

## **ICINDEKİLER**

- BÖLÜM 1** : MATLAB Kullanımı ve Matris İşlemleri
- BÖLÜM 2** : Grafik ve Egri Çizimleri
- BÖLÜM 3** : MATLAB ile Programlama
- BÖLÜM 4** : Kontrol Sistemlerinde Zaman-Frekans Analizi
- EK (Tablo)** : MATLAB Komutları ve Matris Fonksiyonları Tablosu

**NOT:** “ MATLAB Kullanım Kılavuzu ” nda ilk 3 bölüm genel kullanımla alakalı bilgiler içermektedir. Son bölüm ise “ Bilgisayar Destekli Kontrol Sistemleri ” dersi içерigine göre hazırlanmıştır.

## **BÖLÜM 1 : MATLAB KULLANIMI ve MATRIS ISLEMLERI**

“Matlab” programı ( MATrix LAboratory 'nin ilk üç harfleri alınarak isimlendirilmiştir.) mühendislik uygulamalarının, hesaplamalarının ve simülasyonlarının çogunun gerçekleştirildiği matris ve matematik tabanlı kompleks bir programdır. Her türlü grafiksel sonuçlar istenilen tarzda alınabildiği için kullanım alanı çok genistir. Ayrıca MATLAB versiyonlarından en az 6.0 ve üzeri olanlarının kullanılması güncelik açısından daha yararlı olacaktır.

Bu bölümde programı kullanmaya baslamak için giriş komutları, matematiksel fonksiyonlar ve matris operatörleri anlatılacaktır. Ayrıca kılavuzun en son kısmında da en çok kullanılan matris komutları ve fonksiyonları tablo halinde verilmistir.

`help 'fonksiyon ismi'`

komutu yazıldığında yardım istenilen fonksiyon hakkında detaylı bilgiye ulaşılabilirmektedir.

`help help`

yazıldığında ise on-line olarak yardım kılavuzunun nasıl kullanılacağı hakkında bilgilere ulaşılabilirmektedir.

### **-- Matris Operatörleri :**

Asagida verilen simgeler matris islemlerinde kullanılmaktadır:

- + Toplama
- Çıkarma
- \*
- Çarpma
- ^ Kuvvet alma
- ' Konjüge transpozunu alma

### **-- Mantik ve Iliski Operatörleri :**

- |    |            |   |       |
|----|------------|---|-------|
| <  | Küçük      | & | Ve    |
| <= | Küçük esit |   | Veya  |
| >  | Büyük      | ~ | Degil |
| ~= | Esit degil |   |       |

-- Baslangic olarak komut satirina :

date

yazilrsa program tarafindan geçerli olan tarih alinacaktır. Yani :

ans=  
30-Oct-2002

-- MATLAB bir islemin sonucunu ans= .... seklinde gösterir. ( ans = answer = cevap)

-- MATLAB programindan çıkmak için ise exit veya quit yazmak yeterli olacaktır.

-- En son yazilan komutların hepsine üst ve alt yön tuslarına dokunarak kolay bir sekilde ulasılabilir.

-- En son tanımlanan herhangi bir 'x' değeri için yapılan işlemlerden sonra bu 'x' değeri komut satirina yazılıp enter tusuna basılırsa daha önce neye karşılık olarak tanımladığı ekrana yazılacaktır.

--  $nx1$  veya  $1xn$  boyutunda vektör tanımlamak için :

$x=[1 2 3 -4 -5]$  veya

$X=[1,2,3,-4,-5]$  yazılmalıdır.

Yukarıdaki iki yazım biçiminden okuma kolaylığı olması için ilk yazılan tip kullanılacaktır.

-- Tanimlanan bu satır vektörünü sütun vektörüne dönüştürmek için :

$y=x'$  yazılırsa ekranada görülen değer aşağıdaki gibi olacaktır:

y=  
1  
2  
3  
-4  
-5

-- Matris tanimlamak için asagidaki A matrisi verilmis olsun :

$$A = \begin{matrix} 1.2 & 10 & 15 \\ 3 & 5.5 & 2 \\ 4 & 6.8 & 7 \end{matrix}$$

Bu matrisi MATLAB'e tanitmak için su sekilde yazilmalidir :

$$A = [1.2 \ 10 \ 15 ; 3 \ 5.5 \ 2 ; 4 \ 6.8 \ 7]$$

Yani her satirin sonunun neresi oldugunu konulan noktali virgül isareti temsil etmektedir.

-- Örnek olarak asagidaki B matrisini tanitmak için :

$$B = \begin{matrix} 1 & e^{-0.02} \\ \sqrt{2} & 3 \end{matrix}$$

B = [1 exp(-0.02); sqrt(2) 3] seklinde yazilmalidir.

Ekranda ise su sekilde gözükecektir:

$$B = \begin{matrix} 1.0000 & 0.9802 \\ 1.4142 & 3.0000 \end{matrix}$$

-- Apostrofi isareti (' ) matrisin konjuge transpozesinin alınmasına yarar. Eger matris reel bir matris ise basit olarak transpoze alım islemi olarak da tanimlanabilir..

Yeni bir A matrisi tanimlayalim :

$$A = [1 \ 2 \ 3 ; 4 \ 5 \ 6 ; 7 \ 8 \ 9]$$

Ekranda görülecek matris su sekilde olacaktir :

$$A = \begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{matrix}$$

Bu matrisin transpozesini almak için :

$C = A'$  yazılırsa ekranda görülecek transpoze değeri :

$C =$

$$\begin{matrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{matrix}$$

-- Kompleks sayıların girilmesi için ise  $i^2=-1$  denkleminin kökü  $i$  veya  $j$  olarak tanımlanır.

Örnek olarak  $1+j\sqrt{3}$  değerini tanıtmak için :

$X = 1+\sqrt{3}*i$  veya

$X = 1+\sqrt{3}*j$  olarak yazılmalıdır.

Bu kompleks sayı üstel formatta da yazılabilir :

$1+j\sqrt{3} = 2 \exp[(?/3)*j]$

Bu durumda komut satırına aşağıdaki ifade yazılacaktır :

$x = 2 \exp[\pi/3]*j$

$i$  ve  $j$  daha önceden değişken olarak kullanılmışsa tanımlama için  $ii$  ve  $jj$  kullanılacaktır. Yani :

$ii = \sqrt{-1}$

$jj = \sqrt{-1}$

Dolayısıyla aşağıdaki yazım da mümkün olmaktadır :

$X = 1+\sqrt{3}*ii$

$X = 1+\sqrt{3}*jj$

-- Kompleks matris tanimlamak için asagidaki X matrisi verilmis olsun:

$$X = \begin{matrix} 1 & j \\ -j5 & 2 \end{matrix}$$

Komut satirina ise su sekilde girilecektir :

$$X = [1 \ j ; -j5 \ 2]$$

Bu durumda ekranda görülecek deger :

$$X = \begin{matrix} 1.0000+0 & 0+1.0000i \\ 0-5.0000i & 2.0000+0 \end{matrix}$$

Y = X' komutu yazilrsa :

$$Y = \begin{matrix} 1.0000+0 & 0+5.0000i \\ 0-1.0000i & 2.0000+0 \end{matrix}$$

iletisi ekranda okunacaktır.

Daha önce de belirtildigi gibi yukarıdaki islem konjüge transpoze olarak algilanmaktadır. Eger sadece transpoze alinacaksa (konjügesiz) komut su sekilde yazilmalidir :

$$Y.' \quad \text{veya} \quad \text{conj}(Y')$$

Bu durumda ekranda gözükecek degerler :

$$\text{ans} = \begin{matrix} 1.0000+0 & 0-1.0000i \\ 0+5.0000i & 2.0000+0 \end{matrix}$$

-- Toplama ve çıkarma islemlerinin yapılması için asagidaki M ve N matrisleri verilmis olsun :

$$M = \begin{matrix} 2 & 3 \\ 4 & 5 \\ 6 & 7 \end{matrix} \quad N = \begin{matrix} 1 & 0 \\ 2 & 3 \\ 0 & 4 \end{matrix}$$

Bu değerleri ekrana girmek için:

$$M = [2 \ 3 ; 4 \ 5 ; 6 \ 7]$$

$$M = \begin{matrix} 2 & 3 \\ 4 & 5 \\ 6 & 7 \end{matrix}$$

$$N = [1 \ 0 ; 2 \ 3 ; 0 \ 4]$$

$$M = \begin{matrix} 1 & 0 \\ 2 & 3 \\ 0 & 4 \end{matrix}$$

Toplama işlemi için:

$$C = M+N$$

$$C = \begin{matrix} 3 & 3 \\ 6 & 8 \\ 6 & 11 \end{matrix}$$

Eğer x vektörü su aşağıdaki gibi verilirse :

$$X = \begin{matrix} 5 \\ 4 \\ 6 \end{matrix}$$

Bu vektörü su şekilde tanıtmak gereklidir :

$$X = [5 ; 4 ; 6]$$

Örneğin  $T = X-1$  gibi işlemi gerçekleştirmek için :

$$T = X-1$$
$$T = \begin{matrix} 4 \\ 3 \\ 5 \end{matrix}$$

-- Matris çarpımı daha önce de belirtildiği gibi \* çarpma operatörüyle yapılmaktadır. Asağıdaki örnek incelenirse çarpanın da tanımı gereği çarpılan matrislerin boyutlarının uyusması gerekmektedir. Aksi takdirde çarpma işlemi yapılmayacak ve hata mesajı verilecektir.

$$x = [1 ; 2 ; 3]; \quad y = [4 ; 5 ; 6]; \quad A = [1 \ 1 \ 2 ; 3 \ 4 \ 0 ; 1 \ 2 \ 5]$$

$$x' * y$$

$$\begin{aligned} \text{ans} = \\ 32 \end{aligned}$$

$$x * y'$$

$$\begin{aligned} \text{ans} = \\ \begin{matrix} 4 & 5 & 6 \\ 8 & 10 & 12 \\ 12 & 15 & 18 \end{matrix} \end{aligned}$$

$$b = A * x$$

$$\begin{aligned} b = \\ 9 \\ 11 \\ 20 \end{aligned}$$

Bunların dışında matris bir skaler değerle de çarpılabilir :

$$5 * A$$

$$\begin{aligned} \text{ans} = \\ \begin{matrix} 5 & 5 & 10 \\ 15 & 20 & 0 \\ 5 & 10 & 25 \end{matrix} \end{aligned}$$

-- Matris üssü (  $\text{expm}(A)$  )  $n \times n$  matrise uygulanır. Matematiksel tanımı ise şu şekildedir:

$$\text{expm}(A) = I + A + A^2/2! + A^3/3! + \dots$$

Eğer A kompleks bir matris ise  $\text{abs}(A)$  değeri de kompleks modül değerler üzerinden hesaplanacaktır. Yine matematiksel ifadesine bakacak olursak :

$$\text{abs}(A) = \sqrt{\text{real}(A)^2 + \text{imag}(A)^2}$$

`angle(A)` ise faz açılarını radyan cinsinden A kompleks matrisi için hesaplamaktadır. Burada tanım değerleri  $-\pi$  ve  $\pi$  arasında kabul edilmektedir.

Sonuç olarak verilen bir K matrisi için aşağıdaki uygulama incelenebilir :

```
A = [2+2*i    1+3*i ; 4+5*i    6-i];  
abs(A)
```

```
ans =  
2.8284    3.1623  
6.4031    6.0828
```

```
angle(A)
```

```
ans =  
0.7854    1.2490  
0.8961    -0.1651
```

-- Kompleks bir sayının modülü ve faz açısını bulmak için :

```
z = x + iy = re^{iT}
```

```
r = abs(z)  
theta = angle(z)  
z = r*exp(i*theta)
```

-- Bir vektörün elemanlarının teker teker karesinin alınması işlemi şu şekilde yapılmaktadır :

```
x = [1  2  3];  
x.^2
```

```
ans =  
1  4  9
```

Eğer kompleks sayılar mevcut ise :

```
y = [2+5*i    3+4*i    1-i]  
y.^2
```

```
ans =  
-21.0000+20.0000i   -7.0000+24.0000i   0-2.0000i
```

2x2 bir kare matris olursa yine aynı şekilde :

$$A = [1 \quad 2; 3 \quad 4];$$
$$A.^2$$

$$\text{ans} =$$
$$\begin{matrix} 1 & 4 \\ 9 & 16 \end{matrix}$$

-- Eleman elemana çarpma işlemi için çarpma operatörünün önüne bir nokta işaretü (.) konmaktadır :

$$x = [1 \quad 2 \quad 3], \quad y = [4 \quad 5 \quad 6]$$
$$z = [x.y]$$
$$z = [4 \quad 10 \quad 18]$$

Bir örnek daha verilirse :

$$A = \begin{matrix} 1 & 2 & 3 \\ 1 & 9 & 8 \end{matrix} \quad B = \begin{matrix} 4 & 5 & 6 \\ 7 & 6 & 5 \end{matrix}$$

$$C = A.*B$$

$$C =$$
$$\begin{matrix} 4 & 10 & 18 \\ 7 & 54 & 40 \end{matrix}$$

-- Bir matrisin tüm elemanlarının tek tek karesini almak için :

$$A=[1 \quad 2 \quad 3 \quad 4]$$

$$A.^2$$

$$\text{ans} =$$

$$\begin{matrix} 1 & 4 & 9 & 16 \end{matrix}$$

-- Eleman elemana bölme için ise :

$$u = x./y$$

```
U =
0.2500 0.4000 0.4000
0.1429 1.5000 1.6000
```

-- Bir matrisin tersini bulmak için inv(A) komutu kullanılır :

```
A = [0 1 0;0 0 1;-6 -11 -6];
inv(A)
```

```
ans =
-1.8333    -1.0000    -0.1667
 1.0000      0          0
 0          1.0000    0
```

-- Çesitli komutlar ve durumlar tek bir sirada virgül (,) veya noktali virgül ile (;) ayırlarak yazılabilir.

