılır.

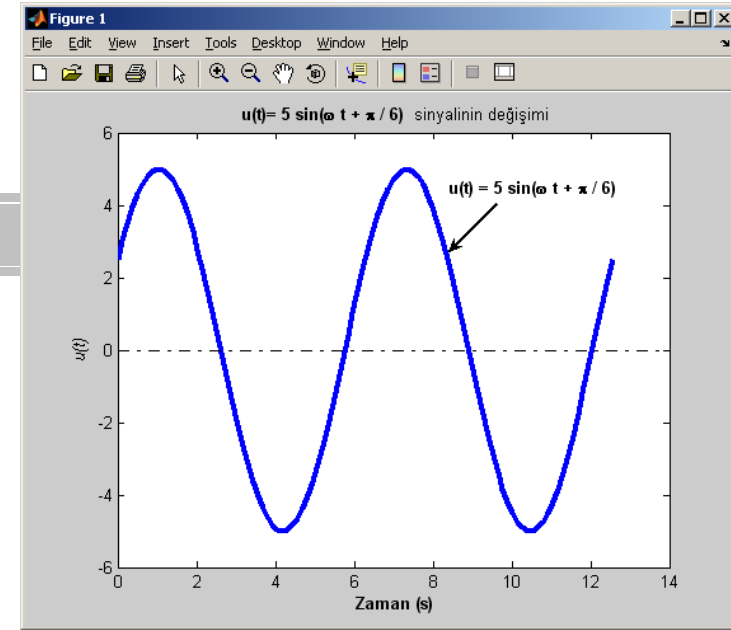
## $\backslash rm$ $\backslash bf$ $\backslash it$ ile açıklama ve eksen/başlık yazılarının düzenlenmesi



### PROGRAM

```

1      % x eksenine açıklamanın kalın bir şekilde yazılması
2      - xlabel(' \bf Zaman (s) ')
3
4      % y eksenine açıklamanın italik bir şekilde yazılması
5      - ylabel(' \it u(t) ')
6
7      % başlığa formülün kalın devamındaki yazının normal bir şekilde yazılması
8      - title(' \bf u(t)= 5 sin(\omega t + \pi / 2) \rm sinyalinin değişimi ')
9
10     % veya
11
12     - title('{ \bf u(t)= 5 sin(\omega t + \pi / 2) } sinyalinin değişimi ')
13
14     % formülün dışında kalan açıklamanın hem koyu hemde italik olarak yazılması
15     - title('{ \bf u(t)= 5 sin(\omega t + \pi / 2) } { \bf \it sinyalinin ...
16           değişimi } ')
    
```



# Grafikler Üzerinde Düzenlemeler

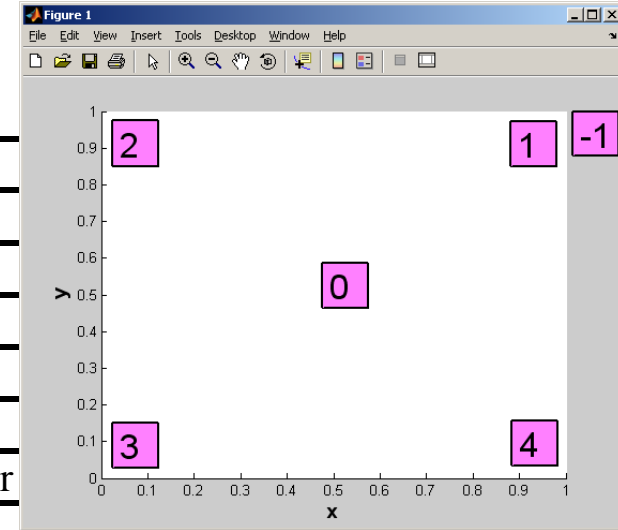
## legend komut ile açıklama ekleme

- Tanımlanan konuma göre figür penceresi üzerinde bir kutu açarak çizim sırasına göre ilgili grafiklerde kullanılan çizim şekli ve rengi göstererek açıklama yazılmasını sağlar.

- `legend ( 'açıklama 1', 'açıklama 2', konum )`  
figüre penceresindeki konum  
2. grafiğe ait açıklama  
1. grafiğe ait açıklama

- Konumu belirten sayısal değerler

Konum tanımlaması	Açıklama kutusunun konumu
0	Grafik penceresine otomatik olarak yerleştirilir
1	Grafik penceresinin sağ üst köşesine yerleştirilir
2	Grafik penceresinin sol üst köşesine yerleştirilir
3	Grafik penceresinin sol alt köşesine yerleştirilir
4	Grafik penceresinin sağ alt köşesine yerleştirilir
-1	Grafik penceresinin dışında sağ üst köşeye yerleştirilir



