

İCİNDEKİLER

BÖLÜM 1 : MATLAB Kullanımı ve Matris İşlemleri

BÖLÜM 2 : Grafik ve Eğri 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 içeriğine göre hazırlanmıştır.

BÖLÜM 1 : MATLAB KULLANIMI ve MATRIS ISLEMLERİ

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

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

help 'fonksiyon ismi'

komutu yazıldığında yardım istenilen fonksiyon hakkında detaylı bilgiye ulaşılabilmektedir.

help help

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

-- Matris Operatörleri :

Aşağıda verilen simgeler matris işlemlerinde kullanılmaktadır:

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

-- Mantık ve İlişki Operatörleri :

<	Küçük	&	Ve
<=	Küçük eşit		Veya
>	Büyük	~	Değil
~=	Eşit değil		

-- Baslangiç olarak komut satirina :

```
date
```

yazilrsa program tarafından geçerli olan tarih alınacaktır.Yani :

```
ans=  
30-Oct-2002
```

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

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

-- En son yazılan komutların hepsine üst ve alt yön tuşlarına dokunarak kolay bir şekilde ulaşılabilir.

-- En son tanımlanan herhangi bir 'x' değeri için yapılan işlemlerden sonra bu 'x' değeri komut satirina yazilip enter tusuna basilrsa daha önce neye karşılık olarak tanımlandığı 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.

-- Tanımlanan bu satir vektörünü sütun vektörüne dönüştürmek için :

```
y=x' yazilrsa ekranda görülen deger asagidaki gibi olacaktır:
```

```
y=  
1  
2  
3  
-4  
-5
```

-- Matris tanımlamak için aşağıdaki A matrisi verilmiş olsun :

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

Bu matrisi MATLAB'e tanıtmak için şu şekilde yazılmalıdır :

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

Yani her satırın sonunun neresi olduğunu konulan noktali virgül isareti temsil etmektedir.

-- Örnek olarak aşağıdaki B matrisini tanıtmak için :

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

$$B = [1 \ \exp(-0.02); \ \text{sqrt}(2) \ 3] \text{ şeklinde yazılmalıdır.}$$

Ekranda ise şu şekilde gözükcektir:

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

-- Apostrofi isareti (') matrisin konjüge transpozisinin alınmasına yarar. Eğer matris reel bir matris ise basit olarak transpoze alim işlemi olarak da tanımlanabilir..

Yeni bir A matrisi tanımlayalım :

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

Ekranda görülecek matris şu şekilde olacaktır :

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

Bu matrisin transpozmesini 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 :

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

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

Bu kompleks sayı üstel formatta da yazılabilmektedir :

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

Bu durumda komut satırına aşağıdaki ifade yazılacaktır :

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

i ve j daha önceden değişken olarak kullanılmısa 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 tanımlamak için aşağıdaki X matrisi verilmiş olsun:

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

Komut satırına ise şu şekilde girilecektir :

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

Bu durumda ekranda görülecek değer :

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

Y = X' komutu yazılırsa :

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

iletisi ekranda okunacaktır.

Daha önce de belirtildiği gibi yukarıdaki işlem konjüge transpoze olarak algılanmaktadır. Eğer sadece transpoze alınacaksa (konjügesiz) komut şu şekilde yazılmalıdır :

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

Bu durumda ekranda gözükcek değerler :

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

-- Toplama ve çıkarma işlemlerinin yapılması için aşağıdaki M ve N matrisleri verilmiş olsun :

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

Bu degerleri ekrana girmek için:

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

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

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

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

Toplama islemi için:

$$C = M+N$$

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

Eger x vektörü su asagidaki gibi verilirse :

$$X = \begin{array}{c} 5 \\ 4 \\ 6 \end{array}$$

Bu vektörü su sekilde tanitmak gerekir :

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

Örneğin $T = X-1$ gibi islemi gerçeklestirmek için :

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

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

```
x = [1 ; 2 ; 3];    y = [4 ; 5 ; 6];    A = [1 1 2 ; 3 4 0 ; 1 2 5]
```

```
x'*y
```

```
ans =  
    32
```

```
x*y'
```

```
ans =  
     4     5     6  
     8    10    12  
    12    15    18
```

```
b = A*x
```

```
b =  
     9  
    11  
    20
```

Bunların dışında matris bir skaler değerle de çarpılabilir :

```
5*A
```

```
ans =  
     5     5    10  
    15    20     0  
     5    10    25
```

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

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

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

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

$\text{angle}(A)$ ise faz açılarını radyan cinsinden A kompleks matrisi için hesaplamaktadır. Burada tanım değerleri $-?$ ve $+?$ 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 :

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

$$r = \text{abs}(z)$$

$$\theta = \text{angle}(z)$$

$$z = r \cdot \exp(i \cdot \theta)$$

-- 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 2; 3 4];  
A.^2
```

```
ans =  
     1     4  
     9    16
```

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

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

Bir örnek daha verilirse :

```
      1 2 3      4 5 6  
A =      1 9 8      B =      7 6 5
```

```
C = A.*B
```

```
C =  
     4 10 18  
     7 54 40
```

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

```
A=[1 2 3 4 ]  
  
A.^2  
  
ans =  
  
     1     4     9    16
```

-- 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 kullanilir :

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

ans =
   -1.8333   -1.0000   -0.1667
    1.0000    0.0000    0.0000
    0.0000    1.0000    0.0000
```

-- Çesitli komutlar ve durumlar tek bir sirada virgöl (,) veya noktali virgöl ile (;) ayrilarak yazilabilir.

-- Çikis formatini istedigimiz uzunlukta elde edebiliriz. Eger matris elemanlari tamsayi ise bu durum sonuça bir degisiklik yapmayacaktır. Bunun için asagidaki komutlari 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    0.000020000000000000
```

-- 1'den 5'e kadar sayilari 0.5'lik araliklarla yazdirmak istersek iki nokta'yi (:)
kullanmak yeterli olacaktır :

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

```
t = 1:0.5:3
```

```
t =  
    1.0000    1.5000    2.0000    2.5000    3.0000
```

Düzgün azalan biçimde yazdirirsak :

```
t = 5:-1:2
```

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

-- Bir matrisin i. satirini veya j. sütununu görüntülemek için asagida tanimlanan A
matrisini komutlariyla inceleyelim :

Asagidaki A matrisinin 2. satiri 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 elemanini bulmak için :

```
k = A(3,3)
```

```
k = -6
```

-- Bir matrisin boyutlarini öğrenmek için size(A) komutu, rankini bulmak için rank(A) kullanilir.

```
A=  
  2  3  2  
  5  4  1  
  2  6  8
```

```
size(A)
```

```
ans =  
     3     3
```

```
rank(A)
```

```
ans =  
     3
```

-- Bir matrisin determinantini bulmak için det(A) komutu kullanilir.

```
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 :

$$\text{norm}(x) = \sum(\text{abs}(x).^2)^{0.5}$$

```
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 satırlık bir işlem gerektirmektedir. Aslında özvektörleri bulmak için verilen [X,D] = eig(A) komutu aynı zamanda öz değerleri de bulduğu için her iki bilgiye aynı anda ulaşma imkanı 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 (köşegenindeki) 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] = 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 :

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

$$r = \begin{array}{l} -3.0000 \\ -2.0000 \\ -1.0000 \end{array}$$

-- 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 çarpım için :

$$\begin{aligned} a &= [1 \ 0 \ -20.6]; \quad b = [1 \ 19.6 \ 151.2] \\ c &= \text{conv}(a,b) \end{aligned}$$

$$c = \begin{array}{l} 1.0e+003 \\ 0.0010 \quad 0.0196 \quad 0.1306 \quad -0.4038 \quad -3.1147 \end{array}$$

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 hesaplamak için polyval(c) komutu kullanılır :

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

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

$$\begin{aligned} \text{ans} &= \\ &86 \end{aligned}$$

-- 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 köşegenindeki elemanlari listelemek için `diag(A)` komutu kullanilir :

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

```
diag(A)
```

```
ans =
```

```
    1  
    5  
    9
```

Köşegenin elemanlari haricindeki matris bileşenleri 0 olarak göstermek istersek :

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