-- Çikis formatini istedigimiz uzunlukta elde edebiliriz. Eger matris elemanları tamsayı ise bu durum sonuçta bir degisiklik yapmayacaktır. Bunun için asagidaki komutları kullanmak gerekmektedir :

```
format short
format long
```

```
x = [1/3    0.00002];
x
```

```
x =
0.3333    0.0000
```

```
format short; x
```

```
x =
0.3333    0.0000
```

```
format long; x
```

```
x =
0.3333333333333333            0.0000200000000000
```

-- 1'den 5'e kadar sayıları 0.5'lik aralıklarla yazdırırmak istersek iki nokta'yı (:) kullanmak yeterli olacaktır :

```
t =  
1 2 3 4 5
```

$t = 1:0.5:3$

```
t =  
1.0000 1.50000 2.0000 2.50000 3.0000
```

Düzgün azalan biçimde yazdırırsak :

$t = 5:-1:2$

$t = 5 4 3 2$

-- Bir matrisin i. satirini veya j. sütununu görüntülemek için aşağıda tanımlanan A matrisini komutlarıyla inceleyelim :

Aşağıdaki A matrisinin 2. satırı görüntülemek için :  $A(i, :)$

```
A = [0 1 0; 0 0 1; -6 -11 -6]  
A(2, :)
```

```
ans =  
0 0 1
```

A matrisinin 3. sütununu görüntülemek için :

$A(:, 3)$

```
ans =  
0  
1  
-6
```

-- Bir matrisin (i,j) ninci elemanını bulmak için :

$k = A(3,3)$

$k = -6$

-- Bir matrisin boyutlarını öğrenmek için size(A) komutu, rankini bulmak için rank(A) kullanılır.

```
A=
```

2	3	2
5	4	1
2	6	8

```
size(A)
```

```
ans =
```

3	3
---	---

```
rank(A)
```

```
ans =
```

3
---

-- Bir matrisin determinantını bulmak için det(A) komutu kullanılır.

```
A=
```

2	3	2
5	4	1
2	6	8

```
det(A)
```

```
ans =
```

-18
-----

-- Bir matrisin normunu bulmak için ise norm(x) yazmak gerekmektedir. Matematiksel norm ifadesini verecek olursak :

```
norm(x) = sum(abs(x).^2)^0.5
```

```
x = [2 3 6]
```

```
norm(x)
```

```
ans =
```

7
---

-- Bir matrisin özdeğerlerini bulmak için eig(A) komutu kullanılır :

```
A = [0 1 ; -1 0]
eig(A)
```

```
ans =
0+1.0000i
0 -1.0000i
```

-- Öz vektörleri bulmak da tek satırlik bir işlem gerektirmektedir. Aslında özvektörleri bulmak için verilen  $[X,D] = \text{eig}(A)$  komutu aynı zamanda öz değerleri de bulduğu için her iki bilgiye aynı anda ulaşma imkani olmaktadır :

```
A = [0 1 0 ; 0 0 1 ; -6 -11 -6]
[X,D] = eig(A)
```

```
X =
-0.5774    0.2182    -0.1048
 0.5774   -0.4364    0.3145
-0.5774    0.8729   -0.9435
```

```
D =
-1.0000      0        0
 0       -2.0000      0
 0         0       -3.0000
```

Burada X sonuç matrisinin her bir sütunu verilen A matrisinin bir öz değerini göstermektedir.

D sonuç matrisinin diyagonalindeki (kösegenindeki) elemanların her biri de verilen A matrisinin özdeğerlerini göstermektedir.

Verilen es boyutlu farklı iki A ve B gibi matrisin genelleştirilmiş öz değerlerini ve öz vektörlerini bulmak için ise  $[X,D] = \text{eig}(A,B)$  komutu yazılmalıdır.

-- Karakteristik denklemi bulmak için poly(A) komutu kullanılır.

```
A = [0 1 0 ; 0 0 1 ; -6 -11 -6]
```

```
p = poly(A)
```

```
p =
1.0000      6.0000     11.0000     6.0000
```

Burada görülen sonuç katsayıları karakteristik denklemin katsayılarıdır. Yani :

$$s^3 + 6s^2 + 11s + 6 = 0$$

-- Bir polinomun köklerini bulmak için roots(a) komutu yazılmalıdır. Yukarıdaki karakteristik denklemin köklerini bulmak istersek :

$$r = \text{roots}(p)$$

$$r =$$

$$\begin{aligned} & -3.0000 \\ & -2.0000 \\ & -1.0000 \end{aligned}$$

-- Polinomların çarpımı için conv(a,b) komutu kullanılır.

$$\begin{aligned} a(s) &= s^2 - 20.6 \\ b(s) &= s^2 + 19.6s + 151.2 \end{aligned}$$

a(s) ve b(s) polinomlarını çarpmak için :

$$\begin{aligned} a &= [1 \ 0 \ -0.26]; \quad b = [1 \ 1.96 \ 151.2] \\ c &= \text{conv}(a,b) \end{aligned}$$

$$c =$$

$$\begin{array}{cccccc} 1.0e+003 & & & & & \\ 0.0010 & 0.0196 & 0.1306 & -0.4038 & -3.1147 & \end{array}$$

Dolayısıyla çarpım sonucu şu şekilde yazılabilir :

$$c(s) = s^4 + 19.6s^3 + 130.6s^2 - 403.8s - 3114.7$$

-- Bir polinomda herhangi bir tamsayı değerini hesaplatmak için polyval(c) komutu kullanılır :

$$p(s) = 3s^2 + 2s + 1$$

$$\begin{aligned} p &= [3 \ 2 \ 1]; \\ \text{polyval}(p, 5) & \end{aligned}$$

$$\text{ans} =$$

$$86$$

-- 1 ve 0 sayılarının istenilen matrisel boyutta çabuk olarak üretilebilmesi için ones(m,n) ve zeros(m,n) komutları kullanılabilir :

```
ones(2,2)
```

```
ans =
```

```
1 1  
1 1
```

```
zeros(3,3)
```

```
ans =
```

```
0 0 0  
0 0 0  
0 0 0
```

-- Birim matris de eye(n) komutuyla istenilen boyutta olusturulabilir :

```
eye(5)
```

```
ans =
```

```
1 0 0 0 0  
0 1 0 0 0  
0 0 1 0 0  
0 0 0 1 0  
0 0 0 0 1
```

-- Bir matrisin kösegenindeki elemanları listelemek için diag(A) komutu kullanılır :

```
A = [1 2 3 ; 4 5 6 ; 7 8 9] ;
```

```
diag(A)
```

```
ans =
```

```
1  
5  
9
```

Kösegenin elemanları haricindeki matris bileşenleri 0 olarak göstermek istersek :

```
diag(diag(A))
```

```
ans =
```

```
1 0 0  
0 5 0  
0 0 9
```

Kösegen matrisi olusturmayla alakali asagidaki diger örnekler de incelenebilir :

```
diag(1:5)
```

```
ans =
```

1	0	0	0	0
0	2	0	0	0
0	0	3	0	0
0	0	0	4	0
0	0	0	0	5

```
diag(0:4)
```

```
ans =
```

0	0	0	0	0
0	1	0	0	0
0	0	2	0	0
0	0	0	3	0
0	0	0	0	4

```
[diag(1:5) - diag(0:4)]
```

```
ans =
```

1	0	0	0	0
0	1	0	0	0
0	0	1	0	0
0	0	0	1	0
0	0	0	0	1

-- Bir matrisi rastgele olarak olusturmak için `rand(n)` komutu kullanılır. 0 ile 1 arasındaki sayıları alır.

```
rand(4)
```

```
ans =
```

0.3654	0.6739	0.3603	0.0493
0.1400	0.9994	0.5485	0.5711
0.5668	0.9616	0.2618	0.7009
0.8230	0.0589	0.5973	0.9623

## **ALISTIRMALAR**

**1-**  $\sin(x)$  'i ilk 10 terim kullanarak bulan bir Matlab programı yazınız. X açı degerinin derece olarak kullanıcidan alıp  $\sin(x)$ 'i bulunuz ?

$$\sin(x) = x - x^3/3! + x^5/5! - x^7/7! + x^9/9! - \dots$$

**2-** Yerden  $V_0$  hızıyla ve  $T$  açısıyla fırlatılan bir bir topun  $t = 0,1,2,\dots,10$  saniye boyunca hareket bilgilerini veren. Sema  $T$ ,  $V_x$ ,  $V_y$  bilgilerini her saniyede görüntüleyebilmelidir.

Not1: Gerekli formüller aşağıda sıralanmıştır. ( $g = 10 \text{ m/s}^2$ , Yer çekimi ivmesi)

$$V_x = V_0 \cos(T); V_y = V_0 \sin(T) - gt; V = (V_x^2 + V_y^2)^{1/2}$$

Not2: Topun tüm harketi boyunca yerden yeterince yüksekte olduğunu varsayıınız.

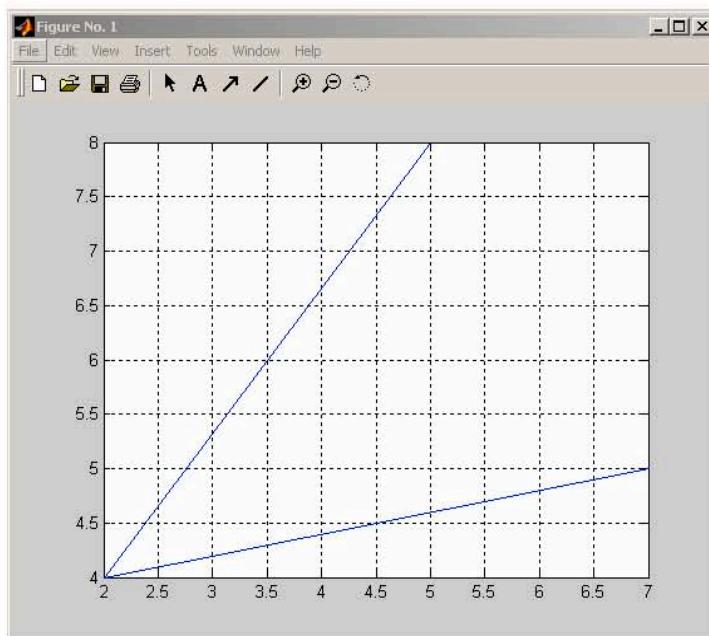
## **BÖLÜM 2 : GRAFIK ve EGRI ÇİZİMLERİ**

-- x ve y vektörleri aynı boyutta ise bu vektörleri ekrana çizdirmek için plot(x,y) komutu kullanılır.

```
A =[ 7 2 5];  
B =[ 5 4 8];  
plot(A,B);  
grid
```

Bu durumda grafik ekrana aşağıda gösterildiği gibi otomatik olarak çizilecektir :

Ayrıca plot(X,Y,'x') komutu çizilen eğriyi 'x' karakterini kullanarak çizmektedir.



-- Asagida grafik çizimiyle ilgili bazi özellikler sıralanmıştır :

**x=3:0.5:10**

Seçilen bir parametreye göre (burada x parametresi seçilmiş) çizdirilmesi planlanan egrinin sınırları yukarıdaki gibi yazılır. 3 ve 10 değerleri çizdirilmek istenen aralığı, ortadaki 0.5 değeri artı miktarını göstermektedir.

**grid**

Grafik arka yüzünün ölçekli olarak gösterilmesini sağlar.

**title('...')**

Çizilen grafige baslik yazmak için kullanılır.

**xlabel('...')**

Çizilen grafigin x-eksenine istenilen açıklamayı yazmak için kullanılır.

### **ylabel('...')**

Çizilen grafigin y-eksenine istenilen açıklamayı yazmak için kullanılır.

### **text('X,Y,'text')**

Grafik ekranı üzerine istenilen koordinatlar dahilinde herhangi bir açıklama yazmak için kullanılır.

. + \* o x

Istenildiği takdirde çizilen egrinin düz çizgi olarak değil de farklı karakterlerle çizdirilebilir. Bunlar için ise yukarıda gösterilen nokta, artı, yıldız, yuvarlak ve x karakterleri kullanılır. Bu karakterleri plot() komutu içerisinde '+' şeklinde yazmak yeterli olacaktır.

r g b w i

Cizilen egrinin rengi de yukarıda gösterilen kısaltmalarla değiştirilebilir. Burada 'r' kırmızı renk (red), 'g' yeşil renk (green), 'b' mavi renk (blue), 'w' beyaz renk (white) ve 'i' ise (invisible) olarak kısaltılmıştır.

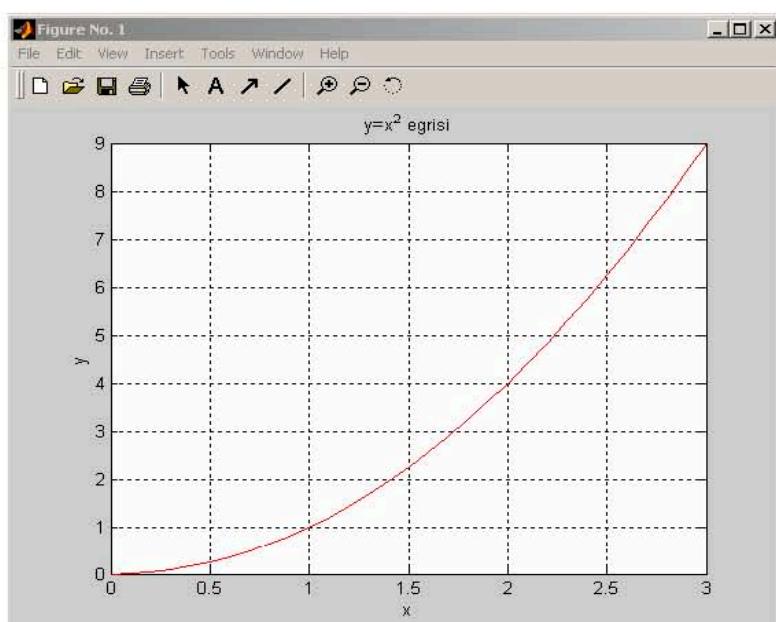
**Not :** Bu özellikler ve daha farklı görüntü özellikleri grafik ekranı üzerindeki "Insert" ve "Tools" menüleri aracılığıyla komut satırını kullanmadan da yapılmaktadır.