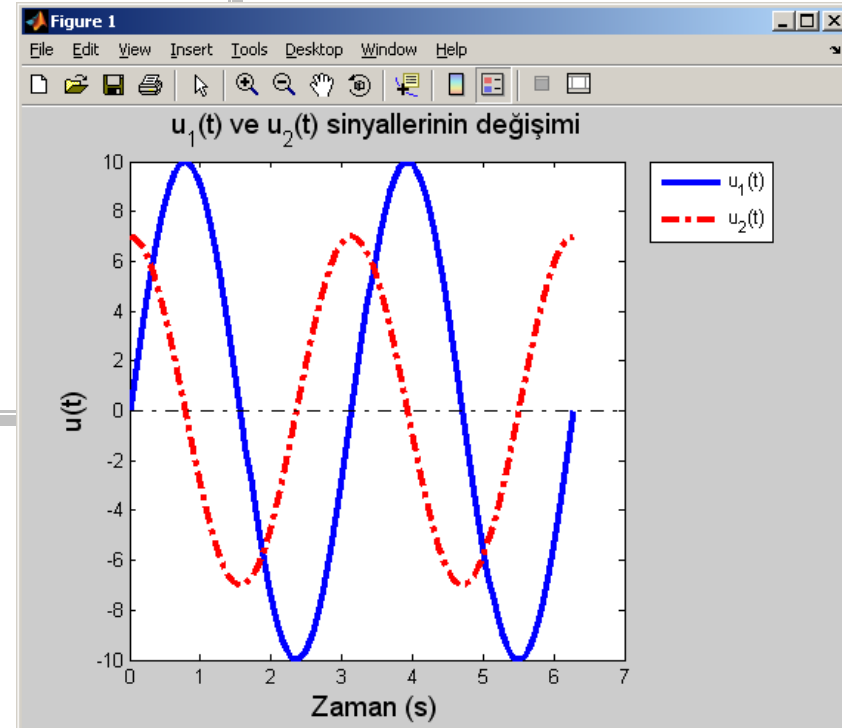
# Grafikler Üzerinde Düzenlemeler

## legend komut ile açıklama ekleme - Örnek -




### PROGRAM

```
1      % Grafik çiziminde kullanılacak u1(t) ve u2(t) sinyallerinin 2*T'ye göre çizimi
2      - w = 2;
3      - T = 2*pi/w;
4      - t = linspace(0,2*T);
5      - u1 = 10*sin(w*t);
6      - u2 = 7*cos(w*t);
7      - plot(t,u1,'-b',t,u2, '-.r', 'linewidth',3)
8      - hold on
9      - plot([0 7],[0 0], '-.k')
10
11     % legend komutu ile açıklama kutusunun oluşturulması
12     - legend('u_1(t)', 'u_2(t)', -1)
13
```



# Grafikler Üzerinde Düzenlemeler

## ginput komutu ile değer okuma

- ❑ Grafik üzerinde fare vasıtasıyla belirtilen nokta veya noktaların koordinatlarını **komut penceresinde sayısal olarak** elde etmeyi sağlar.
- ❑ Bu komutun kullanılabilmesi için figür penceresinin açık olması gerekir.
- ❑ Komutun kullanımı ile figür penceresi üzerinde farenin hareketine göre konum değiştiren eksenlere paralel iki adet doğru parçası görünür. Doğru parçalarının kesişim noktaları istenilen pozisyona getirildiğinde fare vasıtasıyla tıklandığı zaman o noktaya ait koordinatların komut penceresinde sayısal olarak ortaya çıkar.
- ❑ `ginput ( n )`  
 grafik üzerinde işaretlenecek nokta sayısı



# Grafikler Üzerinde Düzenlemeler

## ginput komutu ile değer okuma - Örnek -

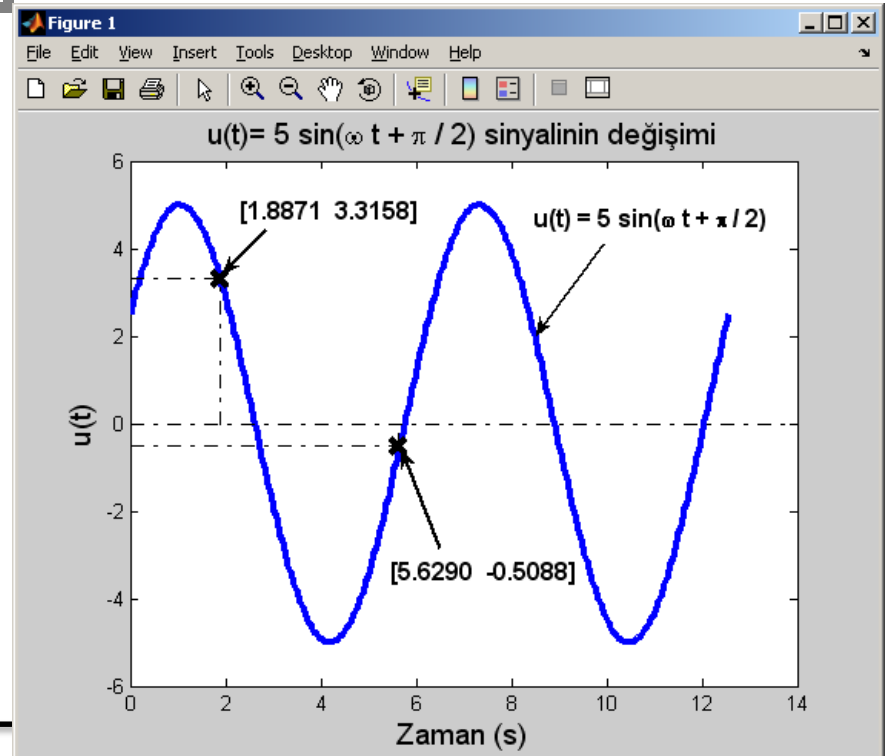


### Komut penceresi

% Grafik üzerinde iki adet noktaya ait koordinatların bulunması  
>> **ginput**(2)

ans =

1.8871	3.3158
5.6290	-0.5088



# Grafikler Üzerinde Düzenlemeler

## semilogx, semilogy ve loglog komutları ile istenilen eksenli logaritmik çizdirmek

MATLAB KOMUTU	X EKSENİ	Y EKSENİ	GRAFİKSEL SONUÇ
<b>semilogx</b>	logaritmik	doğrusal	
<b>semilogy</b>	doğrusal	logaritmik	
<b>loglog</b>	logaritmik	logaritmik	

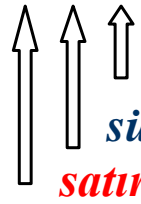
# subplot komutu ile figür penceresini bölme

- ❑ subplot komutunun genel kullanımı
- ❑ Bir örnek uygulama

# subplot komutu ile figür penceresini bölme

- ❑ Figür penceresini istenilen sayıda pencerelere bölerek çizimin yapılacağı pencerenin adreslenmesini sağlar.

subplot(**m**,**n**,p)

 *çizimin yapılacağı pencereye ait adres numarası*  
*sütun sayısı*  
*satır sayısı*

- ❑ subplot komutunun kullanımı sonucunda figür penceresi  $m*n$  adet parçaya bölünmüş olur.
- ❑ Çizimin yapılacağı pencereye ait adres birinci satır birinci sütundaki pencereden başlanılarak satır satır numaralanmak suretiyle ortaya çıkan matris yapıdan elde edilir.