```
ans =
```

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


Köşegen matrisi oluşturmayla alakalı aşağıdaki diğer 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 oluşturmak 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 programi yaziniz. X açi degerinin derece olarak kullanicidan alip sin(x)'i bulunuz ?

$$\text{Sin}(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \dots$$

2- Yerden V_0 hiziyla ve T açisiyla fırlatılan 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 asagida 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 hareketi boyunca yerden yeterince yüksekte olduğunu varsayınız.

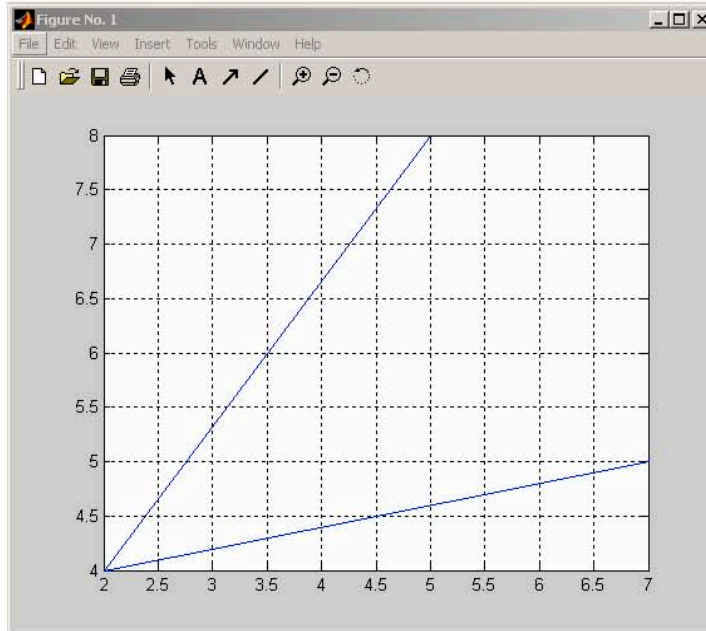
BÖLÜM 2 : GRAFIK ve EGRI ÇIZIMLARI

-- 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.



-- Aşağıda grafik çizimiyle ilgili bazı özellikler sıralanmıştır :

x=3:0.5:10

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

grid

Grafik arka yüzünün ölçekli olarak gösterilmesini sağlar.

title('...')

Çizilen grafiğe başlık yazmak için kullanılır.

xlabel('...')

Çizilen grafiğin x-eksenine istenilen açıklamayı yazmak için kullanılır.

ylabel('...')

Çizilen grafiğin 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

İstenildiği takdirde çizilen eğrinin 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.

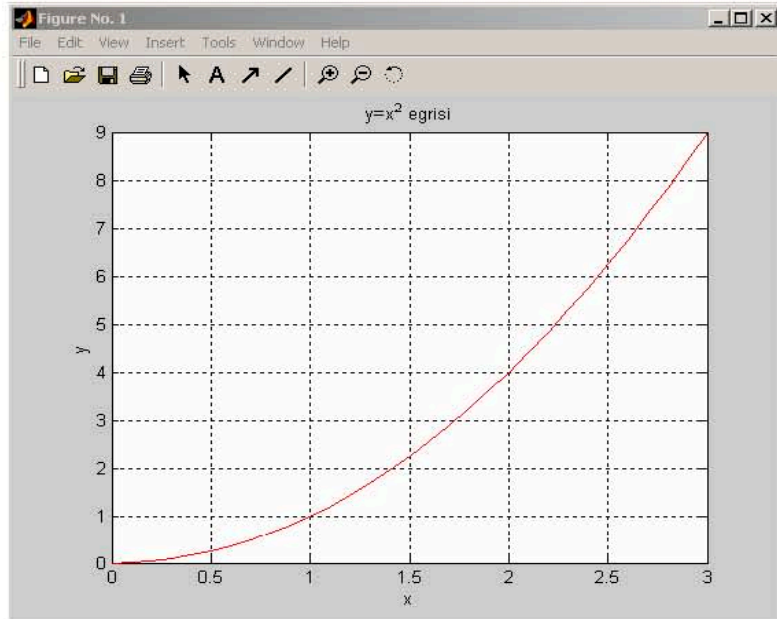
r g b w i

Çizilen eğrinin 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 kısaltımıştır.

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

-- Aşağıdaki örnekte ise $y = x^2$ eğrisini 0 ve 3 aralığında çizdirelim :

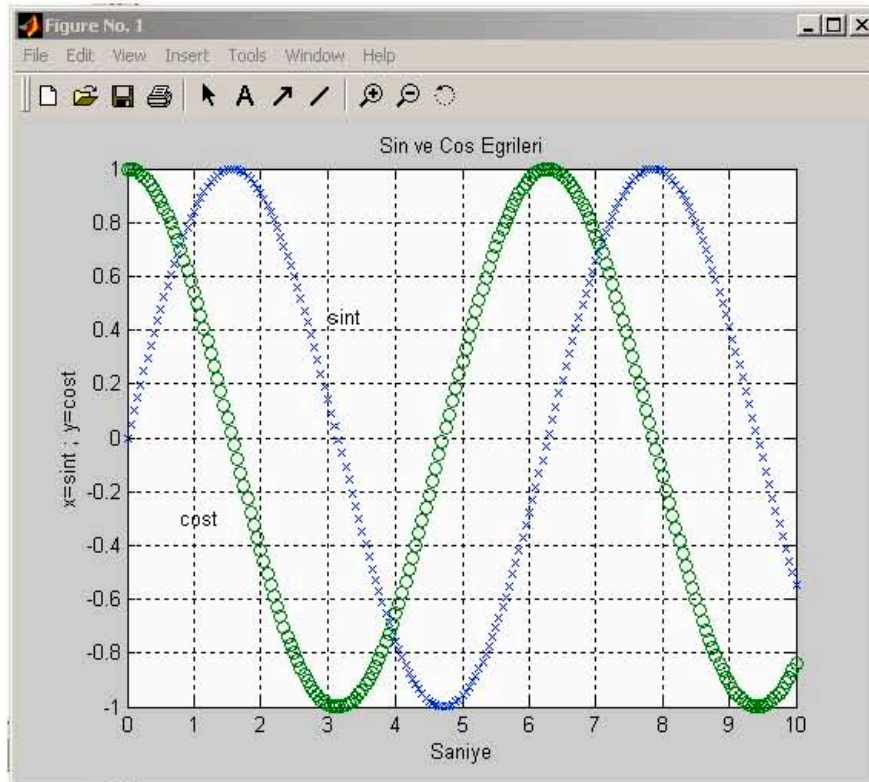
```
x = 0:0.1:3;
y = x.^2;
plot(x,y, 'r');
title('y=x2 eğrisi');
xlabel('x');
grid;
ylabel('y')
```



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

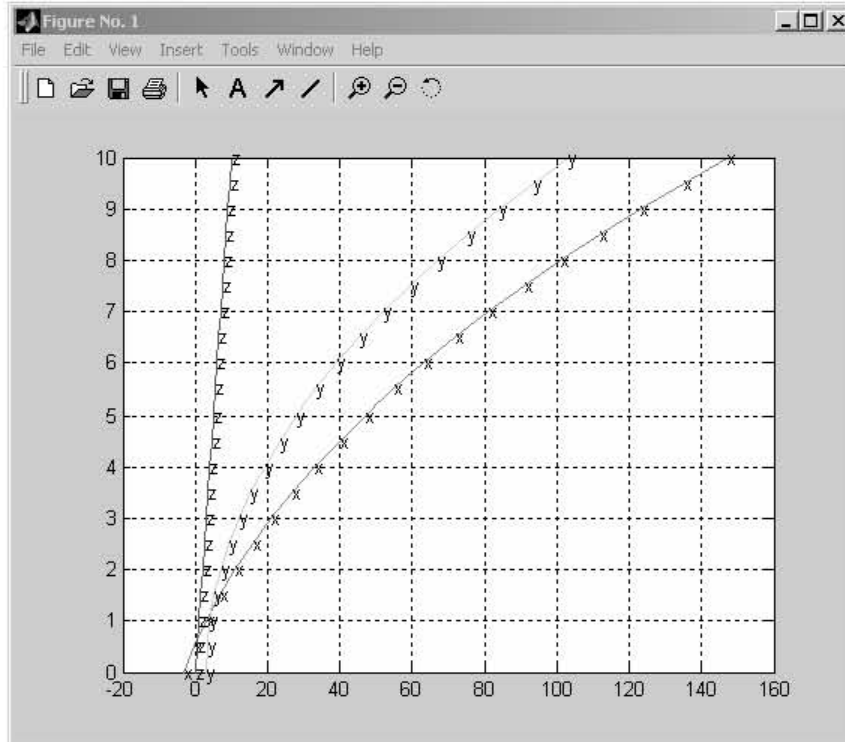
-- Birden fazla egriyi üst üste çizme uygulaması olarak aşağıdaki örnekte $\sin(x)$ ve $\cos(x)$ eğrileri tek bir grafik ekranı üzerinde çizdirilmiştir :

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



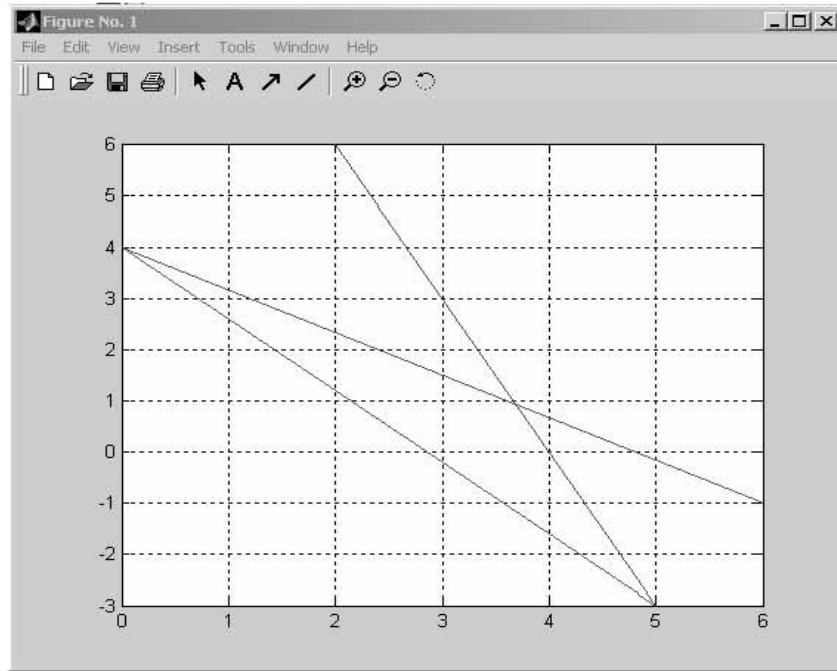
-- Asagidaki örnekte ise 3 farkli egri çizdirilmiştir :