-- Asagidaki örnekte ise  $y = x^2$  egrisini 0 ve 3 aralığında çizdirelim :

```

x = 0:0.1:3;
y = x.^2;
plot(x,y,'r');
title('y=x2 egrisi');
xlabel('x');
grid;
ylabel('y')

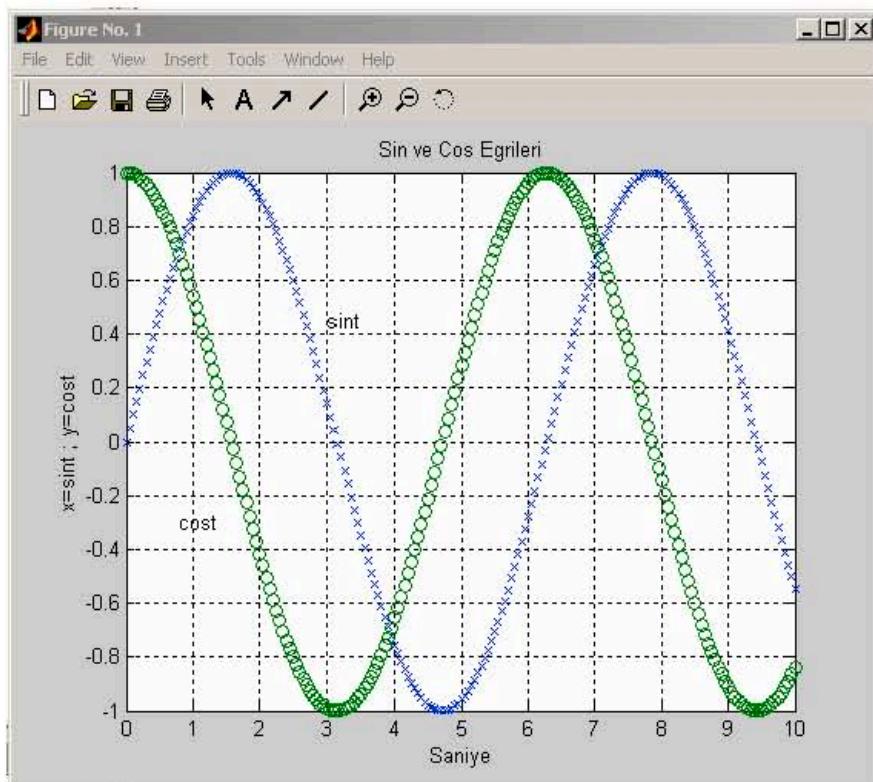
```



-- Birden fazla eğriyi tek bir grafik ekranı üzerinde görmek için çizdirilmesi istenen egriler aynı plot(..) komutu içinde yazılmalıdır.

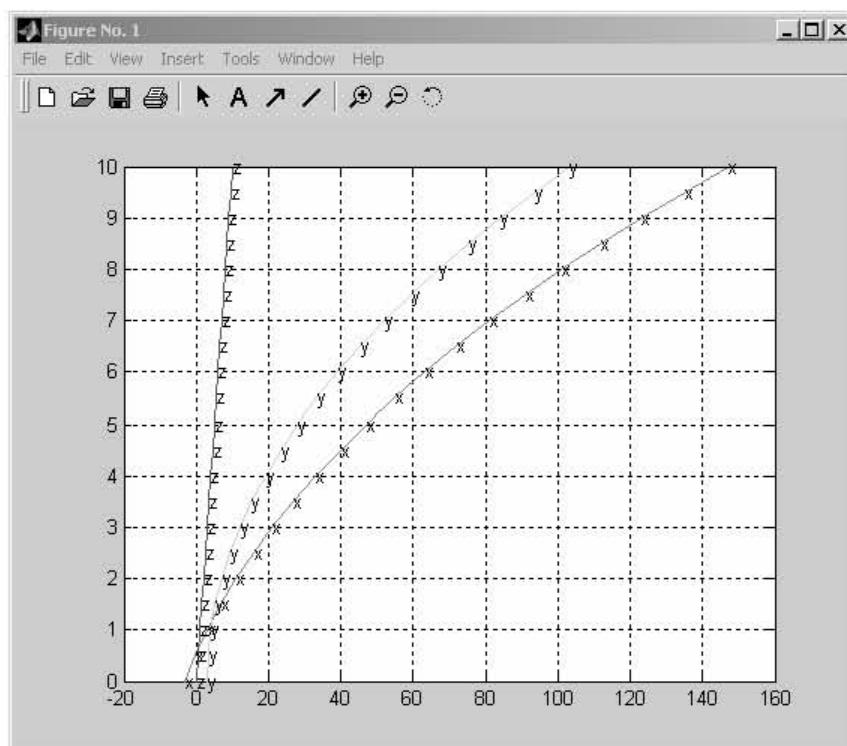
-- Birden fazla eğriyi üst üste çizme uygulaması olarak aşağıdaki örnekte  $\sin(x)$  ve  $\cos(x)$  egrileri tek bir grafik ekranı üzerinde çizdirilmistir :

```
t = 0:0.05:10;  
x = sin(t);  
y = cos(t);  
plot(t,x,'x',t,y,'o');  
grid;  
title('Sin ve Cos Egrileri');  
xlabel('Saniye');  
ylabel('x=sint ; y=cost');  
text(3,0.45,'sint');  
text(0.8,-0.3,'cost')
```



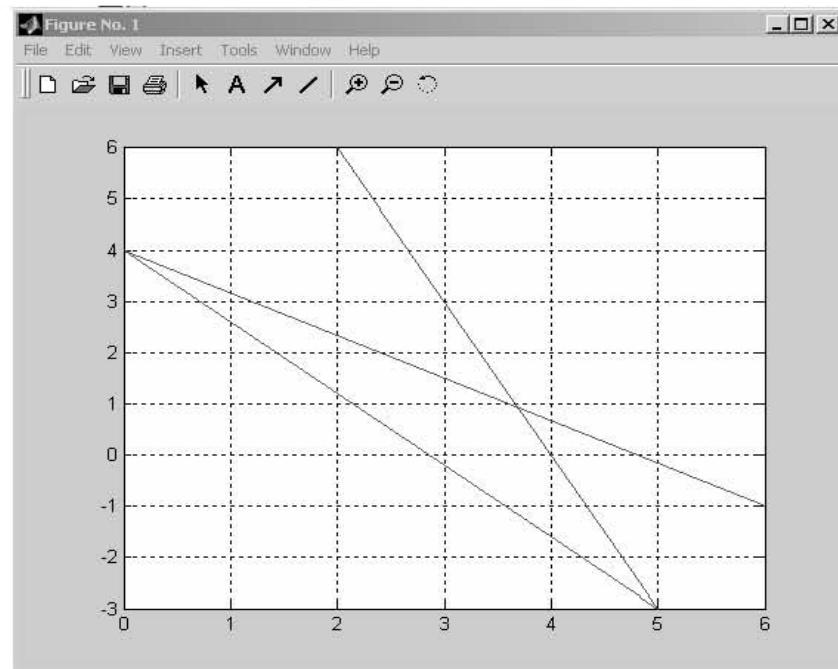
-- Asagidaki örnekte ise 3 farkli egri çizdirilmistir :

```
t=0:0.5:10;  
x=t.^2+5*t-3;  
y=t.^2+3;  
z=t;  
plot(x,t,'r',y,t,'g',z,t,'b');  
grid;  
title('3 Farkli Grafigin Cizimi');  
xlabel('Giris Degerleri');  
ylabel('Cikis Degerleri');  
text(x,t,'x');  
text(y,t,'y');  
text(z,t,'z')
```



-- Kompleks vektörlerin çiziminde plot(z) ifadesi kullanılır. Çizim işleminde ise reel ve imajiner kisimlar ayrı ayrı ikili noktalar olarak kabul edilir :

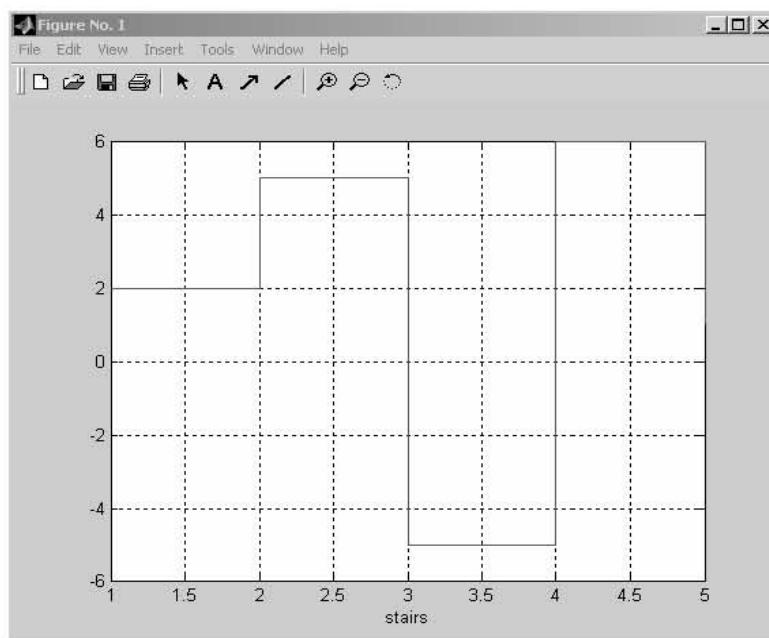
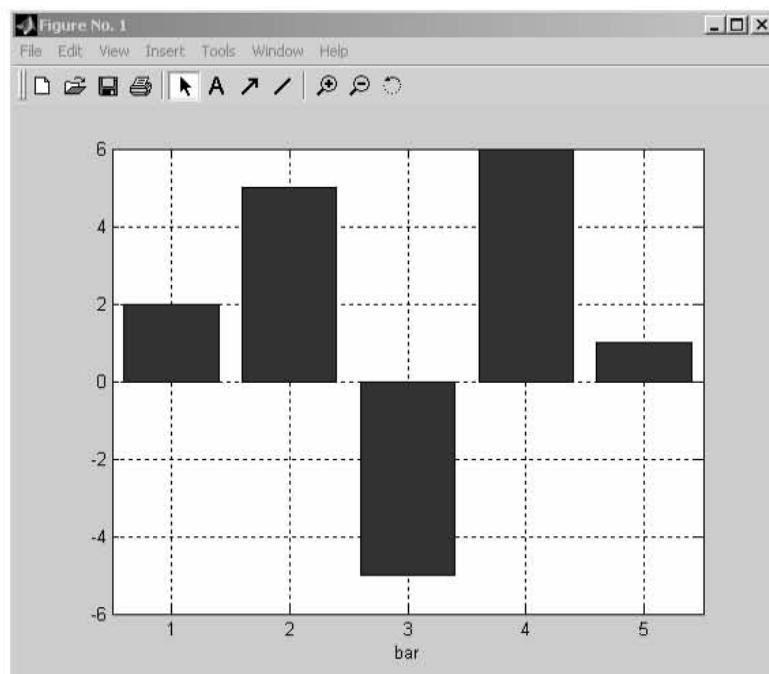
```
C=[2+6i 5-3i 4i 6-i]  
plot(C)  
grid
```



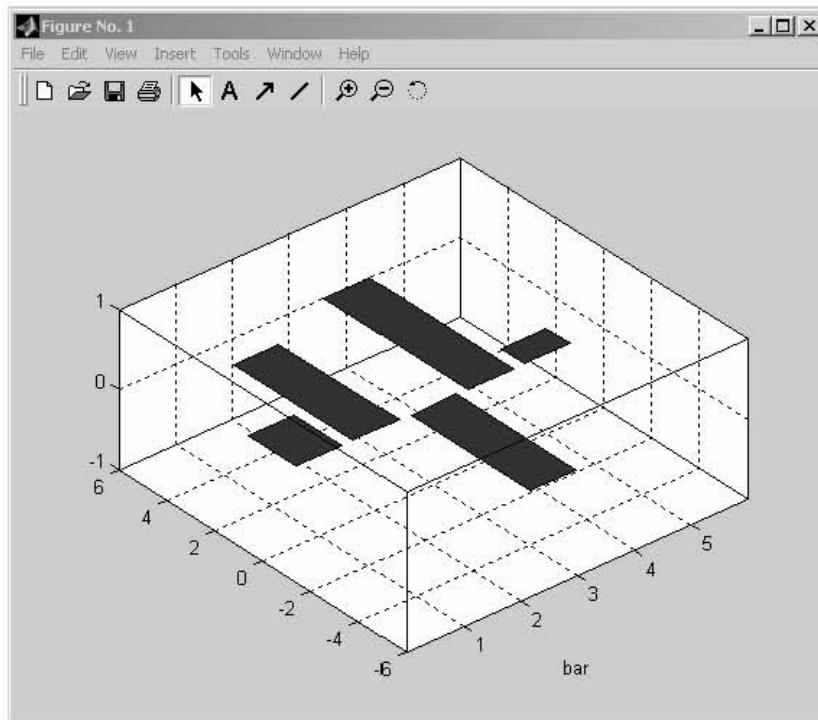
Not : loglog(X) komutu hem x eksenini hem de y ekseni ni logaritmik ölçeklendirmeyi kullanarak X'in grafigini çizdirir

-- Bir A vektörünü " bar grafiklerini " kullanarak çizdirmek için bar(A) komutu kullanılır. " Basamak " fonksiyonu şeklinde çizilecek ise stairs(A) komutu kullanılır. Her iki çizime ait örnek grafikler aşağı ayrı ayrı verilmistir :

```
A = [ 2 5 -5 6 1 ]
bar(A);
grid;
xlabel('bar');
stairs(A);
xlabel('stairs')
```



-- Ayrıca grafik ekranındaki menülerden yararlanarak çeşitli görüntü degisiklikleri yapılabilir. Örnek olarak " Tools " menüsünde " Rotate-3D " seçenekleri kullanılarak mouse yardımıyla iki üstteki " bar " grafiginin görüntüsü aşağıdaki gibi elde edilebilir.