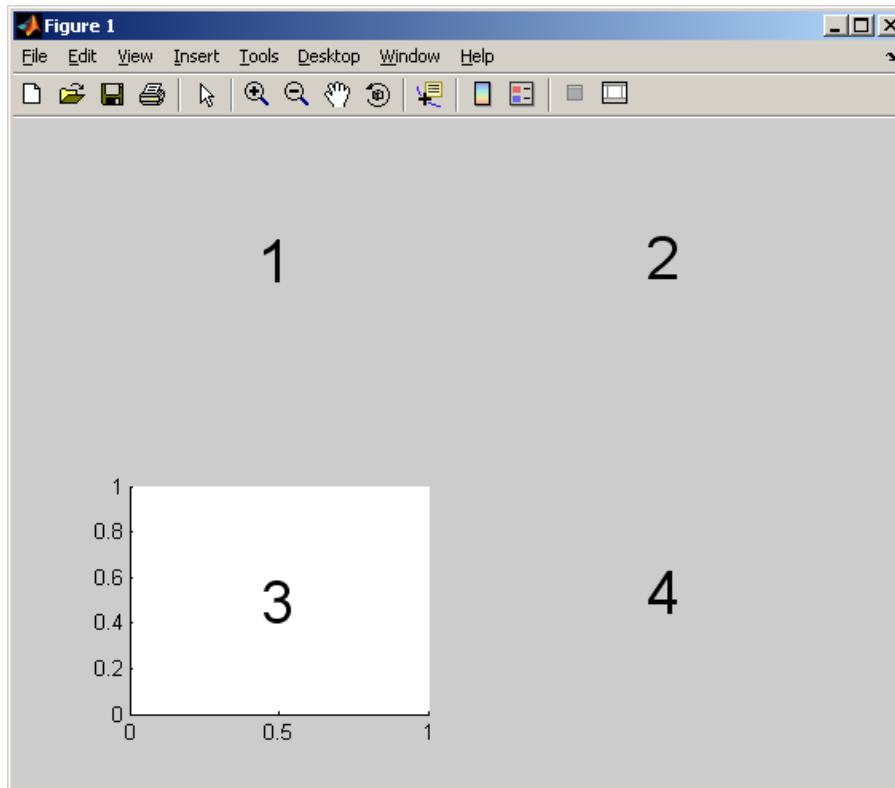
# subplot komutu ile figür penceresini bölme

`subplot ( 2 , 2 , 3 )`  
2 satır      2 sütun      3. pencere



## PROGRAM

```
1      % Figürün 2 satır 2 sütuna bölünerek 3'üncü penceresinin seçilmesi
2      - subplot ( 2 , 2 , 3 )
3
```



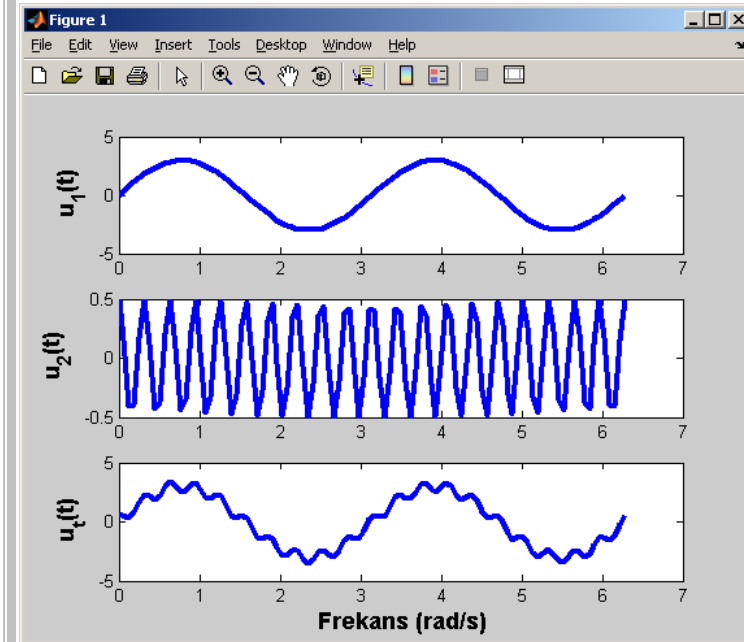
# subplot komutu ile figür penceresini bölme

- ❑ **Örnek :**  $u_1(t) = 3 \sin(\omega t)$  ve  $u_2(t) = 0.5 \cos(\omega t)$  sinyalleri ile bu iki sinyalin toplamını aynı figür penceresi içerisinde çizdiriniz?
- ❑  $\omega = 2 \text{ rad/s}$  ve sinyallerin değişimi  $\omega$ 'ye bağlı 2 periyotluk dilim için olacak



## PROGRAM

```
1 % Grafik çiziminde kullanılacak t zamanının açısal frekansa göre tanımlanması
2 - w = 2;
3 - T = 2*pi/w;
4 - t = linspace(0,2*T);
5
6 % Sinyallerin oluşturulması
7 - u1 = 3*sin(w*t);
8 - u2 = 0.5*cos(10*w*t);
9 - ut = u1+u2;
10
11 % Figür penceresinin bölünerek sinyallerin çizimi ve eksen açıklamalarının yapılması
12 - figure(1); clf
13 - subplot(311); plot(t,u1,'-b','linewidth',3)
14 - ylabel('\bf u_1(t)','fontsize',14)
15 - subplot(312); plot(t,u2,'-b','linewidth',3)
16 - ylabel('\bf u_2(t)','fontsize',14)
17 - subplot(313); plot(t,ut,'-b','linewidth',3)
18 - xlabel('\bf Frekans (rad/s)','fontsize',14)
19 - ylabel('\bf u_t(t)','fontsize',14)
20
```



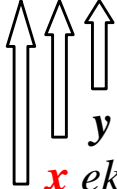
# özel grafikler

- ❑ **plot3** komutu ile 3 boyutlu çizgi grafik çizme
- ❑ **bar** komutu ile çubuk grafik çizme
- ❑ **barh** komutu ile yatay çubuk grafik çizme
- ❑ **bar3** komutu ile 3 boyutlu çubuk grafik çizme
- ❑ **stem** komutu ile grafik çizme
- ❑ **stem3** komutu ile 3 boyutlu grafik çizme
- ❑ **pie** komutu ile pasta grafik çizme
- ❑ **pie3** komutu ile 3 boyutlu pasta grafik çizme
- ❑ **polar** komutu ile kutupsal koordinatlı grafik çizme

# plot3 komutu ile 3 boyutlu grafik çizdirme

- 3 boyutlu grafik çizimini sağlar.