```
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 $\text{plot}(z)$ ifadesi kullanılır. Çizim işleminde ise reel ve imajiner kısımlar ayrı ayrı ikili noktalar olarak kabul edilir :

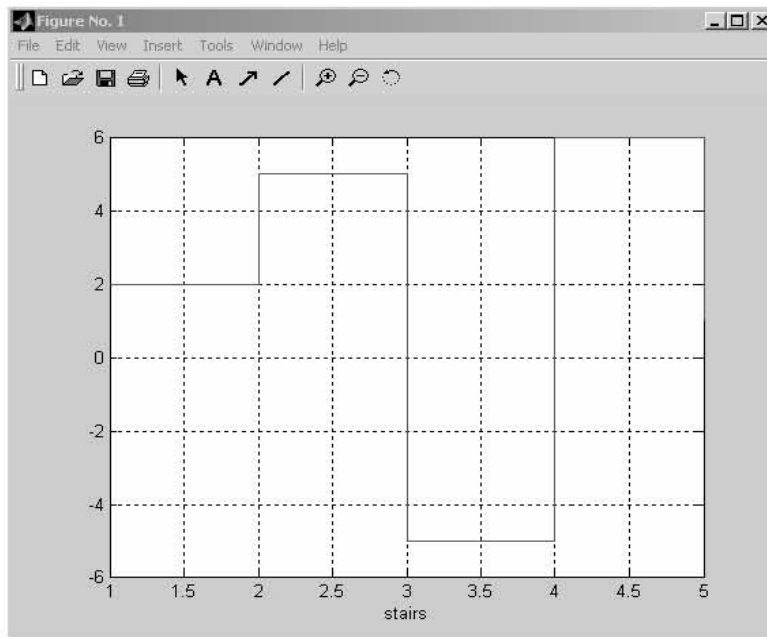
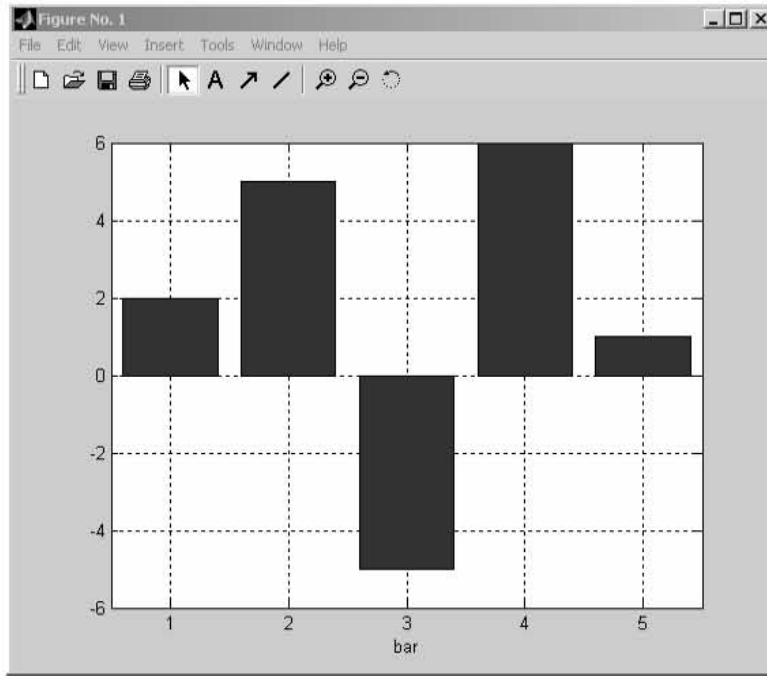
```
C=[2+6i 5-3i 4i 6-i]
plot(C)
grid
```



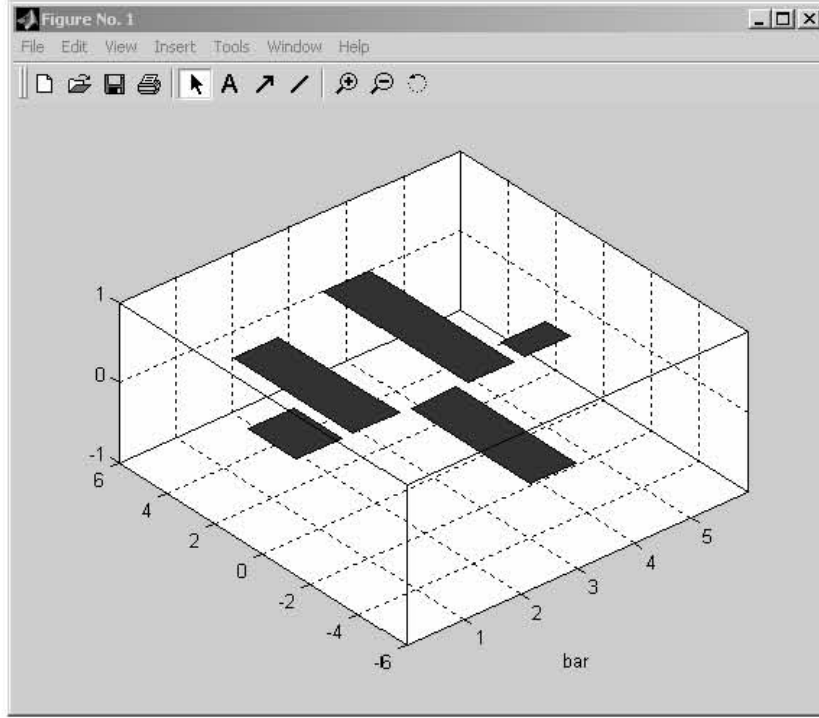
Not : $\text{loglog}(X)$ komutu hem x eksenini hem de y eksenini logaritmik ölçeklendirmeyi kullanarak X'in grafigini çizdirir

-- Bir A vektörünü “ bar grafiklerini ” kullanarak çizdirmek için bar(A) komutu kullanilir. “ Basamak ” fonksiyonu seklinde çizilecek ise stairs(A) komutu kullanilir. Her iki çizime ait örnek grafikler asagida ayrı ayrı verilmistir :

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



-- Ayrica grafik ekranindaki menülerden yararlanarak çeşitli görüntü değişiklikleri yapılabilir. Örnek olarak “ Tools ” menüsünde “ Rotate-3D ” seçeneği kullanılarak mouse yardımıyla iki üstteki “ bar ” grafiğinin görüntüsü aşağıdaki gibi elde edilebilir.

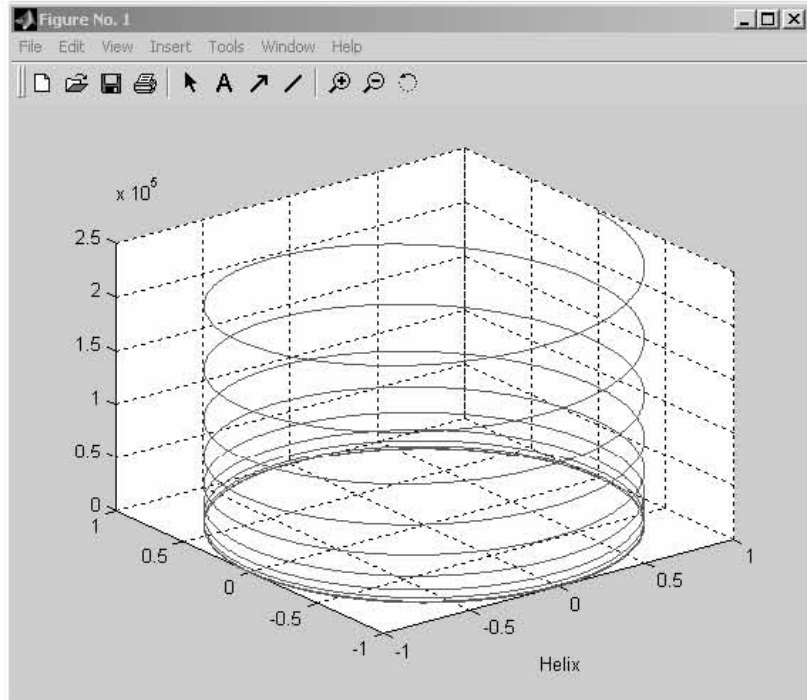


-- Grafik çiziminde grafik çizgi tipleri, işaretler ve renkler aşağıdaki tabloda sıralanmıştır :

Sembol	Renk(RGB)	Çizgi stili	Sembol	Nokta stili	
Y	sari(110)	.	nokta	-	Çizgi
M	magenta(101)	O	yuvarlak	:	Noktali
C	cyan(011)	X	çarpı işareti	-.	çizgili ve noktali
R	kirmizi(100)	+	arti işareti	- -	kesik çizgili
G	yesil(010)	*	yildiz		
B	mavi(001)	S	karekök		
W	beyaz(111)	D	baklava		