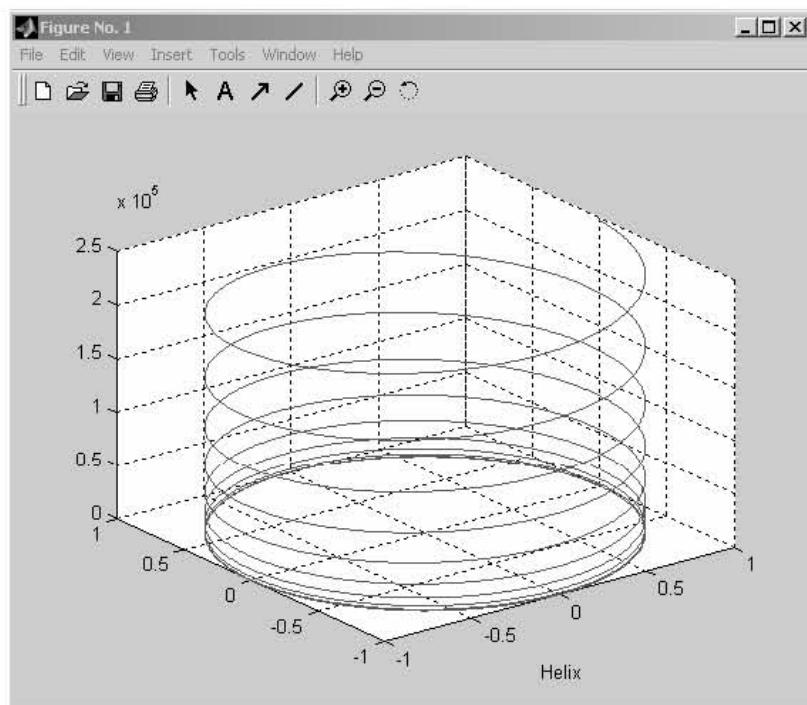


-- Grafik çiziminde grafik çizgi tipleri, işaretler ve renkler aşağıdaki tabloda sıralanmıştır :

Sembol	Renk(RGB)	Çizgi stili	Sembol	Nokta stili	
Y	sarı(110)	.	nokta	-	Çizgi
M	magenta(101)	O	yuvarlak	:	Noktalı
C	cyan(011)	X	çarpi işaret	-.	çizgili ve noktalı
R	kirmizi(100)	+	arti işaret	--	kesik çizgili
G	yesil(010)	*	yıldız		
B	mavi(001)	S	karekök		
W	beyaz(111)	D	baklava		

-- '3-D Line' (3 Boyutlu düz çizgi) çizimi için plot3(...) komutu kullanılır .Asagida heliks çizimi programı verilmistir :

```
t=0.01:0.01:20*pi;  
x=cos(t);  
y=sin(t);  
z=t.^3;  
plot3(x,y,z);  
xlabel('Helix');  
grid
```



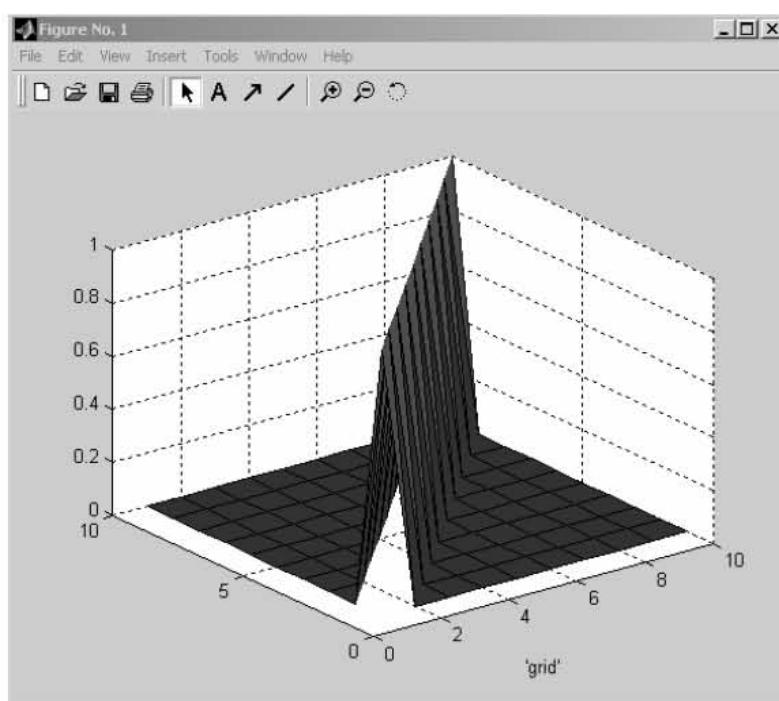
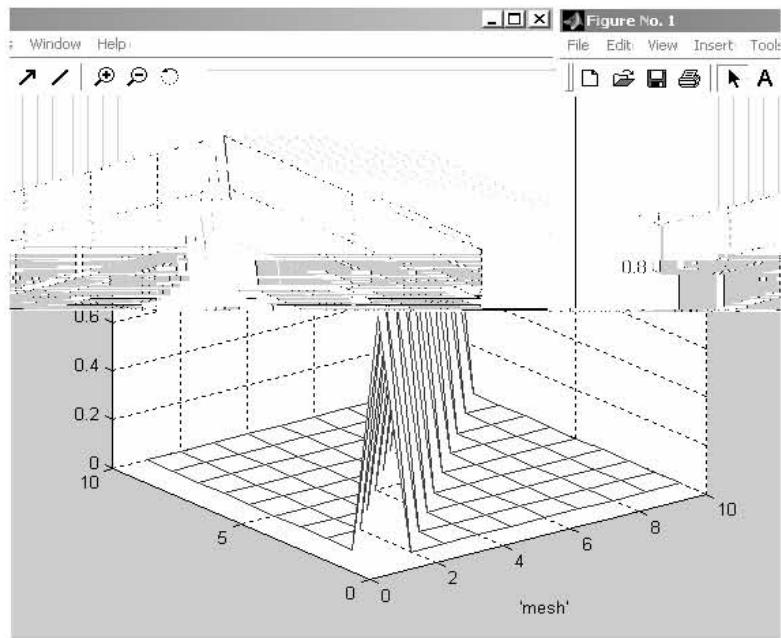
-- 3 boyutlu ag ve yüzey çizimlerinde kullanılan komutlardan biri mesh(...) komutudur. Bu komut verilen girişi z bileseni olarak algılar ve dikdörtgen x-y düzlemi üzerinde z ekseni boyunca çizim yapar. surf(...) komutu ise aynı işi yüzey olarak yapar. Asagidaki komut satırlarının çizim görüntülerini yine alt tarafında verilmistir.

```
mesh(eye(10));
```

```
grid
```

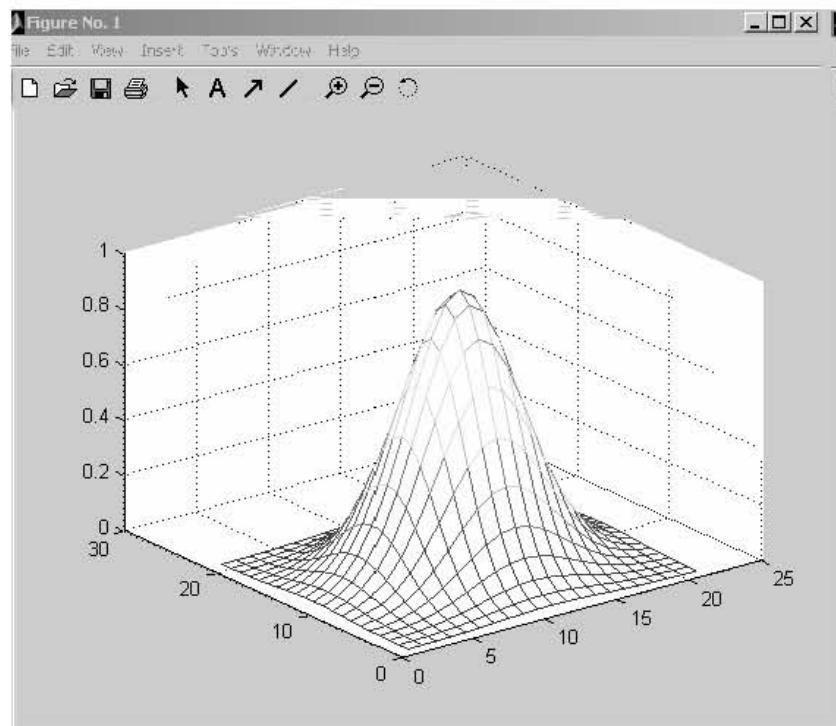
```
surf(eye(10));
```

```
grid
```



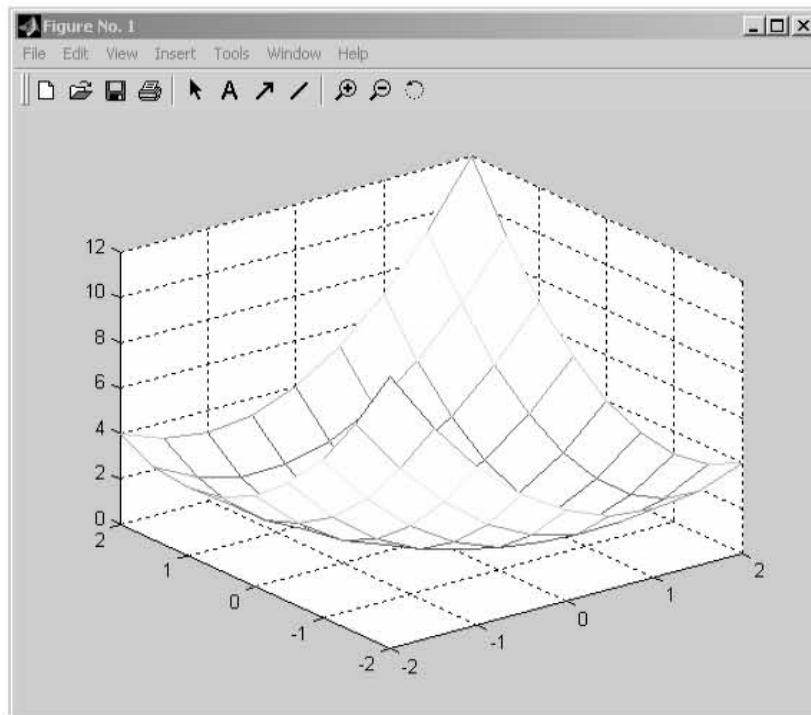
--  $z = \exp(-x^2 - y^2)$  fonksiyon yüzeyini  $[2,2] \times [-2,2]$  tanım aralığında 3 boyutlu olarak çizdirelim :

```
x=-2:0.2:2;
y=x;
[x,y]=meshgrid(x,y);
z=exp(-x.^2-y.^2);
mesh(z)
```



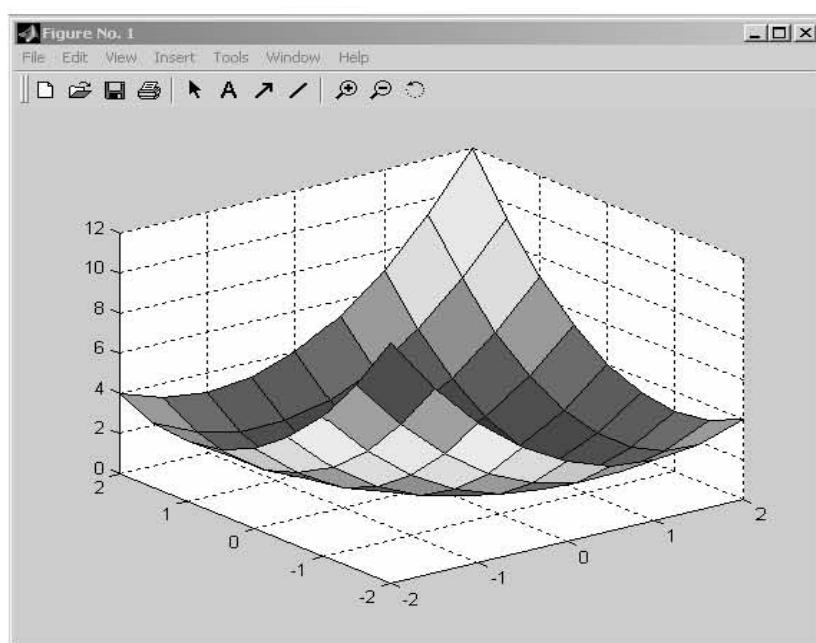
-- Örnek olarak  $z=x^2+y^2+xy$  yüzeyini  $-2 < x < 2$  ve  $-2 < y < 2$  aralığında çizdirelim :

```
[X,Y]=meshgrid(-2:0.5:2,-2:0.5:2);
Z=X.^2+Y.^2+X.*Y;
mesh(X,Y,Z)
```



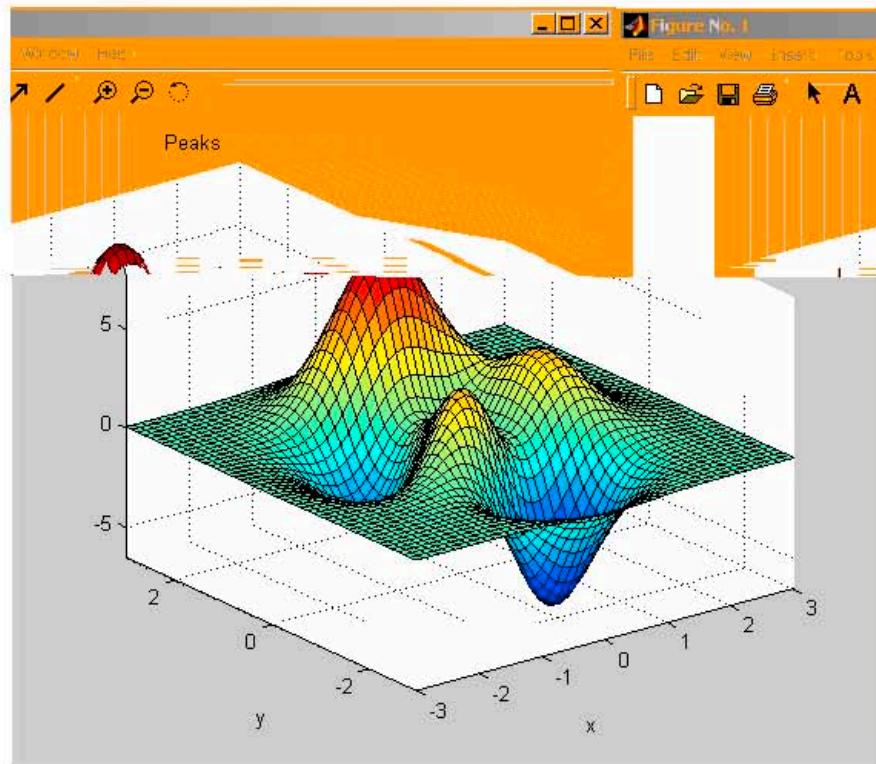
-- Yukarıdaki örnekte çizim fonksiyonu olarak  $\text{mesh}(X,Y,Z)$  yerine  $\text{surf}(X,Y,Z)$  çizim fonksiyonu kullanılırsa grafik yüzeyi aynı fakat her bir karesi farklı renklere boyanmış şekilde çizilecektir:

```
surf(X,Y,Z)
```



-- Herhangi bir yüzey grafiginde tepe ve alt tepe (minimum ve maximum) değerlerini göstererek yapılan çizimlerde peaks(...) komutu kullanılır :

```
[X,Y]=meshgrid(-3:0.125:3);  
peaks(X,Y)
```



## **ALISTIRMALAR**

**1-**  $x = t^3 - 2t + 9$ ,  $y = 6t^5 - t$ ,  $z = t^2 + 7$  egrilerini tek bir grafik ekranında çizdiriniz.

**2-**  $A = [ 5 \ 8 \ -2 \ 6 \ 4 \ 0 \ 7 ]$  giriş verilerini bar grafik ekranında çizdiriniz.

**3-**  $z = 2x^2 + y$  yüzeyini, 0.2 artım değeriyle  $x = (-2, 2)$  ve  $y = (-2, 2)$  aralığını kullanarak çizdiriniz.