**plot3**(**x,y,z**)


  
 $z$  eksenine ait vektörel ifade  
 $y$  eksenine ait vektörel ifade  
 $x$  eksenine ait vektörel ifade

- **Örnek:** Aşağıda  $x$  ve  $y$  eksenlerindeki konumları tanımlayan denklem takımlarının zamana bağlı değişimini  $10\pi$  saniyelik süre için çizdiriniz?

$$x(t) = \sin(2t)(1 - e^{-0.1t})$$

$$y(t) = \cos(2t)(1 - e^{-0.1t})$$



# Özel Grafikler

## plot3 komutu ile 3 boyutlu grafik çizdirme



### Komut penceresi

*% Zaman aralığı*

```
>> t=linspace(0,5*2*pi,1000);
```

*% Fonksiyonlara ait hesaplamalar*

```
>> x=sin(2*t).*(1-exp(-t/10));
```

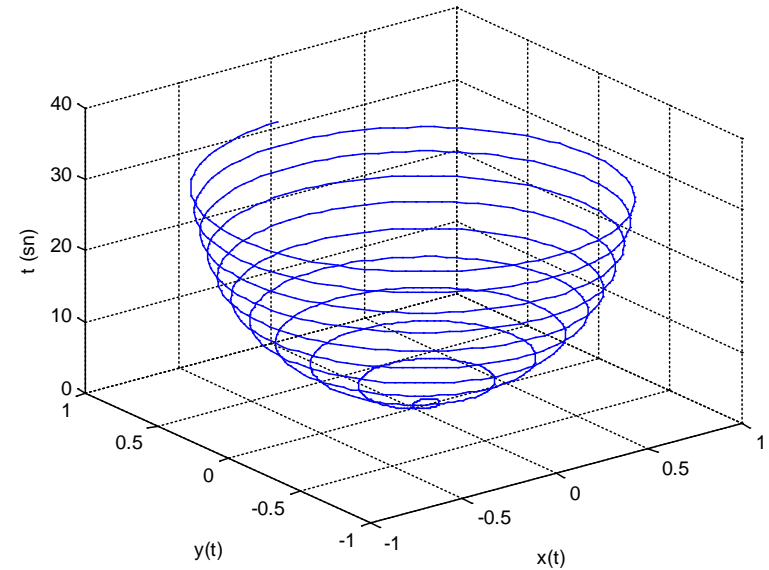
```
>> y=cos(2*t).*(1-exp(-t/10));
```


*% Üç boyutlu çizim işlemi*



```
>> plot3(x,y,t)
```

```
>> xlabel('x(t)');ylabel('y(t)');zlabel('t (sn)')
```

```
>> grid on
```

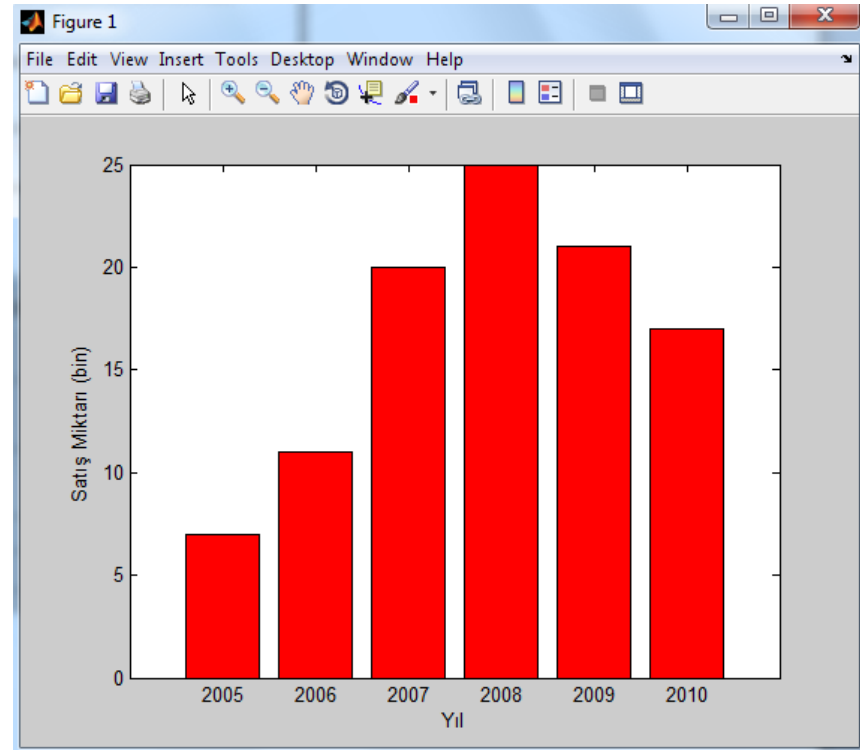


 **bar ( x, y )**

  **y eksenine ait değerler**  
**x eksenine ait değerler**

# PROGRAM

```
1 % Zaman aralığı
2 - yıl = [2005 : 2010];
3 % Yıllara göre satış miktarları
4 - satis= [ 7 11 20 25 21 17 ];
5 % Kırmızı dolgu rengine sahip çubuk grafik çiz
6 - bar(yıl, satis, 'r')
7 - xlabel('Yıl')
8 - ylabel('Satış Miktarı (bin)')
```



## barh komutu ile yatay çubuk grafik çizdirme

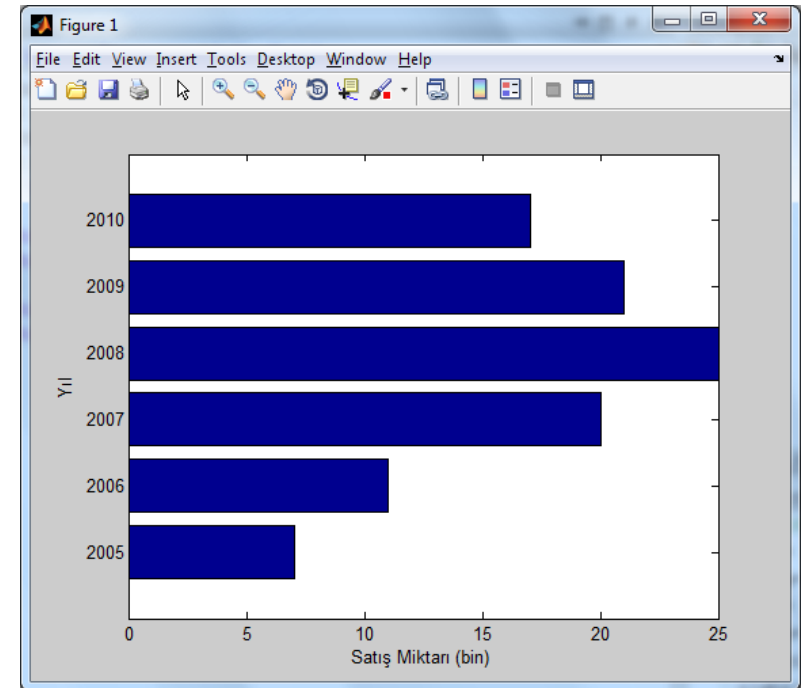
 **barh** ( **x**, **y** )