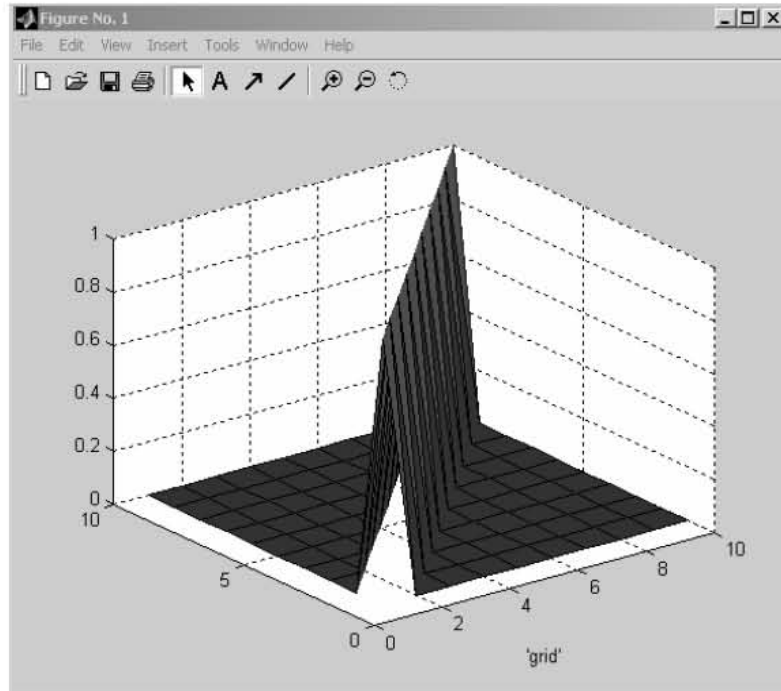
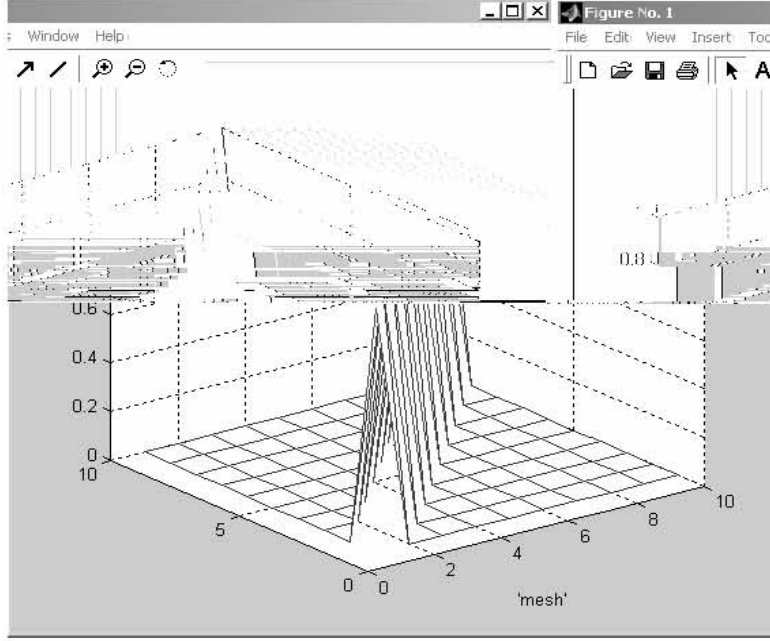
-- '3-D Line' (3 Boyutlu düz çizgi) çizimi için plot3(...) komutu kullanılır .Asagida heliks çizimi programi verilmistir :

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



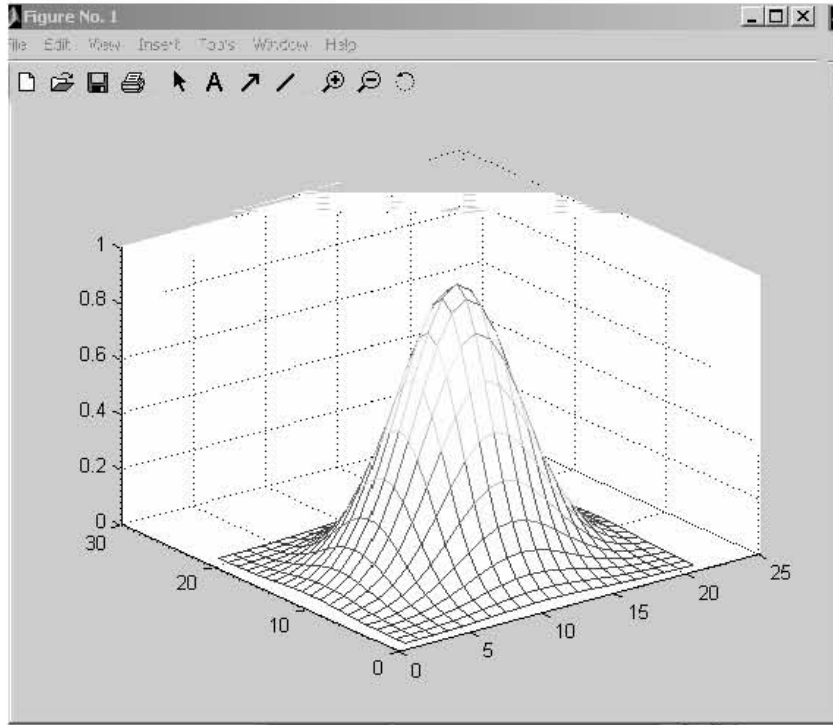
-- 3 boyutlu ag ve yüzey çizimlerinde kullanılan komutlardan biri mesh(...) komutudur. Bu komut verilen girisi z bileşeni olarak algılar ve dikdörtgen x-y düzlemi üzerinde z eksenini boyunca çizim yapar. surf(...) komutu ise aynı işi yüzey olarak yapar. Aşağıdaki komut satırlarının çizim görüntüleri yine alt tarafında verilmiştir.

```
mesh(eye(10));  
grid  
  
surf(eye(10));  
grid
```



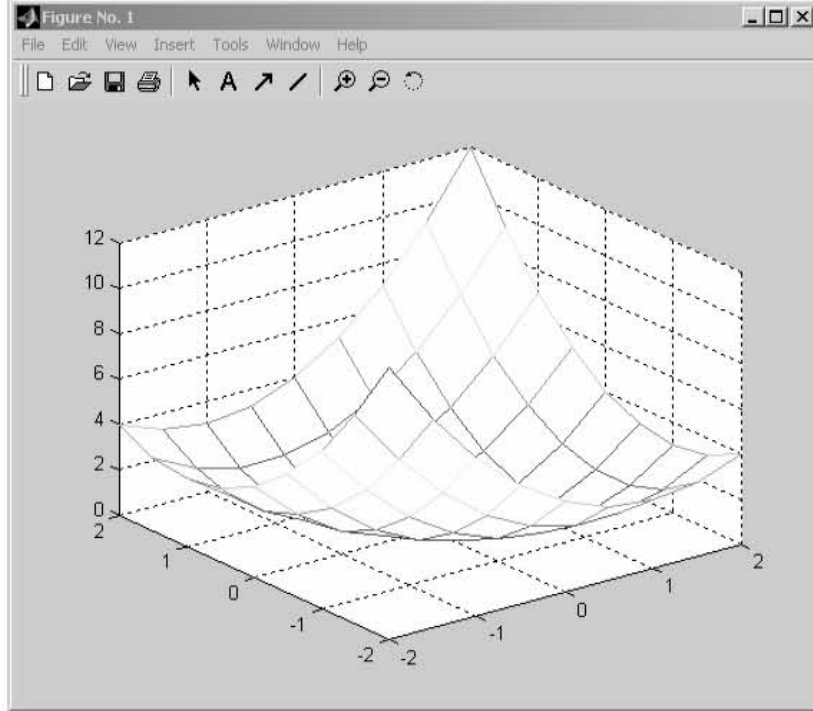
-- $z=\exp(-x^2-y^2)$ fonksiyon yüzeyini $[-2,2] \times [-2,2]$ tanım aralığında 3 boyutlu olarak çizdirelim :

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



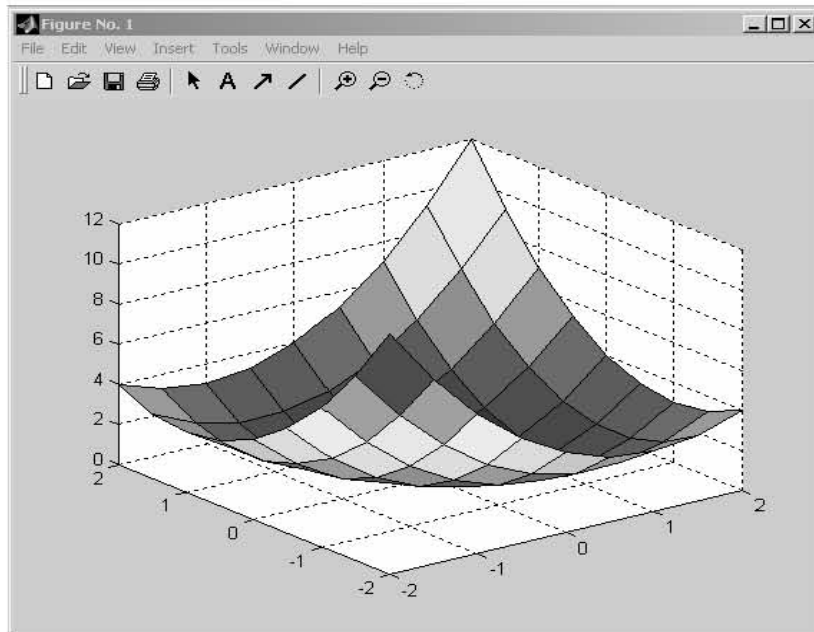
-- Örnek olarak $z=x^2+y^2+xy$ yüzeyini $-2 < x < 2$ ve $-2 < y < 2$ aralığında çizdirelim :