**4-**  $z = e^{-2x} + 4x^3$  grafigini  $(2, 50)$  aralığında çizdiriniz.

**5-**  $x = 9\sin(t)$ ,  $y = 2\tan(3t) + \cos(t)$  grafiklerini  $(0, 10)$  aralığında 0.05 artımla çizdiriniz.

## **BÖLÜM 3 : MATLAB ile PROGRAMLAMA**

Bu bölümde MATLAB yazılımını başlangıçta kolaylıkla kullanabilmek için gerekli olan komut ve fonksiyonlar verilecektir.

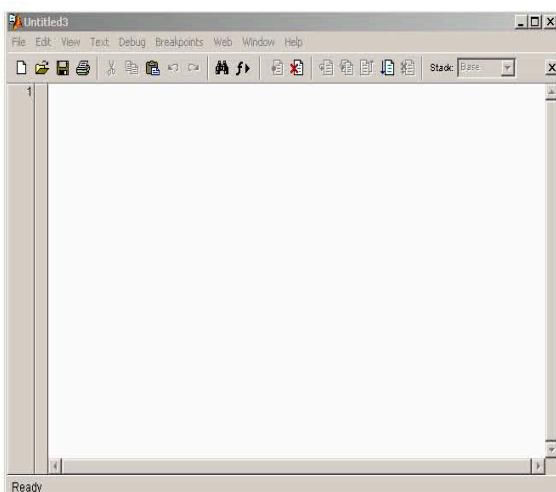
Su ana kadar MATLAB ile yaptığımız uygulamalarda, belirli bir işlemi gerçekleştirmek üzere gerekli olan komut ya da fonksiyonları komut satırından, >> sembollerinden sonra tek tek girerek icra ediyorduk. Oysa, MATLAB komut ya da fonksiyonlarından gerçekleştirmek istediğimiz bir işle ilgili özel bir grubu, bir dosyaya kaydederek, bu dosya isminin çağırılmasıyla icra ettirebiliriz.

Bir deyimler ya da komutlar grubunu içeren bu tip bir dosyaya MATLAB'de M-dosyası (M-File) adı verilir. Bir komut grubu içerdigi için tanım itibariyle bir program dosyasıdır. Dolayısıyla programlama M-dosyaları oluşturularak yapılır.

-- Komut satırına “helpwin” komutu girilirse veya MATLAB’ın Help menüsünden “Help Window” seçeneği seçilirse karsımıza gelecek yardım seçeneklerinden istenilen konu hakkında bilgi alınabilmektedir.

-- **“M-File” Oluşturma** : Programlama işlemi M-File (Program Dosyası) dosyalarında yapılır. Bu nedenle program yazarken en çok kullanılacak olan işlem M-File oluşturma işlemidir. Bunun için “File” menüsünden “New” seçilir ve daha sonra “M-File” seçeneği seçilir. Böylece yeni bir programlama ekranı elde edilir. Programlama işlemi bittikten sonra “M-File” dosyasının “File” menüsünden “Save As” seçeneği seçilir. Yazılan program buradaki “work” çalışma dosyasının içine istenilen isimle kaydedilir. Kaydedilen dosyaların uzantısı .m olarak kaydedilir. (Örnek: **sonuc.m**) İstenen dosyayı çalıştırmak için ise komut satırında dosya ismini yazıp “Enter” tusuna basmak yeterlidir. Ayrıca “work” menüsüne girilip oradan da çalıştırılabilir.

İlk etapta örnek olarak oluşturulmuş olan boş bir M-File dosyası aşağıda gösterilmistir:



## A) GENEL AMAÇLI KOMUTLAR

-- “ **workspace** ” komutu : Çalışma esnasında çalışma alanında (bellegin kullanıcıya ayrılan kısmı) mevcut değişkenler hakkında ayrıntılı bilgi aşağıda gösterildiği gibi bir pencere içinde verilir. Pencere içindeki değişkenlerin üzerine çift tıklayarak o değişkenle ilgili bilgilere ulaşılıp değişiklikler yapılabilir.

The screenshot shows the MATLAB workspace browser window titled "Workspace". The menu bar includes File, Edit, View, Web, Window, and Help. The toolbar has icons for New, Open, Save, and Stack. The stack dropdown is set to "Base". The main area displays a table with four rows:

Name	Size	Bytes	Class
A	1x1	8	double array
B	1x1	8	double array
C	1x1	8	double array
V	1x3	24	double array

At the bottom of the window, there is a "Ready" status bar.

-- “ **clear** ” komutu : Bellekte o anda mevcut bulunan değişkenleri bellekten siler.

-- “ **save** ” komutu : M-file dosyasının kaydedilmesi yukarıda da anlatıldığı gibi “File” menüsünden yapılır. MATLAB komut satırında ise “**save**” komutu kullanılırsa o esnada bellekte bulunan değişkenleri, istenilen dosya ismiyle ve uzantısı .mat olacak şekilde kaydedilir.(Örnek: **sayilar.mat**)

```
>> a=1  
a =  
1  
>> b=2  
b =  
2  
>> save sayilar
```

Yukarıda a ve b sayıları sayilar.mat dosyası olarak kaydedilmiştir.

-- “ **load** ” komutu : Diskte saklı bir dosya içindeki değişkenleri tekrar bellege yükler.

```
>> load sayilar  
>> who  
Your variables are:  
a b
```

Yukarıda “sayilar” dosyası “load” komutuyla bellege tekrar yüklenmiş ve “who” komutuyla bu dosyanın değişkenleri görüntülenmiştir.

-- **“ dir ” komutu** : Bellekte kayitli olan dosyalari listeler.

(Not: Bir programi dogru olarak calistirmak icin, icra ettirmeden once “clear” komutuyla mevcut degiskenler silinebilir.)

```
>> dir
.
.
.. Oy.m          sayilar.mat      simple-report.sgml
        diary        simple-report.html
```

-- **“ type ” komutu** : Bir .m uzantili dosyanin içerigini komut satirinda görüntüler.

-- **“ edit ” komutu** : Bir M-dosyasinin içerisinde degisiklik yapma imkani saglar.

-- **“ open ” komutu** : Uzantisi ile belirtilen dosyayı açar.

-- Her yazılım türünde olduğu gibi MATLAB’de de, isletim sistemi kontrolüne geçmeden isletim sisteminin görevi olan bazi islemleri gerçeklestirebilmek mümkündür. Bu tür islemler asagidaki tabloda özetlenmiştir.

Komutun Adı	Komutun Islevi
cd	Aktif dizini degistirir.
dir	Aktif dizinin içindekileri listeler.
ls	Aktif dizinin içindekileri listeler.
delete	Belirtilen dosyayı siler.
type	Belirtilen dosyanın içeriğini listeler.

-- **“ clc ” komutu** : Komut satirini tamamen siler.

## **B) DEGISKEN ATAMA**

“ C ” ve “ PASCAL ” gibi programlama dillerinde, programın ana gövdesinin olusturulmasına baslamadan hemen önce, programdaki tüm degiskenlerin hangi tip degisken olduklarini belirtmek ve programın bilgisayar belleginden uygun miktarda alani bu degiskenler için tahsis etmesini saglamak gereklidir. MATLAB’de degiskenler, kendilerine ait bir isim ve onlara atanacak degerler yardımıyla olusturulurlar. Önceden degisken tipini belirtmeksiz, degiskene verilen degere baglı olarak MATLAB, uygun degisken tipini belirler ve bilgisayar belleginden yeteri kadar yeri bu degiskene tahsis eder.

Birinci bölümde temel atamalarla ilgili bazi bilgiler (sabit, degisken, matris, dizi, vektör tanimlama...) verilmisti.

### **-- Mantik ve Iliski Operatörleri :**

==	Esittir	&	and	Ve
~=	Esit degil	&	and	Ve
<	Küçük	~	not	Degil
<=	Küçük esit			
>	Büyük			
>=	Büyük esit			

-- **global()** komutu : Farklı M-dosyaları için aynı degisken tanımlanacaksa o degisken global(x) olarak tanımlanabilir.

-- **disp('')** komutu : Istenen açıklamayı görüntüler.

```
>> disp('Programlamaya Giris')
Programlamaya Giris
```

-- **input('')** komutu : Kullanicidan klavye aracılığıyla programci tarafından girilmesi istenen degisken istenir ve ilgili degiskene atanır.

```
>> Yas=input('Yasinizi giriniz :')
Yasinizi giriniz :23
Yas =
23
```

-- **“ fprintf ” komutu** : Bir açıklama ifadesiyle birlikte bir veya birden fazla degerin görüntülenebilmesini saglar.

```
>> a=231565465;
>> fprintf('Hesap = %d ',a)
Hesap = 231565465.000000
```

**Not :** “ fprintf ” fonksiyonu, kompleks sayilarin sadece reel kismini gösterir. Bu nedenle kompleks sayı uygulamalarında “ disp ” fonksiyonunu kullanilmalidir.

“ sprintf ” fonksiyonunda kullanılan çeşitli “ biçim tipleri ” asagida gösterilmistir :

%d : Virgülü sayıları 10'un kuvvetleri şeklinde gösterir.

%f : Kayan noktalı şekilde gösterir, aksi belirtilmemişçe virgülden sonra 6 basamak gösterir.

%e : Sayiyi üstel şekilde gösterir.

-- **linspace ve logspace komutları** : İlk değeri ve son değeri belirtilen bir diziyi lineer veya logaritmik olarak belirtilen sayıdaki elemani kullanarak yapılandırır. Belirtilen aralığı otomatik olarak verilen eleman sayısına göre böler ve her böldüğü sayiyi görüntüler.

fonksiyon(ilk\_deger , son\_deger , eleman\_sayisi)

```
>> B=linspace(0,10,6)
B =
    0    2    4    6    8    10
```

-- **Baslangic, son deger ve artis miktarı belli dizilerin atanması** :

```
>> dizi=10:5:30
dizi =
    10    15    20    25    30
```

--**Hazır fonksiyon özelliklerini kullanarak oluşturulan diziler için kullanılan komutlar** :

zeros(n,m) : nxm boyutunda 0'lardan oluşan matris üretir.

ones(n,m) : nxm boyutunda 1'lerden oluşan matris üretir.

eye(n,m) : nxm boyutunda birim matris üretir.

length(x) : "x" dizisinin satır sayısını verir.

size(x) : "x" matrisinin boyutlarını (satır ve sütun) verir.

format short : İşlem sonuçlarını virgülden sonra 4 basamaklı olarak gösterir.

format long : İşlem sonuçlarını virgülden sonra 14 basamaklı olarak gösterir.

**Not:** MATLAB'in yapısında önceden tanımlanmış bazi özel sabit veya açıklama değerler :

pi (Pi sayısı) ; i,j (Kompleks i sayısı) ; eps (Epsilon:Iki sayı arasındaki en küçük fark) ; Inf (Sayı/Sıfır belirsizliği ve diğer belirsizlikler karşısında üretilen tanımsızlık cevabı : Infinite)

### -- Vektör ve matrislerin tanimlanması :

```
>> A=[ 2 3 5 9 -2 ]  
A =  
    2    3    5    9   -2
```

3x3 boyutunda bir matrisi tanımlamak için :

```
>> C=[3 6 9 ; 8 2 4 ; 0 8 3 ]  
C =  
    3    6    9  
    8    2    4  
    0    8    3
```

-- “**who**” komutu : Çalışma alanındaki o esnada mevcut olan değişkenlerin isimlerini listeler. Ayrıca “ whos ” komutu değişkenler hakkında daha ayrıntılı bilgi verir.

```
>> who  
Your variables are:  
A B C V
```

-- length() komutu : Girilen bir vektörün uzunlugunu (eleman sayısını) görüntüler.

```
>> A=[ 5 8 7 2 5 9 6 ]  
A =  
    5    8    7    2    5    9    6  
>> length(A)  
ans =  
    7
```

## C) FONKSIYON OLUSTURMA VE DIGER KOMUTLAR:

-- sum( ), min( ), max( ), mean( ) komutları : Yandaki komutlar sırasıyla bir vektörün elemanlarının toplamını, en küçük elemanını, en büyük elemanını ve ortalamasını bulur.

-- “**function**” komutu : Fonksiyon tanımlamak için kullanılır. Bu özellik aşağıda örnek üzerinde açıklanmıştır :

Örnek olarak kullanıcı tarafından girilen n adet rakamın (Bu rakamların MATLAB’de vektör formunda girilmesi gerekmektedir) ortalamasını, en küçük elemanını, en büyük elemanını ve standart sapmasını bulacak bir fonksiyonu ist(a) adıyla oluşturalı :

The screenshot shows the MATLAB R12 workspace window. The title bar reads 'C:\matlabR12\work\ist.m'. The menu bar includes File, Edit, View, Text, Debug, Breakpoints, Web, Window, and Help. Below the menu is a toolbar with various icons. The main area contains the following MATLAB code:

```
function [ort,enk,enb,stsapma]=ist(a)
1 - enk=min(a)
2 - enb=max(a)
3 - n=length(a)
4 - ort=sum(a)/n
5 - stsapma=sqrt(sum((a-ort).^2)/n)
6 -
7 - end
```

At the bottom left, there is a 'Ready' status message.