  **y** eksenine ait deęerler  
**x** eksenine ait deęerler



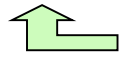
## PROGRAM

```
1 % Zaman aralığı
2 - yıl = [2005 : 2010];
3 % Yıllara göre satış miktarları
4 - satis= [ 7 11 20 25 21 17 ];
5 % Yatay çubuk grafik çizdir
6 - barh(yıl, satis)
7 - ylabel('Yıl')
8 - xlabel('Satış Miktarı (bin)')
```



## bar3 komutu ile 3 boyutlu çubuk grafik çizdirme

❑ bar3 (Y)



x,y,z koordinat değerlerine sahip matris.

Y'deki her eleman ayrı bir çubuktur.



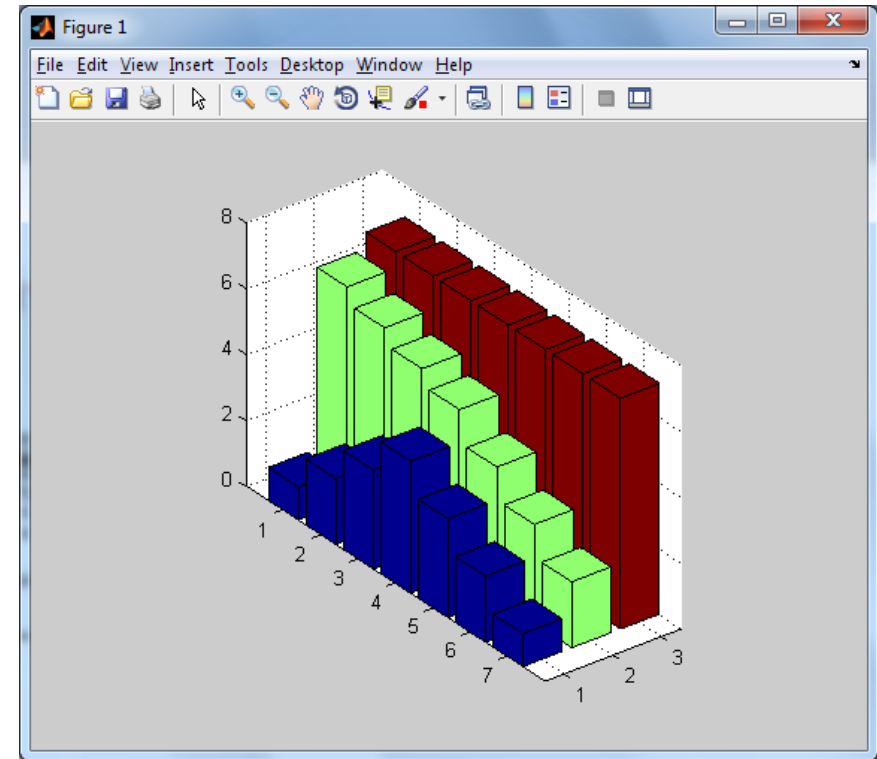
### Komut penceresi

*% 3 boyutlu grafiği çizdirilecek matrisin tanımlanması*

```
>> Y = [ 1 6.5 7; 2 6 7; 3 5.5 7; 4 5 7;  
        3 4 7; 2 3 7; 1 2 7 ];
```