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



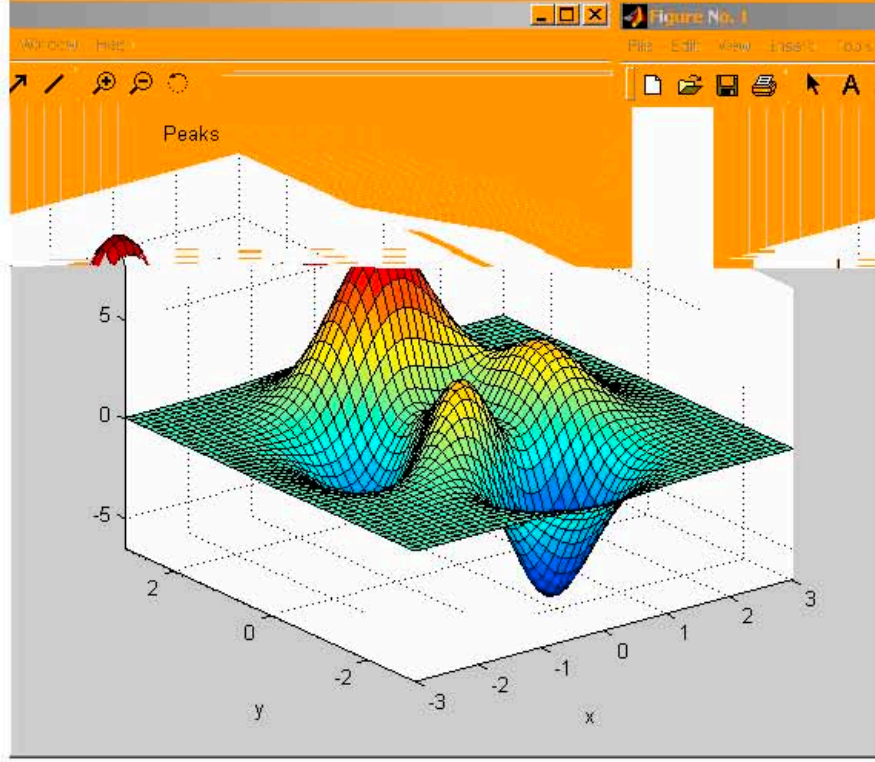
-- Yukarıdaki örnekte çizim fonksiyonu olarak mesh(X,Y,Z) yerine surf(X,Y,Z) çizim fonksiyonu kullanılırsa grafik yüzeyi aynı fakat her bir karesi farklı renklere boyanmış şekilde çizilecektir:

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



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

```
[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 degeriyle $x = (-2,2)$ ve $y = (-2,2)$ araligini kullanarak çizdiriniz.
- 4- $z = e^{-2x} + 4x^3$ grafigini (2,50) araliginda çizdiriniz.
- 5- $x = 9\sin(t)$, $y = 2\tan(3t) + \cos(t)$ grafiklerini (0,10) araliginda 0.05 artimla çizdiriniz.

BÖLÜM 3 : MATLAB ile PROGRAMLAMA

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

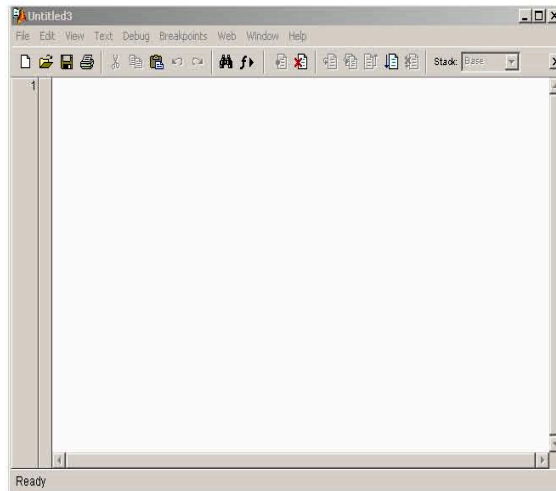
Su ana kadar MATLAB ile yaptığımız uygulamalarda, belirli bir işlemi gerçekleştirmek üzere gerekli olan komut ya da fonksiyonları komut satırından, >> sembollerinden sonra tek tek girerek icra ediyorduk. Oysa, MATLAB komut ya da fonksiyonlarından gerçekleştirmek istediğimiz bir işle ilgili özel bir grubu, 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 içerdiği için tanım itibarıyla bir program dosyasıdır. Dolayısıyla programlama M-dosyaları oluşturularak yapılır.

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

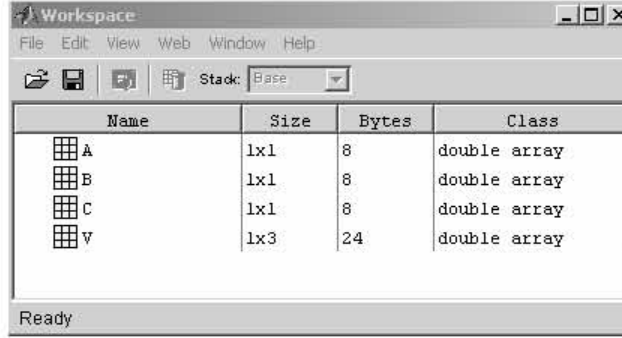
-- “**M-File**” **Olusturma** : Programlama işlemi M-File (Program Dosyası) dosyalarında yapılır. Bu nedenle program yazarken en çok kullanılacak olan işlem M-File oluşturma işlemidir. Bunun için “File” menüsünden “New” seçilir ve daha sonra “M-File” seçeneği seçilir. Böylece yeni bir programlama ekranı 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 uzantisi **.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 gösterilmiştir:



A) GENEL AMAÇLI KOMUTLAR

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



-- “ **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 uzantisi **.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 belleğe yükler.

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

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

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

(Not: Bir programi dogru olarak çalistirmek için, icra ettirmeden önce “clear” komutuyla mevcut degiskenler silinebilir.)

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

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

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

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

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

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

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

B) DEGISKEN ATAMA

“ C ” ve “ PASCAL ” gibi programlama dillerinde, programın ana gövdesinin oluşturulmasına başlamadan hemen önce, programdaki tüm değişkenlerin hangi tip değişken olduklarını belirtmek ve programın bilgisayar belleğinden uygun miktarda alanı bu değişkenler için tahsis etmesini sağlamak gerekir. MATLAB’de değişkenler, kendilerine ait bir isim ve onlara atanacak değerler yardımıyla oluşturulurlar. Önceden değişken tipini belirtmeksizin, değişkene verilen değere bağlı olarak MATLAB, uygun değişken tipini belirler ve bilgisayar belleğinden yeteri kadar yeri bu değişkene tahsis eder.

Birinci bölümde temel atamalarla ilgili bazı bilgiler (sabit, değişken, matris, dizi, vektör tanımlama...) verilmisti.

-- Mantık ve İlişki Operatörleri :

==	Esittir	&	and	Ve
~=	Esit değil	&	and	Ve
<	Küçük	~	not	Değil
<=	Küçük esit			
>	Büyük			
>=	Büyük esit			

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

-- **disp() komutu** : İstenen açıklamayı görüntüler.

```
>> disp('Programlamaya Giriş')
Programlamaya Giriş
```

-- **input() komutu** : Kullanıcıdan klavye aracılığıyla programcı tarafından girilmesi istenen değişken istenir ve ilgili değişkene atanır.

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

-- **fprintf** komutu : Bir açıklama ifadesiyle birlikte bir veya birden fazla değerini görüntülenebilmesini sağlar.

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

Not : “ fprintf ” fonksiyonu, kompleks sayilarin sadece reel kismini gösterir. Bu nedenle kompleks sayi uygulamalarinda “ disp ” fonksiyonunu kullanilmalidir.

“ fprintf ” fonksiyonunda kullanılan çeşitli “ biçim tipleri ” aşağıda gösterilmiştir :

%d : Virgüllü sayıları 10’un kuvvetleri şeklinde gösterir.

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

%e : Sayıyı üstel şekilde gösterir.

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

fonksiyon(ilik_deger , son_deger , eleman_sayisi)

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

-- **Baslangic, son deger ve artis miktari belli dizilerin atanmasi :**

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

--**Hazir fonksiyon özelliklerini kullanarak olusturulan 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 bazı özel sabit veya açıklama değerler :

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

-- **Vektör ve matrislerin tanımlanması :**

```
>> 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 uzunluğunu (eleman sayısını) görüntüler.

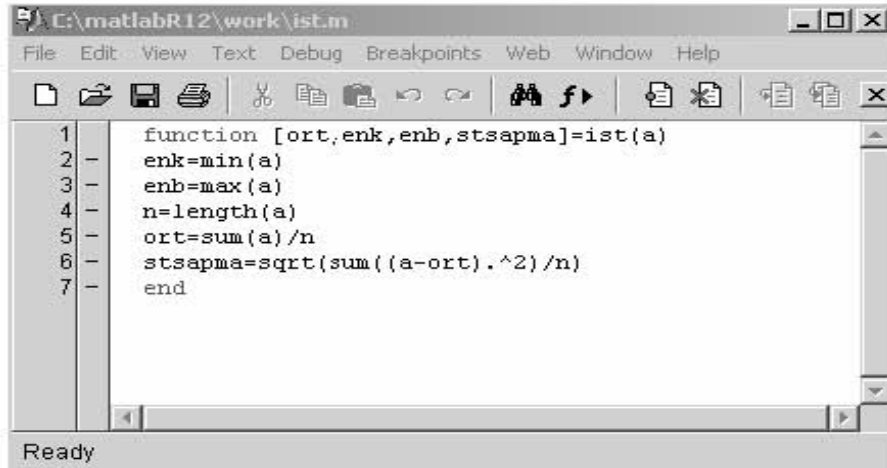
```
>> A=[ 5 8 7 2 5 9 6 ]
A =
     5     8     7     2     5     9     6
>> length(A)
ans =
     7
```

C) FONKSİYON OLUSTURMA VE DİĞER 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 oluşturalım :



The screenshot shows a MATLAB R12 editor window titled 'C:\matlabR12\work\ist.m'. The window contains the following MATLAB code:

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

The status bar at the bottom of the window displays 'Ready'.