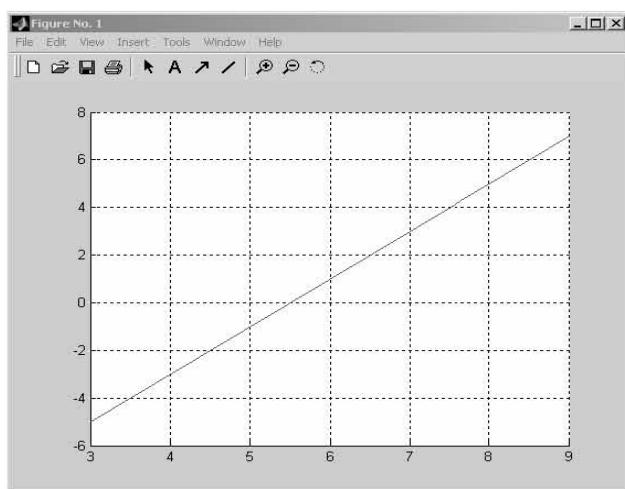
**Not:** Herhangi bir .m dosyasi penceresinde " f " butonuna mouse ile bir defa dokunuldugunda (yukaridaki sekilde de gorulmektedir) "work" dizini altinda kayitli olan bütün fonksiyonlar listelenir.

Simdi de programi komut satirinden icra ettirelim :

```
>> A=[5 3 6 9 73 6 5];
>> ist(A)
enk =
    3
enb =
    73
n =
    7
ort =
    15.2857
stsapma =
    23.6203
ans =
    15.2857
```

Bu program komut satirinda calistirilirsa asagida gösterildigi gibi icra edilir ve grafik de otomatik olarak yine asagida gösterildigi gibi çizdirilir.

```
>> dogru
*****Iki Noktasi Bilinen Dogrunun Cizimi*****
1. Noktanin apsisini giriniz :3
1. Noktanin ordinatini giriniz :9
2. Noktanin apsisini giriniz :-5
2. Noktanin ordinatini giriniz :7
>>
```

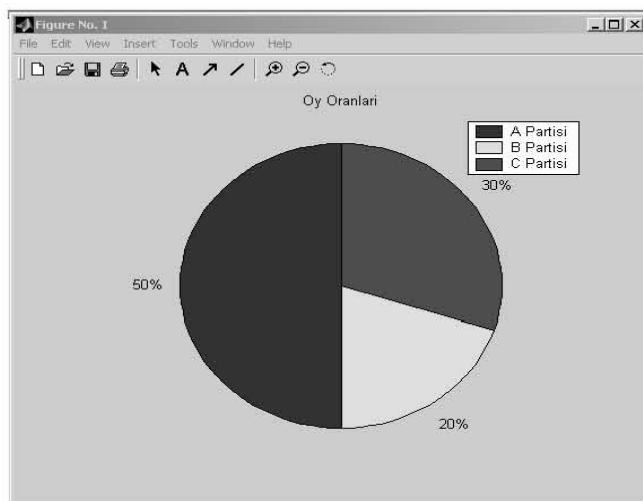


Örnek Program: “ 3 farklı partinin yapılan genel seçimler sonucu aldıkları oy oranlarını pasta grafik dilimi üzerinde gösteren grafigin çizimi ” programı aşağıda gösterildiği gibi Oy.m dosyası adı altında yazılmıştır. Çizilmesi istenen pasta dilimi grafigi için kullanıcidan bu üç partiye ait yüzdelik oy oranı rakam olarak istenmekte ve grafik otomatik olarak çizdirilmektedir.

```
C:\matlabR12\work\Oy.m
File Edit View Text Debug Breakpoints Web Window Help
Stack: Base
1 - clear
2 - disp('<><><><><><><><><><><><><><><><><><>');
3 - disp('Partilerin Oy Oranlari Grafigi');
4 - disp('<><><><><><><><><><><><><><><><><><>');
5 - A=input('A Partisinin aldigı oyun yuzdelik dilimdeki rakamini giriniz: ');
6 - B=input('B Partisinin aldigı oyun yuzdelik dilimdeki rakamini giriniz: ');
7 - C=input('C Partisinin aldigı oyun yuzdelik dilimdeki rakamini giriniz: ');
8 - V=[A B C];
9 - pie(V);
10 - title('Oy Oranlari');
11 - legend('A Partisi','B Partisi','C Partisi')
12
```

Bu program komut satirinda çalıştırılırsa asagida gösterildiği gibi icra edilir ve grafik de otomatik olarak yine asagida gösterildiği gibi çizdirilir.

```
>> Oy
<><><><><><><><><><><><><><><><><>
Partilerin Oy Oranları Grafigi
<><><><><><><><><><><><><><><><><>
A Partisinin aldığı oyun yuzdelik dilimdeki rakamini giriniz: 50
B Partisinin aldığı oyun yuzdelik dilimdeki rakamini giriniz: 20
C Partisinin aldığı oyun yuzdelik dilimdeki rakamini giriniz: 30
```



-- **“ keyboard ” komutu** : Isletilen programların kodlarına ve içeriklerine disaridan müdahale edebilmek için kullanılır. Böylece programın isletilmesi durdurulur, fonksyonun kontrol edilebilmesini saglar, degiskenler üzerinde inceleme ve degisiklik yapılabılır. Bu komutun aktif oldugu “K>>” seklindeki özel bir komut istem simbolü ile belirtilir. Bu özellikten çikip programın akisina devam edebilmek için, “ return ” komutu girilir ve “ enter ” tusuna basılır. Bu özellik sayesinde çalışma alanında üretilen sonuclara anında ulasabilmek mümkün olmaktadır. Asagidaki örnekte M-File ve komut icra satirları verilmistir

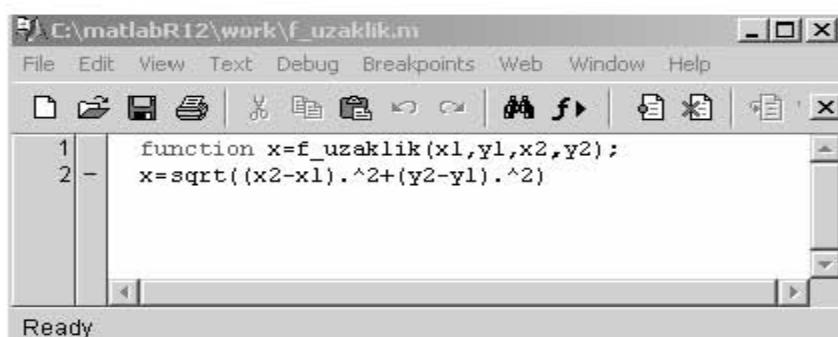
```
C:\matlabR12\work\keyboard_ornek.m
File Edit View Text Debug Breakpoints Web Window Help
1 - a=input('Birinci sayiyi giriniz: ');
2 - b=input('Ikinci sayiyi giriniz: ');
3 - keyboard
4 - d=a*b*c
5 - end
```

Bu program komut satirinda calistirilirsa asagida gösterildigi gibi icra edilir

```
Birinci sayiyi giriniz: 4
Ikinci sayiyi giriniz: 2
K>> c=10
c =
    10
K>> return
d =
    80
K>> return
>>
```

**Örnek Program:** İki nokta arasindaki uzakligi bulan basit bir programi, önce fonksiyon yapisini kullanarak sonra da ayni programi bu fonksiyon yapisini bellekten çagirma islemini uygulayarak icra ettirelim :

a) Fonksiyon yapisini kullanarak :



Komut satirinden icra edilirse :

```
>> f_uzaklik(2,6,3,0)
x =
    6.0828
ans =
    6.0828
```

b) Fonksiyon yapısını bellekten çağırarak :

The screenshot shows the MATLAB R12 interface with a window titled 'C:\matlabR12\work\uzaklik.m'. The menu bar includes File, Edit, View, Text, Debug, Breakpoints, Web, Window, and Help. Below the menu is a toolbar with various icons. The code editor displays the following M-file:

```
1 - disp('----- Iki nokta arasindaki uzaklik -----');
2 - ax=input('1.noktanin x degerini giriniz : ');
3 - ay=input('1.noktanin y degerini giriniz : ');
4 - bx=input('2.noktanin x degerini giriniz : ');
5 - by=input('2.noktanin y degerini giriniz : ');
6 - uzaklik=f_uzaklik(ax,ay,bx,by);
7 - fprintf('Iki nokta arasindaki uzaklik:%f ',uzaklik);
8 - end
```

**Komut satırından içra edilirse :**

```
>> uzaklik  
----- İki nokta arasindaki uzaklik -----  
1.noktanin x degerini giriniz : 2  
1.noktanin y degerini giriniz : 5  
2.noktanin x degerini giriniz : 4  
2.noktanin y degerini giriniz : 3  
x =  
    2.8284  
İki nokta arasindaki uzaklik:2.828427
```

#### **D) DÖNGÜ ve SARTLI İFADE UYGULAMALARI:**

## **1. SARTLI İFADELER :**

-- **"if"** yapisi : "if" komutunun MATLAB'de 3 farklı sekli mevcuttur :

```
a) if kosul
    deyim1
    deyim 2
    deyim_n
end
```

Kosul doğru ise deyim1, deyim1, ..., deyim\_n, ile belirtilen deyimler grubu icra edilir ve programın kontrolü end’i izleyen deyime geçer; kosul yanlış ise bu durumda deyim1, deyim2, ..., deyim\_n ile belirtilen deyimler grubu icra edilmeden kontrol end’i izleyen deyime geçecektir.

```
b) if kosul
    deyim1
    deyim 2
    deyim_n
else deyim_n+1
    deyim_n+2
    deyim_m
end
```

Kosul doğru ise deyim1, deyim1, ..., deyim\_n, ile belirtilen deyimler grubu icra edilir ve programın kontrolü end'i izleyen deyime geçer, kosul yanlış ise bu durumda da sadece else' i izleyen, deyim1\_n+1, deyim\_n+2, ..., deyim\_m ile belirtilen deyimler grubu icra edilecek ve kontrol end' i izleyen deyime geçecektir.

```
c) if kosull1
    deyim1
elseif kosul2
    deyim2
elseif kosul3
    deyim3
...
elseif kosul_n
    deyim_n
else
    deyim_n+1
end
```

Bu yapı içerisinde kontrol edilen koşullardan herhangi biri doğru ise onunla ilişkili deyim icra edilir ve kontrol end' i izleyen deyime geçer. Koşulların hepsi de yanlışsa, kontrol else' i izleyen deyim\_n+1'e geçer ve bu deyim de icra edildikten sonra kontrol end' i izleyen deyime geçecektir.

**Örnek Program:** Kullanıcı tarafından bir sayı istenip bu sayının pozitif, negatif veya 0 mi olduğunu sorulayan ve ekrana yazdırın program aşağıda incelenebilir :

```
C:\matlabR12\work\if_1.m
File Edit View Text Debug Breakpoints Web Window Help
1 - x=input('Bir sayı giriniz: ');
2 - if x < 0
3 -     disp('Sayınız negatiftir... ');
4 - elseif x==0
5 -     disp('Sayınız sıfırdır... ');
6 - else
7 -     disp('Sayınız pozitiftir... ')
8 - end
Ready
```

Komut satırından icra edilirse :

```
Bir sayı giriniz: 55
Sayınız pozitiftir...
>> if_1
Bir sayı giriniz: -9
Sayınız negatiftir...
>> if_1
Bir sayı giriniz: 0
Sayınız sıfırdır...
```

-- **“ switch - case ” yapısı** : Ikiden fazla durumu kontrol etmek için, if – elseif – else – end yapısına alternatif olarak kullanılan bir kontrol yapısıdır.

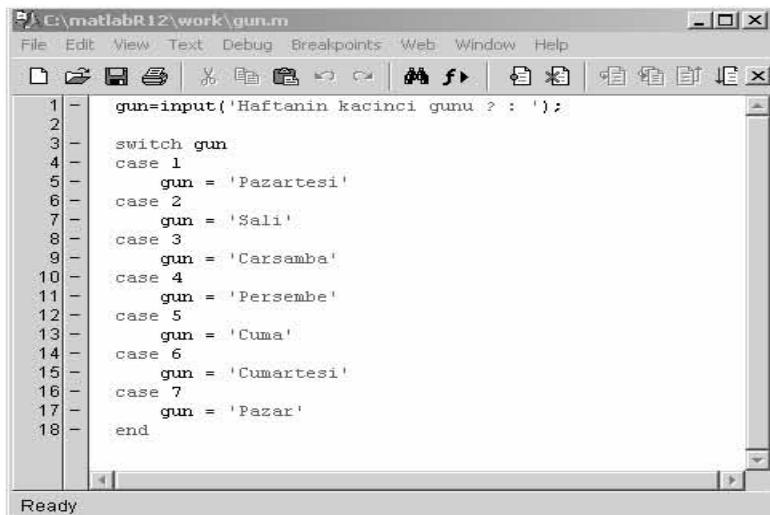
Ifadenin değeri kontrol edilir ve buna göre farklı bir deyim veya deyimler grubu icra edilir. Örneğin, ifadenin değeri değer1 ise, deyim1 icra edilir ve kontrol end’i izleyen deyime geçer.

Ifadenin değeri , değer2, değer3 ya da değer4 ’e eşitse bu durumda deyim2 icra edilir ve kontrol end’i izleyen deyime geçer.