*% 3 boyutlu çubuk grafik*

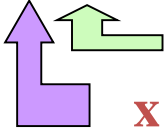
```
>> bar3(Y)
```



# Özel Grafikler

## stem komutu ile grafik çizdirme

❑ **stem ( x, y )**

 **y** eksenine ait değerler  
**x** eksenine ait değerler



### Komut penceresi

*% Zaman aralığı*

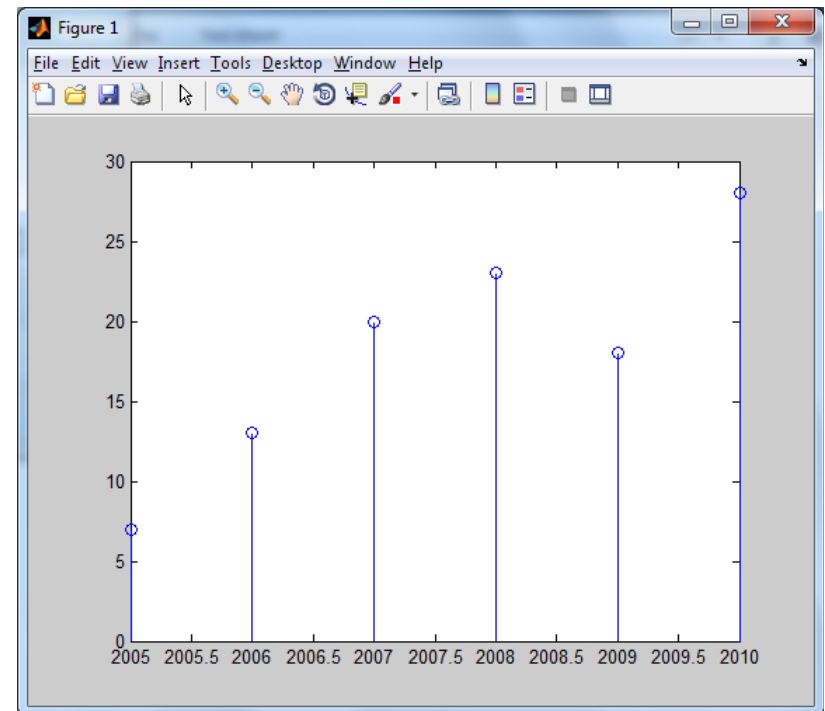
```
>> yıl = [ 2005 : 2010 ];
```

*% Yıllara göre satış miktarları*

```
>> satis = [ 7 13 20 23 18 28 ];
```

*% stem grafiğın çizdirilmesi*

```
>> stem( yıl, satis)
```



# Özel Grafikler

## stem3 komutu ile 3 boyutlu grafik çizdirme

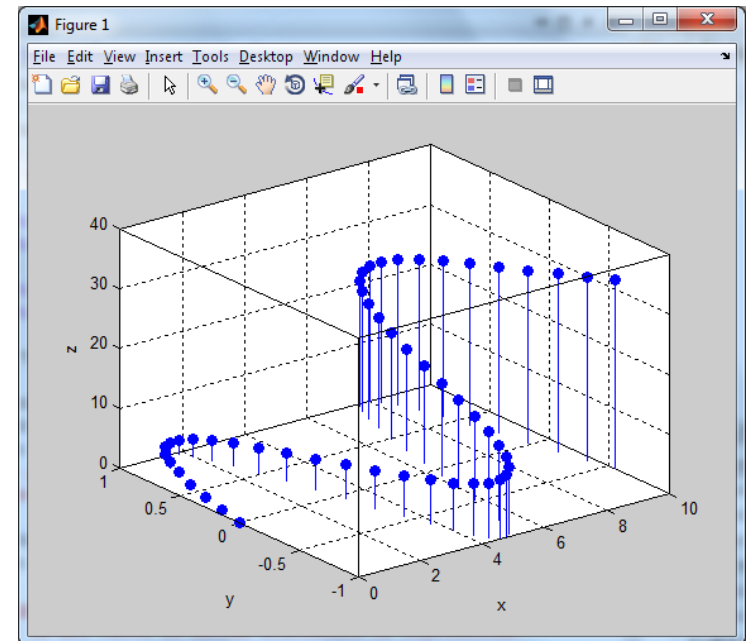
□ **stem3 ( x, y, z )**

↑ z eksenine ait değerler  
↑ y eksenine ait değerler  
↑ x eksenine ait değerler



### PROGRAM

```
1 % Zaman aralığı
2 - t = 0 : 0.2 : 10;
3 % x eksenini, y eksenini ve z eksenini
4 - x = t; y = sin (t); z = t.^1.5;
5 % 3 boyutlu işaret yerlerinin içi dolu stem grafik
6 stem3(x, y, z, 'fill')
7 % grid ekle
8 grid on
% eksenlere etiket verilmesi
9 xlabel ('x');
10 ylabel ('y');
11 zlabel ('z');
```



# Özel Grafikler

## pie komutu ile pasta grafik çizdirme

❏ **pie ( x )**

↗ pasta grafikteki her bir dilime ait yüzdeyi içeren matris



### Komut penceresi

*% pasta grafikteki dilimlerin değerlerini tanımla*

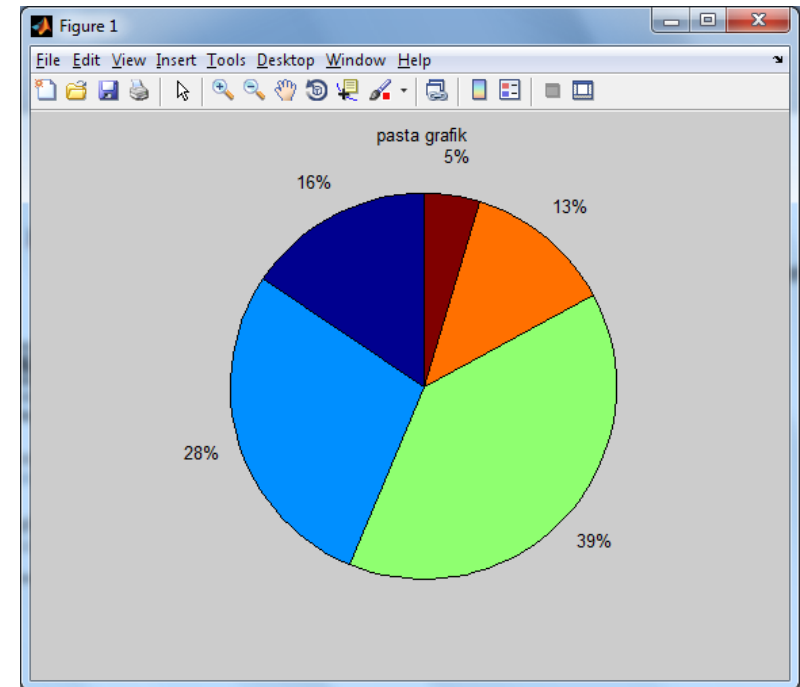
```
>> deger = [ 10 18 25 8 3 ];
```

*% pasta grafiği çizdir*

```
>> pie (deger)
```

*% grafiğin başlığı*

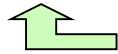
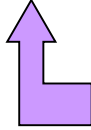
```
>> title( 'pasta grafik')
```



# Özel Grafikler

## pie3 komutu ile 3 boyutlu pasta grafik çizdirme

❏ **pie3 ( x , explode )**



x ile aynı boyutlu vektör. Dilimler arasındaki boşluğu gösterir.  
pasta grafikteki dilimlere ait değerleri içeren matris