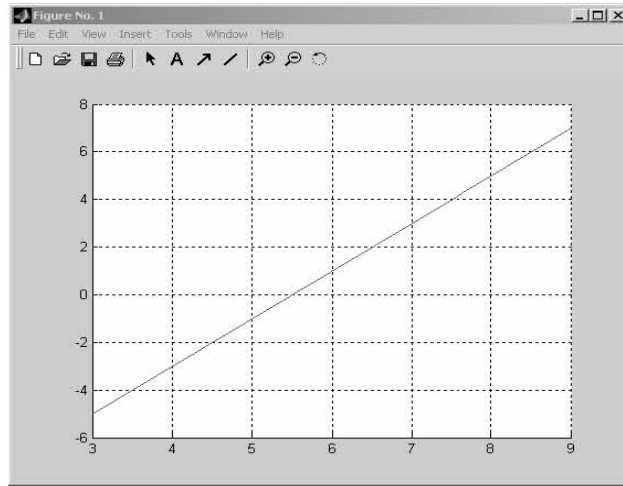
Not: Herhangi bir .m dosyasi penceresinde “ f ” butonuna mouse ile bir defa dokunuldugunda (yukaridaki sekilde de görülmektedir) “work” dizini altinda kayitli olan bütün fonksiyonlar listelenir.

Simdi de programi komut satirindan 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 çalistirilirse asagida gösterildigi gibi icra edilir ve grafik de otomatik olarak yine asagida gösterildigi gibi çizdirilir.

```
>> dogru
*****İki 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 grafiğin çizimi ” programı aşağıda gösterildiği gibi Oy.m dosyası adı altında yazılmıştır. Çizilmesi istenen pasta dilimi grafiği için kullanıcıdan bu üç partiye ait yüzdelik oy oranı rakamı olarak istenmekte ve grafik otomatik olarak çizdirilmektedir.

```
C:\matlabR12\work\Oy.m
File Edit View Text Debug Breakpoints Web Window Help
Stack: Base

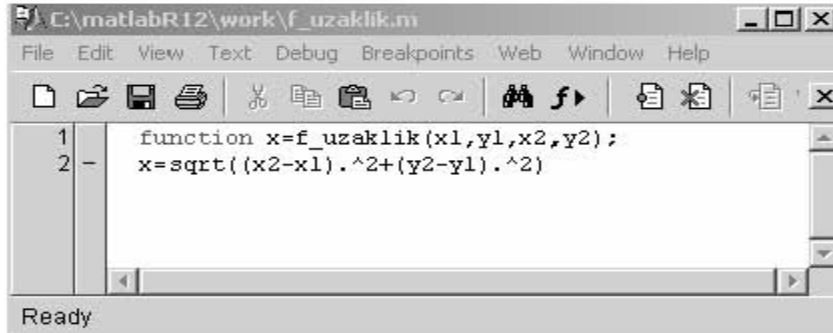
1 - clear
2 - disp('◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇');
3 - disp('Partilerin Oy Oranları Grafiği');
4 - disp('◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇');
5 - A=input('A Partisinin aldığı oyun yüzdelik dilimdeki rakamını giriniz: ');
6 - B=input('B Partisinin aldığı oyun yüzdelik dilimdeki rakamını giriniz: ');
7 - C=input('C Partisinin aldığı oyun yüzdelik dilimdeki rakamını giriniz: ');
8 - V=[A B C];
9 - pie(V);
10 - title('Oy Oranları');
11 - legend('A Partisi','B Partisi','C Partisi')
12
```


Bu program komut satirinda çalistirilirse 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 çağirma islemini uygulayarak icra ettirelim :

a) Fonksiyon yapisini kullanarak :



The screenshot shows a MATLAB editor window titled 'C:\matlabR12\work\f_uzaklik.m'. The window contains the following code:

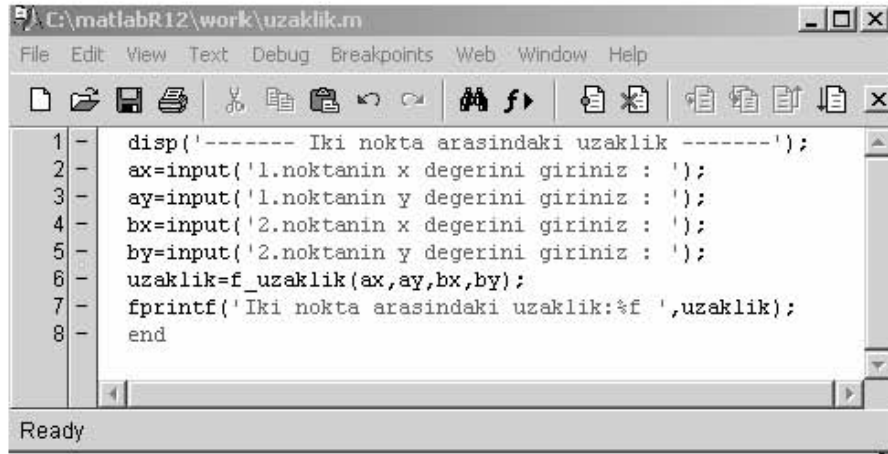
```
1 function x=f_uzaklik(x1,y1,x2,y2);
2 x=sqrt((x2-x1).^2+(y2-y1).^2)
```

The status bar at the bottom of the window displays 'Ready'.

Komut satirindan icra edilirse :

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

b) Fonksiyon yapısını bellekten çağırarak :



```
C:\matlabR12\work\uzaklik.m
File Edit View Text Debug Breakpoints Web Window Help
[Icons]
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
Ready
```

Komut satirından icra edilirse :

```
>> uzaklik
----- Iki 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
Iki nokta arasindaki uzaklik:2.828427
```

D) DÖNGÜ ve SARTLI İFADE UYGULAMALARI :

1.SARTLI İFADELER :

-- “ **if** ” yapısı : “ if ” komutunun MATLAB ’de 3 farklı şekli mevcuttur :

a) **if** koşul
deyim1
deyim 2
deyim_n
end

Koşul doğru ise deyim1, deyim1, ... , deyim_n, ile belirtilen deyimler grubu icra edilir ve programın kontrolü end’i izleyen deyim’e geçer; koşul yanlış ise bu durumda deyim1, deyim2, ..., deyim_n ile belirtilen deyimler grubu icra edilmeden kontrol end’i izleyen deyim’e geçecektir.

```

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

```

Kosul dogru ise deyim1, deyim1, ... , deyim_n, ile belirtilen deyimler grubu icra edilir ve programin kontrolü end'i izleyen deyime geçer; kosul yanlis 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.

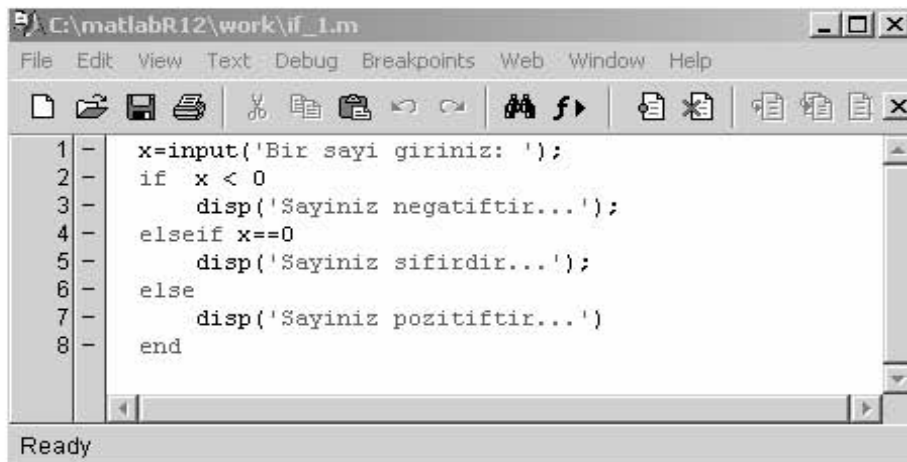
```

c) if kosul1
    deyim1
elseif kosul2
    deyim2
elseif kosul3
    deyim3
    ...
elseif kosul_n
    deyim_n
else
    deyim_n+1
end

```

Bu yapı içerisinde kontrol edilen kosullardan herhangi biri dogru ise onunla ilişkili deyim icra edilir ve kontrol end' i izleyen deyime geçer. Kosullarin hepsi de yanlissa, kontrol else' i izleyen deyim_n+1'e geçer ve bu deyim de icra edildikten sonra kontrol end'i izleyen deyime geçecektir.

Örnek Program: Kullanici tarafından bir sayi istenip bu sayinin pozitif, negatif veya 0 mi oldugunu sorgulayan ve ekrana yazdiran program asagida incelenebilir :