Ifadenin değeri, case’lerle kontrol edilen değerlerden hiç birine eşit değilse bu durumda da otherwise sözcüğünü izleyen deyim\_n+1 icr edilecektir.

```
switch(ifade)
case değer1
  deyim1
case değer2,değer3,değer4
  deyim2
  ...
case değer_n
  deyim_n
otherwise
  deyim_n+1
end
```

**Örnek Program:** Haftanın kaçinci gününün ne olduğunu bulan basit bir program aşağıda incelenebilir :



```
C:\matlabR12\work\gun.m
File Edit View Text Debug Breakpoints Web Window Help
1 gun=input('Haftanın kaçinci gunu ? : ');
2
3 switch gun
4 case 1
5     gun = 'Pazartesi'
6 case 2
7     gun = 'Sali'
8 case 3
9     gun = 'Carsamba'
10 case 4
11     gun = 'Persembe'
12 case 5
13     gun = 'Cuma'
14 case 6
15     gun = 'Cumartesi'
16 case 7
17     gun = 'Pazar'
18 end
Ready
```

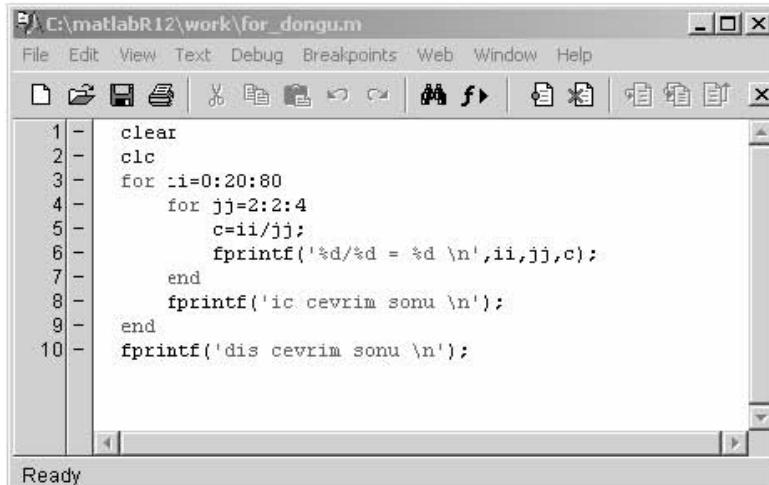
```
>> gun
Haftanın kaçinci gunu ? : 5
gun =
    Cuma
```

## 2.DÖNGÜ İFADELERİ :

-- **“ for ” döngüsü :** Parametre değeri başlangıç değerinden baslayarak ve her seferinde artım değeri kadar arttırılarak son degere erişene kadar degistirilir. Parametrenin her değeri için, deyim1, deyim2, deyim\_n şeklinde belirtilen ve for-end sözcükleri arasında yer alan deyimler grubu icra edilir. Parametrenin değeri son degeri asinca, programın kontrolü end’i izleyen deyime yani çevrimin disina çıkacaktır.

```
for parametre=baslangic:artim:son_deger
    deyim1
    deyim2
    ...
    deyim_n
end
```

**Örnek Program:** Arka arkaya bölme islemlerinin yapildigi asagidaki for döngüsü programini incelenebilir :



```
C:\matlabR12\work\for_dongu.m
File Edit View Text Debug Breakpoints Web Window Help
1 - clear
2 - clc
3 - for ii=0:20:80
4 -     for jj=2:2:4
5 -         c=ii/jj;
6 -         fprintf('%d/%d = %d \n',ii,jj,c);
7 -     end
8 -     fprintf('ic cevrim sonu \n');
9 - end
10 - fprintf('dis cevrim sonu \n');

Ready
```

Komut satirinden icra edilirse :

```
0/2 = 0
0/4 = 0
ic cevrim sonu
20/2 = 10
20/4 = 5
ic cevrim sonu
40/2 = 20
40/4 = 10
ic cevrim sonu
60/2 = 30
60/4 = 15
ic cevrim sonu
80/2 = 40
80/4 = 20
ic cevrim sonu
dis cevrim sonu
```

-- **“ while ” döngüsü :** Belirli bir üst sinira kadar istenilen islemeleri tekrarlayarak yapar. Kosul dogru oldugu surece, deyim1, deyim2, ... , deyim\_n seklinde belirtlen deyimler grubunu icra eder. Kosul yanlis oldugu anda, end'i izleyen deyime yani cevrim disina çıkar.

**while** kosul

    deyim1  
    deyim2

...

    deyim\_n

**end**

**Örnek Program:** A=5 ilk değerinden başlayarak A<8 olduğu sürece A'ya 1 eklemek için aşağıdaki program incelenebilir :

The screenshot shows the MATLAB R12 interface with a window titled 'C:\matlabR12\work\while\_A.m'. The menu bar includes File, Edit, View, Text, Debug, Breakpoints, Web, Window, and Help. Below the menu is a toolbar with various icons. The code editor displays the following script:

```
1 - A=5
2 - while A<8
3 -     A=A+1
4 - end
```

**Komut satırından icra ettirilirse :**

$$\begin{array}{r} A = \\ 5 \\ A = \\ 6 \\ A = \\ 7 \\ A = \\ 8 \end{array}$$

## **ALISTIRMALAR**

- 1-** Kenarları kullanıcı tarafından istenen bir üçgenin çeşitkenar, ikizkenar veya eskenar üçgen mi olduğunu bulan MATLAB programını yazınız.
- 2-** “  $ax^2+bx+c$  ” ikinci dereceden denkleminin köklerini bulduran MATLAB programını yazınız.
- 3-** Bir otomobil, durgun halden harekete baslayarak 10 dakika boyunca hızlanıyor, hızı 60 km/saat oluyor. Sonra 15 dakika boyunca sabit hızla hareketine devam ediyor ve 10 dakika boyunca yavaşlayarak hızı 0 oluyor. Disarıdan girilen herhangi bir tıkanda otomobilin hızını veren bir MATLAB programı yazınız.
- 4-** Kullanıcıdan doğum gününü soran ve bu kullanıcının kaç yıl, kaç ay ve kaç yıl yasadığını bulan bir MATLAB programı yazınız.

## **BÖLÜM 4 : KONTROL SİSTEMLERİNDE ZAMAN-FREKANS ANALİZİ**

Matematiksel modeller lineer sistemlere veya baska sistemlere MATLAB komutlari vasitasiyla kolaylikla dönüştürülebilir.

Asagida kontrol sistemleri için gerekli bazi dönüşümler açıklanmistir :

-- Transfer fonksiyonundan durum uzayina çevirme :

$[A \ B \ C \ D] = tf2ss(num,den)$  komutu,

$$\frac{Y(s)}{U(s)} = \frac{num}{den} = \mathbf{C} (\ s\mathbf{I}-\mathbf{A} )^{-1} \ \mathbf{B} + \mathbf{D} \text{ sistemini,}$$

$$\frac{dx}{dy} = \mathbf{Ax} + \mathbf{Bu}$$

$y = \mathbf{Cx} + \mathbf{Du}$       durum uzayı formatına çevirir.

-- Durum uzayindan transfer fonksiyonuna çevirme :

$[num,den] = ss2tf(A,B,C,D)$

Eger sistemin birden fazla girişi varsa asagidaki komut kullanılır :

$[num,den] = ss2tf(A,B,C,D,iu)$

Örnek olarak asagida verilen iki girişli bir sistemin diferansiyel denklem takiminin MATLAB komut satirindaki islemleri ve çıkışları incelenebilir. Ayrıca sistemin iki girişi olduğu için  $u_1$  girişi ele alındığında  $u_2$  sıfır,  $u_2$  girişi ele alındığında da  $u_1$  girişi sıfır olarak kabul edilmektedir.

$$\frac{d}{dt} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$y = [1 \ 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + [0 \ 0] \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

```
Command Window
>> A = [ 0  1 ; 2  3];
>> B = [ 1  0 ; 0  1];
>> C = [ 1  0];
>> D = [ 0  0];
>> [num,den] = ss2tf(A,B,C,D,1)

num =
      0    1.0000   -3.0000

den =
      1    -3    -2

>> [num,den] = ss2tf(A,B,C,D,2)

num =
      0    0    1

den =
      1    -3    -2
>>
```

Buradan elde edilen katsayilarla :

$$\frac{Y(s)}{U_1(s)} = \frac{s^2 + 3s + 2}{s^3 + 3s^2 + 2s}$$

r,p,k degerleri sirasiyla rezidü (n.inci basit kesrin payi), kutup (n.inci basit kesrin paydasi) ve sabit terimleri temsil etmektedir.

$$\frac{B(s)}{A(s)} = \frac{r1}{s - p(1)} + \frac{r2}{s - p(2)} + \dots + \frac{r(n)}{s - p(n)} + k(s)$$

Örnek olarak asagidaki kesir ifadesini basit kesirlerine ayiralim.

$$\frac{B(s)}{A(s)} = \frac{2s^3 + 5s^2 + 3s + 6}{s^3 + 6s^2 + 11s + 6}$$

```
Command Window
>> num = [2 5 3 6];
>> den = [1 6 11 6];
>> [r,p,k] = residue(num,den)

r =
    -6.0000
    -4.0000
    3.0000

p =
    -3.0000
    -2.0000
    -1.0000

k =
    2

>>
```

Böylece kesir ifadesi su sekilde basit kesirlerine ayrılmıştır :

$$\frac{B(s)}{A(s)} = \frac{-6}{s+3} + \frac{-4}{s+2} + \frac{3}{s+1} + 2$$

-- Sürekli zaman sistemleri için geçici cevap analizi :

Geçici tepki analizleri ( Basamak fonksiyonu, anlık darbe foksiyonu, birim basamak fonksiyonu, rampa fonksiyonu,...) kontrol sistemlerinin zaman degiskeni karsisindaki karakteristiklerini belirlemek için kullanılır.

Basamak fonksiyonu ( Step function ):

step(num,den)                  step(num,den,t)

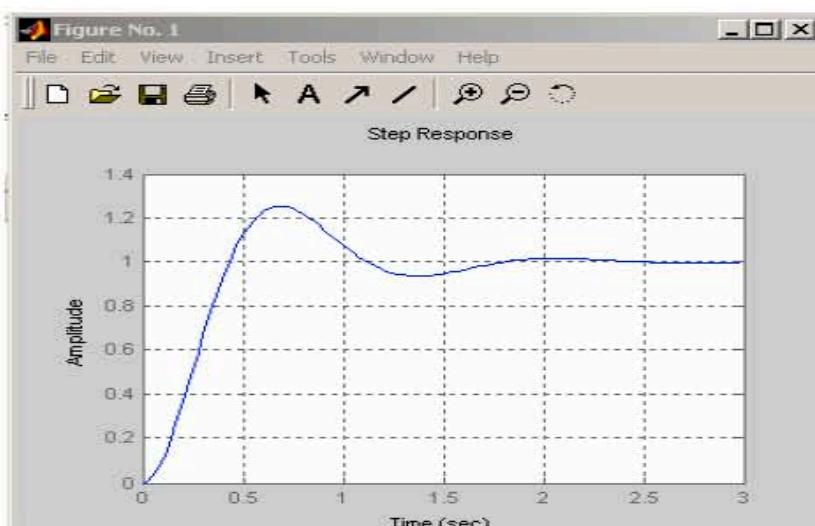
Durum uzayında ifade edilen (A,B,C,D) durum uzayinin matrislerini göstermek kosuluyla) bir kontrol sistemi için :

step(A,B,C,D)

**Örnek:** Transfer fonksiyonu  $\frac{C(s)}{R(s)} = \frac{25}{s^2 + 4s + 25}$  olarak verilen bir sistemin birim basamak fonksiyonuna vereceği cevap analizi aşağıda gösterilmistir.

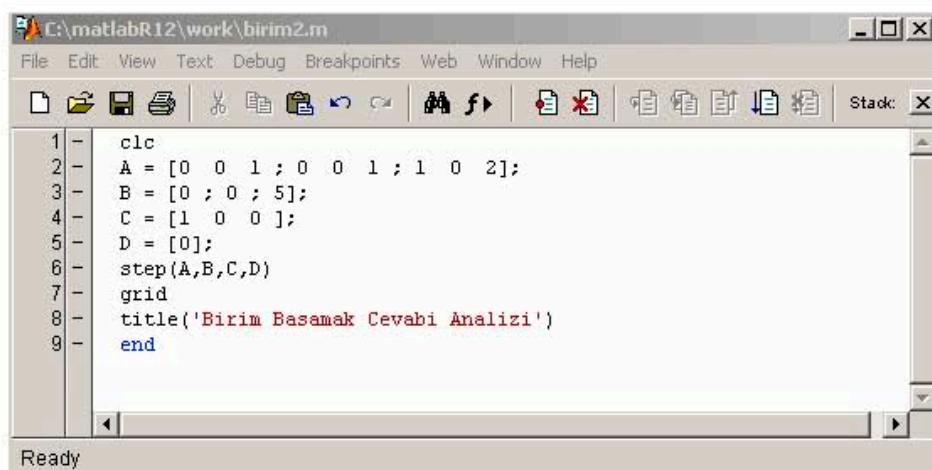
The screenshot shows the MATLAB Editor window titled 'C:\matlabR12\work\step1.m'. The code in the editor is as follows:

```
% Birim basamak (step function) fonksiyonu cevabi %
1 clc
2 clear all
3 num=[0 0 25];
4 den=[1 4 25];
5 step(num,den)
6 grid
7 title('Birim Basamak Cevabi');
8 end
```



Örnek:  $A = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 2 \end{bmatrix}$ ;  $B = \begin{bmatrix} 0 \\ 0 \\ 5 \end{bmatrix}$ ;  $C = [1 \ 0 \ 0]$ ;  $D = 0$

Yukarıda A,B,C,D durum uzayı matrisleri verilen bir sistemin birim basamak fonksiyonuna karşı vereceği cevabı çizdiriniz.