### Komut penceresi

*% pasta grafikteki dilimlerin değerlerini tanımla*

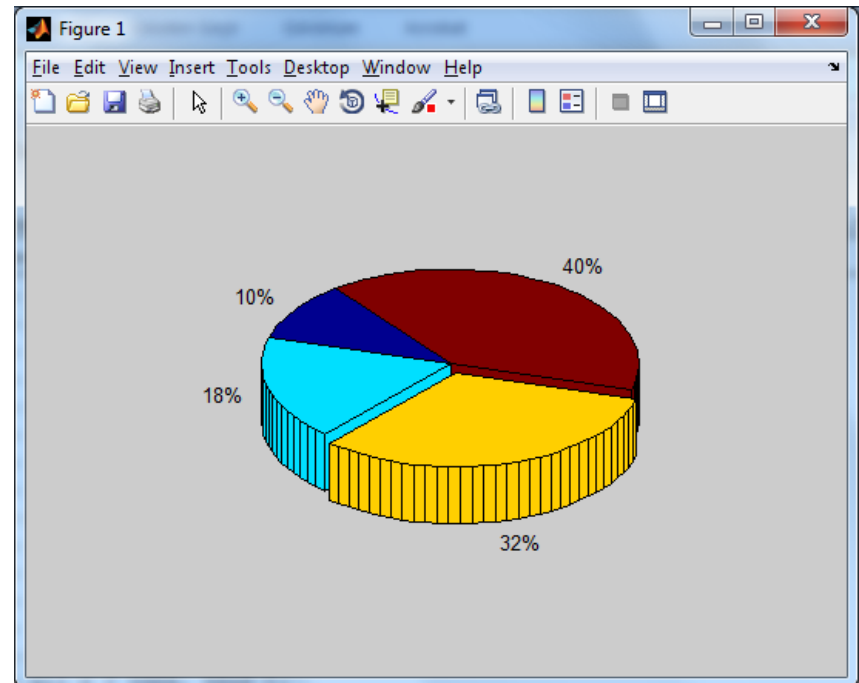
```
>> x = [ 5 9 16 20 ];
```

*% explode vektörü tanımla*

```
>> explode = [ 0 0 1 0 ];
```

*% 3 boyutlu pasta grafik*

```
>> pie3( x, explode )
```






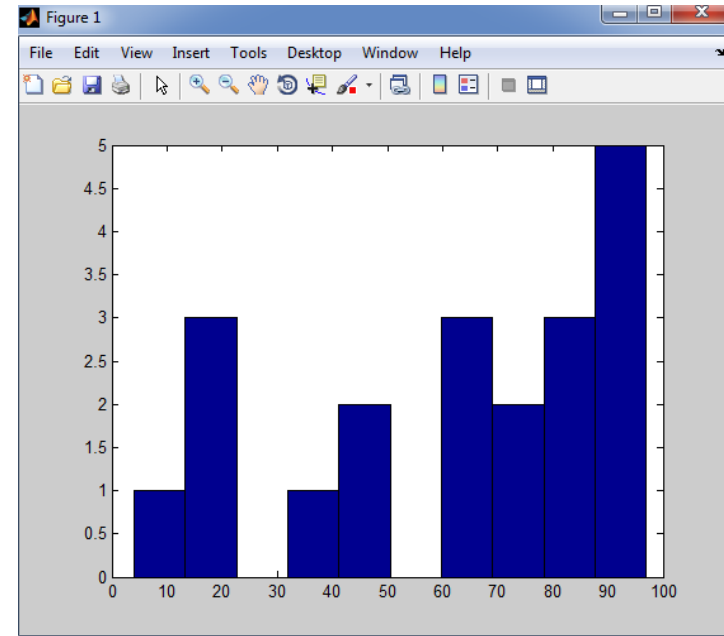
# Özel Grafikler

## hist komutu ile istatistiksel grafik çizdirme

- ❑ Veri bloğunun istatistiksel dağılımını gösteren istatistiksel ölçüler ile ilişkili özel bir grafik çizer.
- ❑ Verilerin sıklık (frekans) değerleri hesaplanır ve histogram grafikleri çizilir.
- ❑ **hist ( x )**  
↑ histogramı çizilecek veri grubu (standart olarak 10 çubukta gösterir)



```
% 1-200 arasında rasgele sayılardan oluşan  
% 1x20 boyutunda bir dizi oluştur  
>> x=round ( rand (1, 20)*100);  
  
% histogram grafik çizdir  
>> hist(x)
```



# Özel Grafikler

## polar komutu ile kutupsal koordinatlı grafik çizdirme

- ❑ Kompleks sayılar, sayı büyüklüğü ve açı ile kutupsal koordinat sisteminde gösterilebilir.
- ❑ **polar ( theta , rho )**

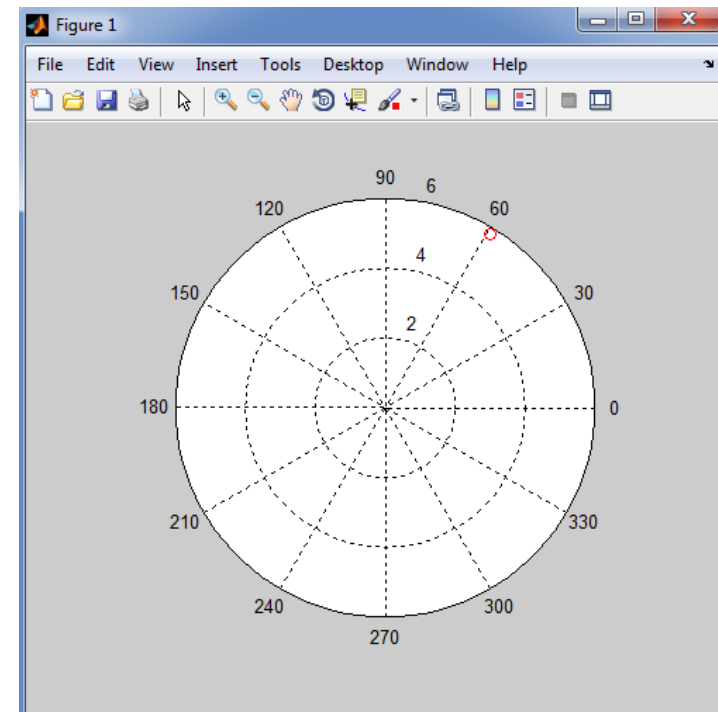
↑ kutupsal koordinatın büyüklük değeri  
↑ kutupsal koordinatın açı değeri

```
% Kompleks sayı tanımlama
>> z = 3 + 5i;

% Kompleks sayının açısını hesapla
>> theta=angle(z);

% Kompleks sayının büyüklüğünü hesapla
>> r=abs(z);

% Kutupsal koordinatlarda göster
>> polar( theta, r, 'or')
```



# Özel Grafikler

## polar komutu ile kutupsal koordinatlı grafik çizdirme

❑ **Örnek:**  $\sin 2\theta$  nın grafiğini kutupsal olarak çizdiriniz?