```

C:\matlabR12\work\if_1.m
File Edit View Text Debug Breakpoints Web Window Help
[Icons]
1 - x=input('Bir sayi giriniz: ');
2 - if x < 0
3 -     disp('Sayiniz negatiftir...');
4 - elseif x==0
5 -     disp('Sayiniz sifirdir...');
6 - else
7 -     disp('Sayiniz pozitifdir...')
8 - end
Ready

```

Komut satirindan icra edilirse :

```
Bir sayi giriniz: 55
Sayiniz pozitiftir...
>> if_1
Bir sayi giriniz: -9
Sayiniz negatiftir...
>> if_1
Bir sayi giriniz: 0
Sayiniz sifirdir...
```

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

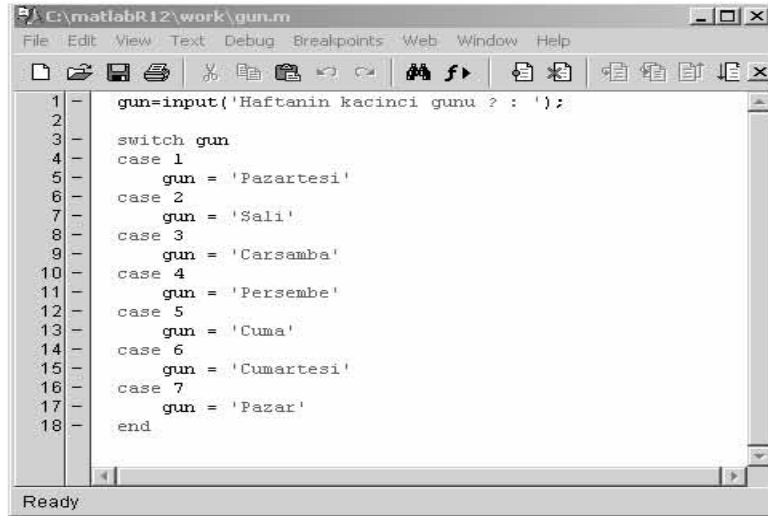
Ifadenin degeri kontrol edilir ve buna göre farkli bir deyim veya deyimler grubu icra edilir. Örneğin, ifadenin degeri deger1 ise, deyim1 icra edilir ve kontrol end’i izleyen deyime geçer.

Ifadenin degeri , deger2, deger3 ya da deger4 ’e esitse bu durumd deyim2 icra edilir ve kontrol end’i izleyen deyime geçer.

Ifadenin degeri, case’lerle kontrol edilen degerlerden hiç birine esit degilse bu durumda da otherwise sözcüğünü izleyen deyim_n+1 icr edilecektir.

```
switch(ifade)
case deger1
    deyim1
case deger2,deger3,deger4
    deyim2
...
case deger_n
    deyim_n
otherwise
    deyim_n+1
end
```

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



```
1 - gun=input('Haftanın kacinci 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
```

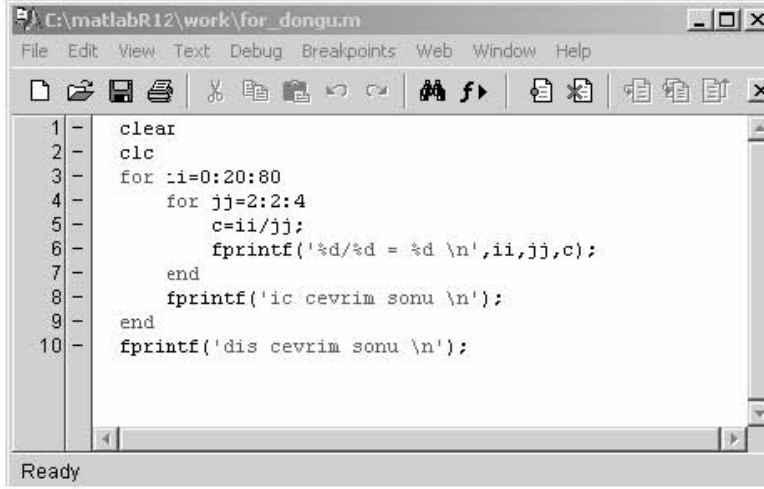
```
>> gun
Haftanın kacinci gunu ? : 5
gun =
    Cuma
```

2.DÖNGÜ İFADELERİ :

-- “ **for** ” **döngüsü** : Parametre degeri baslangiç degerinden baslayarak ve her seferinde artim degeri kadar arttirilerek son degere erisene kadar degistirilir. Parametrenin her degeri için, deyim1, deyim2, deyim_n seklinde belirtilen ve for-end sözcükleri arasında yer alan deyimler grubu icra edilir. Parametrenin degeri son degeri asinca, programin kontrolü end’i izleyen deyime yani çevrimin disina çikacaktır.

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

Örnek Program: Arka arkaya bölme işlemlerinin yapıldığı aşağıdaki for döngüsü programını inceleyebilir :



```
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');
```

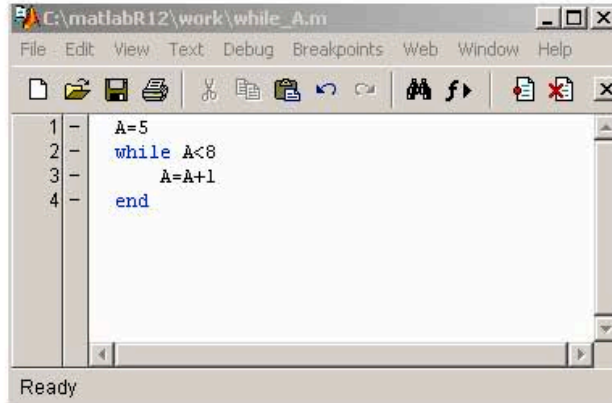
Komut satirindan icra edilirse :

```
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 sınıra kadar istenilen işlemleri tekrarlayarak yapar. Kosul doğru olduğu sürece, deyim1, deyim2, ... , deyim_n şeklinde belirtilen deyimler grubunu icra eder. Kosul yanlış olduğu anda, end’i izleyen deyime yani çevrim dışına çıkar.

```
while kosul
    deyim1
    deyim2
    ...
    deyim_n
end
```

Örnek Program: A=5 ilk degerinden baslayarak A<8 oldugu sürece A'ya 1 eklemek için asagidaki program incelenebilir :



```
C:\matlabR12\work\while_A.m
File Edit View Text Debug Breakpoints Web Window Help
[Icons]
1 - A=5
2 - while A<8
3 -     A=A+1
4 - end
Ready
```

Komut satirindan icra ettirilirse :

```
A =
    5
A =
    6
A =
    7
A =
    8
```

ALISTIRMALAR

1- Kenarlari kullanıcı tarafından istenen bir üçgenin çeşitkenar, ikizkenar veya eşkenar 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 başlayarak 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 t anında otomobilin hızını veren bir MATLAB programı yazınız.

4- Kullanıcıdan doğum gününü soran ve bu kullanıcının kaç yıl, kaç ay ve kaç yıl yaşadığını bulan bir MATLAB programı yazınız.

BÖLÜM 4 : KONTROL SİSTEMLERİNDE ZAMAN-FREKANS ANALIZI

Matematiksel modeller lineer sistemlere veya başka sistemlere MATLAB komutları vasıtasıyla kolaylıkla dönüştürülebilir.

Aşağıda kontrol sistemleri için gerekli bazı dönüşümler açıklanmıştır :

-- Transfer fonksiyonundan durum uzayına çevirme :

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

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

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

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

-- Durum uzayından transfer fonksiyonuna çevirme :

$$[\text{num}, \text{den}] = \text{ss2tf}(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$$

Eğer sistemin birden fazla girişi varsa aşağıdaki komut kullanılır :

$$[\text{num}, \text{den}] = \text{ss2tf}(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, i, u)$$

Örnek olarak aşağıda verilen iki girişli bir sistemin diferansiyel denklem takımının MATLAB komut satırındaki işlemleri 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.