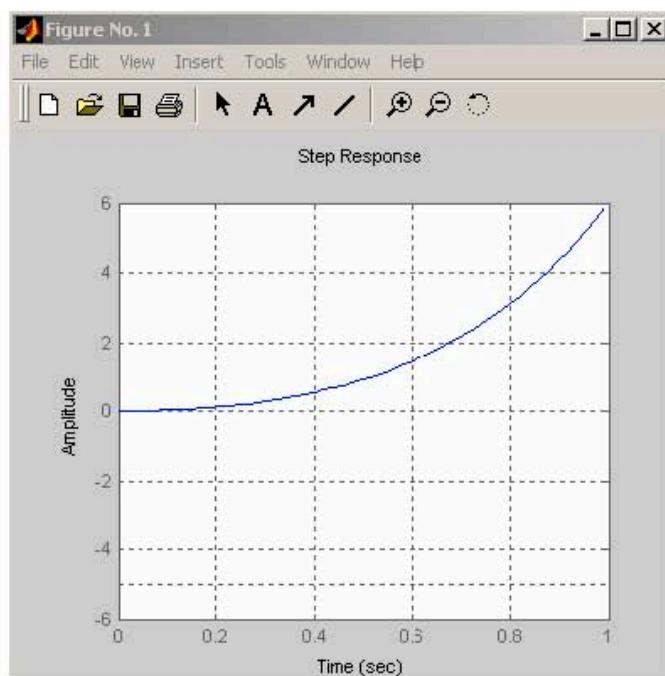


```

C:\matlabR12\work\birim2.m
File Edit View Text Debug Breakpoints Web Window Help
File Open Save Save As New Close Find Replace Run Stop Stack
1 - clc
2 - A = [0 0 1 ; 0 0 1 ; 1 0 2];
3 - B = [0 ; 0 ; 5];
4 - C = [1 0 0];
5 - D = [0];
6 - step(A,B,C,D)
7 - grid
8 - title('Birim Basamak Cevabi Analizi')
9 - end

```

Ready



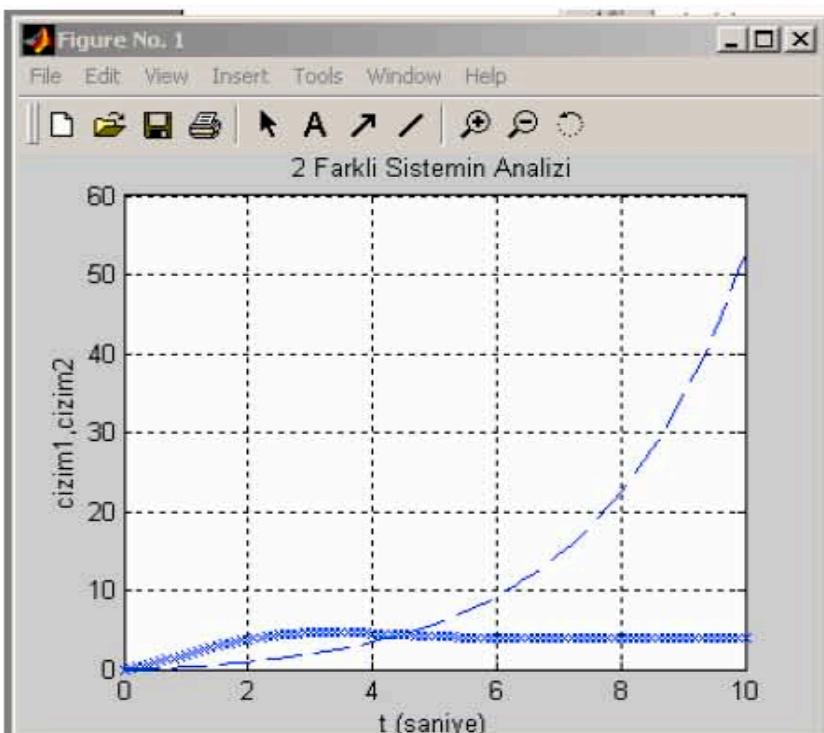
Bir sistemin anlık darbe fonksiyonu ( Impulse Function ) ve rampa fonksiyonuna (Ramp Function) vereceği cevabı bulmak için de bu transfer fonksiyonlarının önlerine sırasıyla  $1/s$  ve  $1/s^2$  çarpanı oluşturulur ve normal basamak fonksiyonu gibi çizilir.

**Örnek:** İki farklı transfer fonksiyonuna ilişkin değerler aşağıda verilmiştir. Bu sistemlerin 0 ve 10. saniyeler arasında frekans cevabını 0.1 aralıklarla, aynı grafik ekranı üzerinde çizdiriniz.

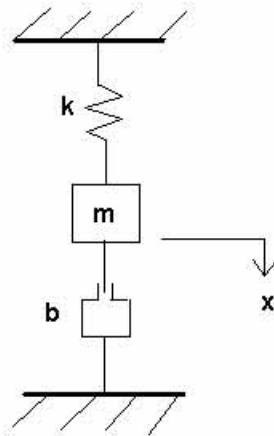
$\text{num1} = [0 \ 0 \ 1]$  ;  $\text{den1} = [1 \ 2 \ -1]$

$\text{num2} = [0 \ 1 \ 4]$  ;  $\text{den2} = [1 \ 1 \ 1]$

```
C:\matlabR12\work\sis.m
File Edit View Text Debug Breakpoints Web Window Help
 1 num1=[0 0 1];
 2 den1=[1 2 -1];
 3 num2=[0 1 4];
 4 den2=[1 1 1];
%num3=[1 2 1];
%den3=[0 1 1];
 5
 6
 7 t=0:0.1:10;
 8
 9 a=step(num1,den1,t);
10 plot(t,a,'--')
11 hold
12
13 b=step(num2,den2,t);
14 plot(t,b,'x')
15 hold
16
17 grid
18 title('2 Farklı Sistemin Analizi');
19 xlabel('t (saniye)');
20 ylabel('cizim1,cizim2');
21 end
```



Örnek:



Yukarıdaki mekanik sistemi Laplace Dönüşümü ile analiz ederek transfer fonksiyonunu bulunuz ve birim basamak fonksiyonu yardımıyla grafigini çizerek x'in (yol miktari) genlik degisimini inceleyiniz.

(Baslangic kosullari :  $x(0)=0.1$  m ve  $x^{(1)}(0)=0.05$  m/s )

Not: (1) ve (2) ifadeleri birinci ve ikinci türevleri temsil etmektedir.

Çözüm için önce aşağıdaki analitik yöntem, sonra da grafik incelenebilir.

Sistem denklemi :

$$mx^{(2)} + bx^{(1)} + kx = 0$$

$$m[s^2 X(s) - sx(0) - x^{(1)}] + b[sX(s) - x(0)] + kX(s) = 0$$

$$(ms^2 + bs + k)X(s) = mx(0)s + m x^{(1)}(0) + bx(0)$$

Gerekli matematiksel işlemler yapılip  $X(s)$  ifadesi çekilirse :

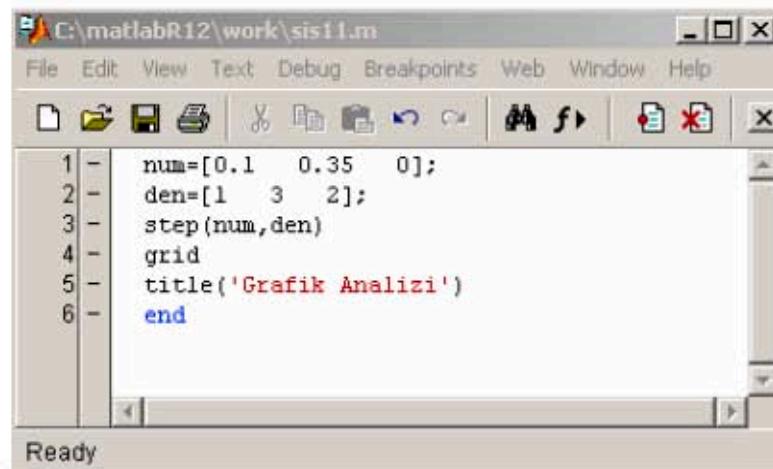
$$X(s) = \frac{mx(0)s + mx(0) + bx(0)}{ms^2 + bs + k} = \frac{0.1s + 0.35}{s^2 + 3s + 2}$$

$X(s)$  ifadesi  $\frac{1}{s}$  çarpanıyla yeniden düzenlenirse :

$$X(s) = \frac{0.1s^2 + 0.35s}{s^2 + 3s + 2} \cdot \frac{1}{s}$$

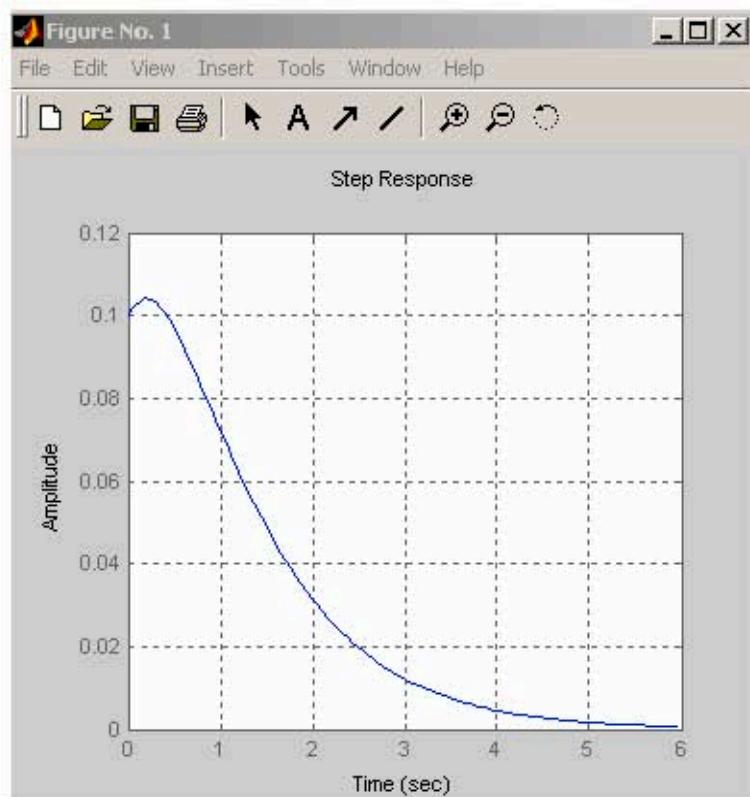
Böylece  $X(s)$  transfer fonksiyonu birim basamak fonksiyonu çizimi için  $G(s)$  fonksiyonuna dönüştürüülüp aşağıdaki sonuç elde edilir :

$$G(s) = \frac{0.1s^2 + 0.35s}{s^2 + 3s + 2}$$



```
C:\matlabR12\work\sist1.m
File Edit View Text Debug Breakpoints Web Window Help
 1 num=[0.1 0.35 0];
 2 den=[1 3 2];
 3 step(num,den)
 4 grid
 5 title('Grafik Analizi')
 6 end

Ready
```



## **ALISTIRMALAR**

**1-** İki farklı transfer fonksiyonuna iliskin değerler aşağıda verilmiştir. Bu sistemlerin 0 ve 15. saniyeler arasında birim basamak frekans cevabını 0.2 aralıklarla, aynı grafik ekranı üzerinde çizdiriniz.

$$\text{num1} = [1 \ 5 \ 1] ; \text{den1} = [3 \ 2 \ 4]$$

$$\text{num2} = [1 \ 0 \ 1] ; \text{den2} = [1 \ 0 \ 2]$$

**2-** Aşağıda A,B,C,D durum uzayı matrisleri verilen bir sistemin birim basamak fonksiyonuna karşı vereceği frekans cevabını çizdiriniz.

$$A = \begin{bmatrix} 3 & 1 & 1 & 1 \\ 2 & 3 & 0 & 1 \\ 1 & 9 & 5 & 3 \\ 0 & 7 & 1 & 1 \end{bmatrix} ; \quad B = \begin{bmatrix} 0 \\ 2 \\ 5 \\ 60 \end{bmatrix} ; \quad C = [1 \ 0 \ 0 \ 0] ; \quad D = 0$$

**3-** Bir sisteme ait transfer fonksiyonuna iliskin matris değerleri aşağıda verilmiştir. Bu sistemin 0 ve 40. saniyeler arasında birim basamak frekans cevabını 0.1 aralıklarla, çizdirip inceleyiniz.

$$\text{num1} = [4 \ 8 \ 9] ; \text{den1} = [1 \ 3 \ 2]$$

## **EK(Table): MATLAB KOMUTLARI ve MATRIS FONKSİYONLARI TABLOSU**

Komut	Görevi
abs	Mutlak Deger, kompleks büyüklük
angle	Faz Açısi
ans	Islem sonucu açıklaması olmadigi zamanki sonuç ifadesi (cevap)
atan	Arktanjant
axis	Eksen ölçekleme
bode	Bode diyagramini çizdirir
clear	Önceden tanimlanmis degiskenleri bellekten siler
clc	Komut satirini temizler
clg	Grafik ekranini temizler
computer	Kullanilan bilgisayar hakkında bilgi verir
conj	Kompleks konjüge
conv	Konvolüsyon
correcoef	Korelasyon katsayisi
cos	Kosinüs
cosh	Hiperbolik kosinüs
cov	Kovaryans
date	Tarihi görüntüler
deconv	Dekonvolüsyon
det	Determinant
diag	Kösegen matris
eig	Özdeger ve özvektörler
exit	Programi kapatır
exp	e tabanlı sayisinin üssü
eye	Birim matris
format long	Islem sonuçlarını virgülden sonraki haneleri kısa olarak gösterir
format short	Islem sonuçlarını virgülden sonraki haneleri uzun olarak gösterir
freqs	Laplace dönüşümü frekans sevabi
freqz	z-dönüşümü frekans cevabi
grid	Grafik ekranini ölçekli olarak gösterir
hold	Grafik ekranındaki son grafigi kaldırır
i	$\sqrt{-1}$
j	$\sqrt{-1}$
length	Vektör boyutu
log	Logaritma
loglog	Logaritmik x-y çizimi
logm	Matris logaritması
log10	10 tabanında logaritma
max	En büyük deger
mean	Ortalama
median	Orta deger
min	En küçük deger

Komut	Görevi
nyquist	Nyquist frekans cevabı çizimi
ones	Sabit 1 veya 1 matrisi üretme
pi	Pi sayısı
plot	Lineer x-y çizimi
polar	Kutupsal çizim
poly	Karakteristik polinom
polyval	Polinomda değer hesaplama
polyvalm	Matris polinomu hesabı
quit	Programı sonlandırır
rand	Rastgele sayılar ve matrisler üretir
rank	Matris ranki
real	Gerçel kısım
residue	Parçalı kesir pay kısımları hesaplama
roots	Polinom kökleri
sign	Signum Fonksiyonu
sin	Sinüs
sinh	Hiperbolik sinüs
size	Satır ve sütun boyutları
sqrt	Karekök
sqrtm	Matris kökü
std	Standart sapma
step	Birim basamak fonksiyonu çizimi
sum	Eleman toplamı
tan	Tanjant
tanh	Hiperbolik Tanjant
text	Grafik üzerine açıklama yazma
title	Grafik başlığı
trace	Matris izi hesaplama
who	Bellekteki tüm değişkenleri listeler
xlabel	x-ekseni açıklaması
ylabel	y-ekseni açıklaması
zeros	Sıfır veya sıfır matrisi üretme

## **KAYNAKLAR**

- [1] “ MATLAB ” , Yrd.Doç.Dr. Mehmet Uzunoglu, Türkmen Kitabevi, 2002
- [2] “ Solving Control Engineering Problems with MATLAB ” , Katsuhiko Ogata, 1994
- [3] “ MATLAB Primer – Third Edition ” , Kermit Sigmon, University of Florida
- [4] “Lineer Cebir ve MATLAB Uygulamalari ”, Prof.Dr. Mithat Uysal, Prof.Dr. Aysenur Uysal, Beta Yayınlari, 2000
- [5] “ [www.mathworks.com](http://www.mathworks.com) ”
- [6] “ <http://www.math.siu.edu/matlab/tutorial3.pdf> ”