*% açı tanımla*

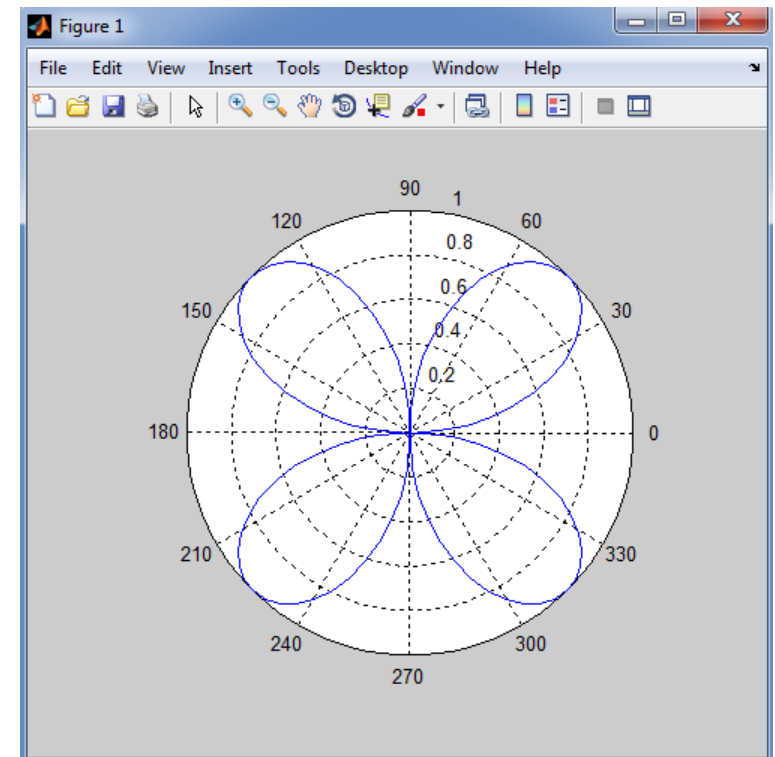
```
>> theta=linspace(0,2*pi);
```

*% büyüklük tanımla*

```
>> r=sin(2*theta);
```

*% Kutupsal koordinatlarda göster*

```
>> polar( theta, r )
```



# Grafiklere sembol ekleme

<code>\alpha</code>	$\alpha$	<code>\upsilon</code>	$\upsilon$	<code>\sim</code>	$\sim$
<code>\beta</code>	$\beta$	<code>\phi</code>	$\phi$	<code>\leq</code>	$\leq$
<code>\gamma</code>	$\gamma$	<code>\chi</code>	$\chi$	<code>\infty</code>	$\infty$
<code>\delta</code>	$\delta$	<code>\psi</code>	$\psi$	<code>\clubsuit</code>	$\clubsuit$
<code>\epsilon</code>	$\epsilon$	<code>\omega</code>	$\omega$	<code>\diamondsuit</code>	$\diamondsuit$
<code>\zeta</code>	$\zeta$	<code>\Gamma</code>	$\Gamma$	<code>\heartsuit</code>	$\heartsuit$
<code>\eta</code>	$\eta$	<code>\Delta</code>	$\Delta$	<code>\spadesuit</code>	$\spadesuit$
<code>\theta</code>	$\theta$	<code>\Theta</code>	$\Theta$	<code>\leftrightarrow</code>	$\leftrightarrow$
<code>\vartheta</code>	$\vartheta$	<code>\Lambda</code>	$\Lambda$	<code>\leftarrow</code>	$\leftarrow$
<code>\iota</code>	$\iota$	<code>\Xi</code>	$\Xi$	<code>\uparrow</code>	$\uparrow$
<code>\kappa</code>	$\kappa$	<code>\Pi</code>	$\Pi$	<code>\rightarrow</code>	$\rightarrow$
<code>\lambda</code>	$\lambda$	<code>\Sigma</code>	$\Sigma$	<code>\downarrow</code>	$\downarrow$
<code>\mu</code>	$\mu$	<code>\Upsilon</code>	$\Upsilon$	<code>\circ</code>	$\circ$
<code>\nu</code>	$\nu$	<code>\Phi</code>	$\Phi$	<code>\pm</code>	$\pm$
<code>\xi</code>	$\xi$	<code>\Psi</code>	$\Psi$	<code>\geq</code>	$\geq$

<code>\pi</code>	$\pi$	<code>\Omega</code>	$\Omega$	<code>\propto</code>	$\propto$
<code>\rho</code>	$\rho$	<code>\forall</code>	$\forall$	<code>\partial</code>	$\partial$
<code>\sigma</code>	$\sigma$	<code>\exists</code>	$\exists$	<code>\bullet</code>	$\bullet$
<code>\varsigma</code>	$\varsigma$	<code>\ni</code>	$\ni$	<code>\div</code>	$\div$
<code>\tau</code>	$\tau$	<code>\cong</code>	$\cong$	<code>\neq</code>	$\neq$
<code>\equiv</code>	$\equiv$	<code>\approx</code>	$\approx$	<code>\aleph</code>	$\aleph$
<code>\Im</code>	$\Im$	<code>\Re</code>	$\Re$	<code>\wp</code>	$\wp$
<code>\otimes</code>	$\otimes$	<code>\oplus</code>	$\oplus$	<code>\oslash</code>	$\oslash$
<code>\cap</code>	$\cap$	<code>\cup</code>	$\cup$	<code>\supseteq</code>	$\supseteq$
<code>\supset</code>	$\supset$	<code>\subseteq</code>	$\subseteq$	<code>\subset</code>	$\subset$
<code>\int</code>	$\int$	<code>\in</code>	$\in$	<code>\o</code>	$\o$
<code>\rfloor</code>	$\rfloor$	<code>\lceil</code>	$\lceil$	<code>\nabla</code>	$\nabla$
<code>\lfloor</code>	$\lfloor$	<code>\cdot</code>	$\cdot$	<code>\ldots</code>	$\ldots$
<code>\perp</code>	$\perp$	<code>\neg</code>	$\neg$	<code>\prime</code>	$\prime$
<code>\wedge</code>	$\wedge$	<code>\times</code>	$\times$	<code>\emptyset</code>	$\emptyset$
<code>\rceil</code>	$\rceil$	<code>\surd</code>	$\surd$	<code>\mid</code>	$\mid$
<code>\vee</code>	$\vee$	<code>\varpi</code>	$\varpi$	<code>\copyright</code>	$\copyright$
<code>\langle</code>	$\langle$	<code>\rangle</code>	$\rangle$		