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

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

```
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 katsayılarla :

$$\frac{Y(s)}{U1(s)} = \frac{1}{3s^2 - 2}$$

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

$$\frac{B(s)}{A(s)} = \frac{r_1}{s-p(1)} + \frac{r_2}{s-p(2)} + \dots + \frac{r(n)}{s-p(n)} + k(s)$$

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 ayrilmistir :

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

-- Sürekli zaman sistemleri için geçici cevap analizi :

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

Basamak fonksiyonu (Step function):

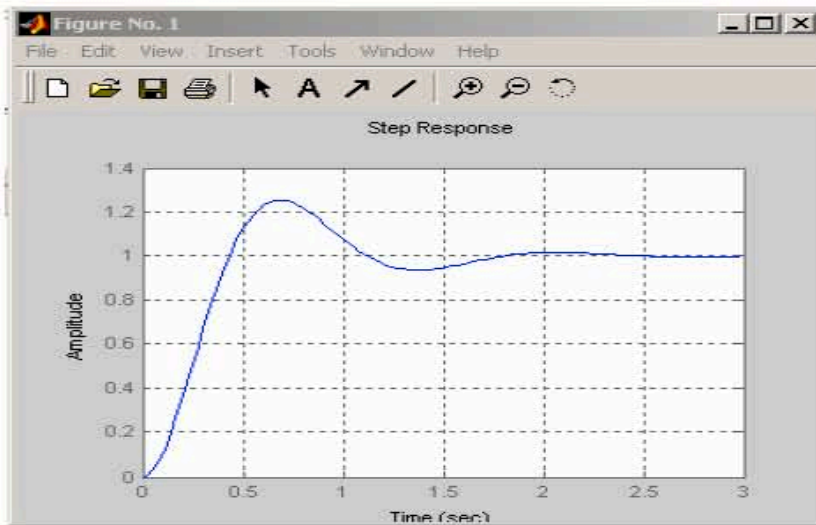
`step(num,den)` `step(num,den,t)`

Durum uzayinda 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 verecegi cevap analizi asagida gösterilmistir.

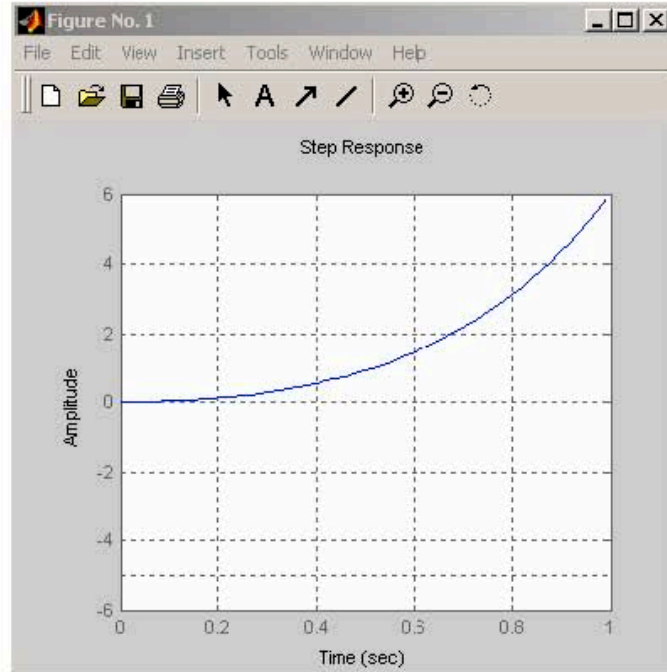
```
C:\matlabR12\work\step1.m
File Edit View Text Debug Breakpoints Web Window Help
[Icons]
1 % Birim basamak (step function) fonksiyonu cevabi %
2 clc
3 clear all
4 num=[0 0 25];
5 den=[1 4 25];
6 step(num,den)
7 grid
8 title('Birim Basamak Cevabi');
9 end
Ready
```



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
[Icons] Stack: X
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
```



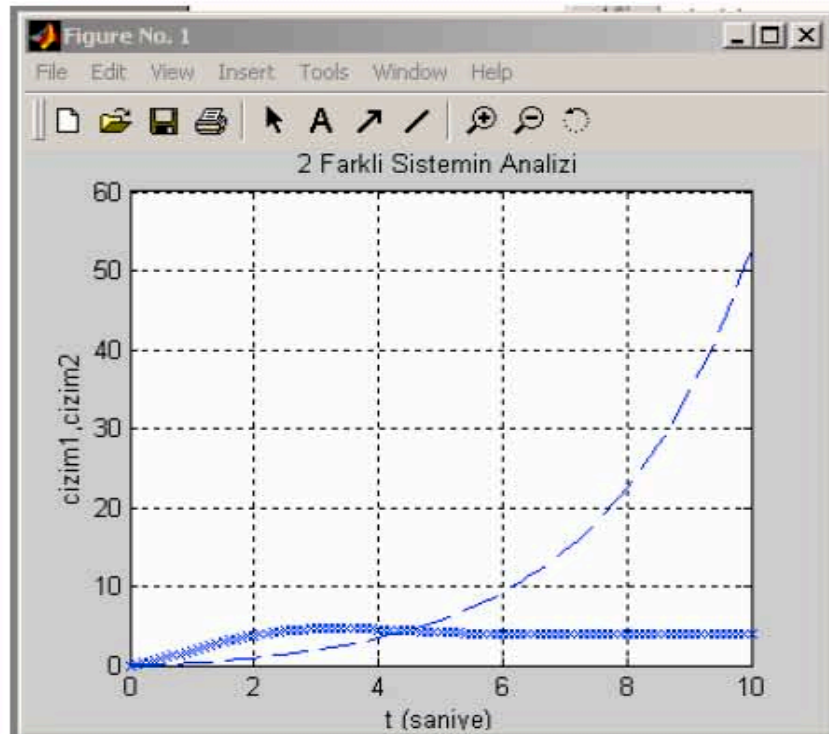
Bir sistemin anlık darbe fonksiyonu (Impulse Function) ve rampa fonksiyonuna (Ramp Function) vereceği cevabı bulmak için de bu transfer fonksiyonlarının önlerine sırasıyla $1/s$ ve $1/s^2$ çarpanı oluşturulur ve normal basamak fonksiyonu gibi çizilir.

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.

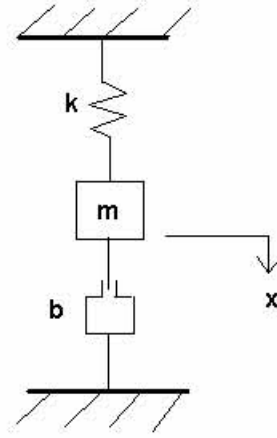
$num1 = [0 \ 0 \ 1]$; $den1 = [1 \ 2 \ -1]$

$num2 = [0 \ 1 \ 4]$; $den2 = [1 \ 1 \ 1]$

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



Örnek:



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

(Başlangıç koşulları : $x(0)=0.1$ 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 denklemleri :

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

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

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

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

$$X(s) = \frac{mx(0)s + m x^{(1)}(0) + bx(0)}{ms^2 + bs + k} = \frac{0.1s + 0.35}{s^2 + 3s + 2}$$

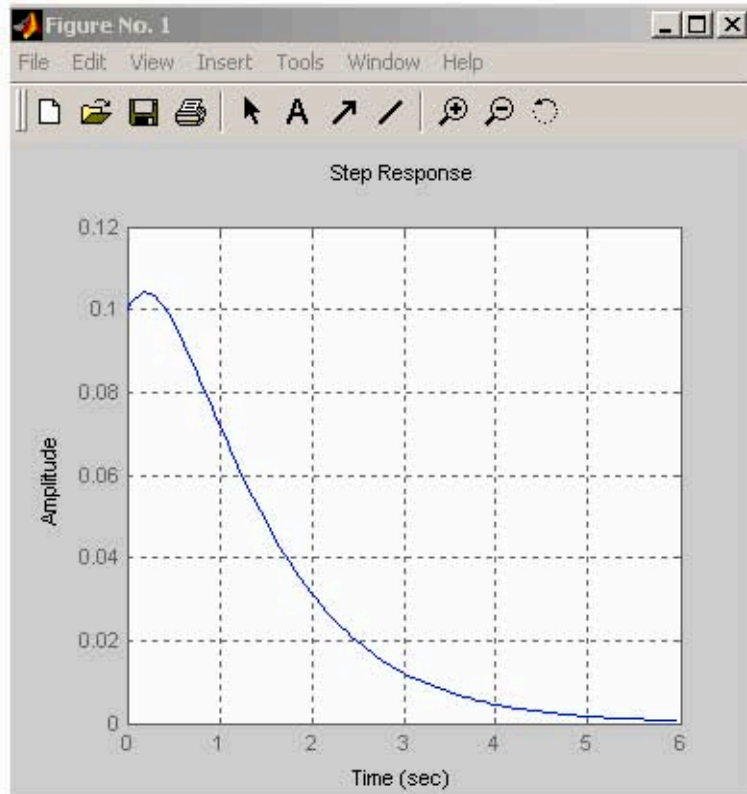
$X(s)$ ifadesi $\frac{1}{s}$ çarpanıyla yeniden düzenlenirse :

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

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

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

```
C:\matlabR12\work\sis1.m
File Edit View Text Debug Breakpoints Web Window Help
[Icons]
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 ilişkin 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.

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

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

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

EK(Tablo):MATLAB KOMUTLARI ve MATRIS FONKSIYONLARI TABLOSU

Komut	Görevi
abs	Mutlak Deger, kompleks büyüklük
angle	Faz Açisi
ans	Islem sonucu açıklaması olmadigi zamanki sonuç ifadesi (cevap)
atan	Arktanjan
axis	Eksen ölçekleme
bode	Bode diyagramini çizdirir
clear	Önceden tanımlanmış degiskenleri bellekten siler
clc	Komut satirini temizler
clf	Grafik ekranini temizler
computer	Kullanılan bilgisayar hakkında bilgi verir
conj	Kompleks konjüge
conv	Konvolüsyon
corrcoef	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 kapatir
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 grafiği kaldırir
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 cevabi çizimi
ones	Sabit 1 veya 1 matrisi üretme
pi	Pi sayısı
plot	Lineer x-y çizimi
polar	Kutupsal çizim
poly	Karakteristik polinom
polyval	Polinomda deger hesaplama
polyvalm	Matris polinomu hesabi
quit	Programi sonlandırır
rand	Rastgele sayılar ve matrisler üretir
rank	Matris ranki
real	Gerçel kısım
residue	Parçali kesir pay kısmi katsayıları
roots	Polinom kökleri
sign	Signum Fonksiyonu
sin	Sinüs
sinh	Hiperbolik sinüs
size	Satir ve sütun boyutları
sqrt	Karekök
sqrtn	Matris kökü
std	Standart sapma
step	Birim basamak fonksiyonu çizimi
sum	Eleman toplami
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	Sifir veya sifir 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 Uygulamaları ”, Prof.Dr. Mithat Uysal, Prof.Dr. Aysenur Uysal, Beta Yayinlari, 2000
- [5] “ www.mathworks.com ”
- [6] “ <http://www.math.siu.edu/matlab/tutorial3.pdf> ”