

## DENEY 1: Matlab’de Temel Uygulamalar

---

### I. AMAÇ

Bu deneyde MATLAB (MATrix LABoratory) programının temel özellikleri anlatılmakta, öğrencinin sinyal işleme ve haberleşme uygulamalarında kullanabilmesi için gerekli altyapının oluşması hedeflenmektedir.

### II. ÖN HAZIRLIK

1) **rand, randn, figure, plot, subplot, hold on, stem, ones, zeros, eye, diag, mean, fix, floor, ceil, round, rem, close, clear, clc, lookfor, whos, linspace, logspace** komutlarının kullanımını komut satırına `help <komut_ismi>` yazarak öğrenin.

2) Matris çarpımı (\*) ve nokta çarpımı (.\* ) arasındaki farkı inceleyiniz.

3) Deneyde kullanılacak tüm komutları öğrenerek geliniz.

### III. ÖN BİLGİ

MATLAB programında temelde üç farklı çalışma penceresi vardır:

- Komut penceresi (command window)
- Grafik penceresi (figure)
- Editör penceresi

Komut penceresi ekranı kullanılarak basit komutlar yazılabilir. Grafik penceresi çizdirilen grafiklerin görüntülediği penceredir. Editör penceresi ile ise “m-file” olarak adlandırılan ve MATLAB kodlarını içeren dosya oluşturulabilir.

MATLAB’de “help” komutu kullanılarak birçok bilgiye ulaşmak mümkündür. Komut penceresine `help <komut_ismi>` yazılarak istenilen komut hakkında özet bilgiye ulaşılabilir. Daha detaylı bilgi için `doc <komut_ismi>` kullanılabilir.

#### A) Aritmetik işlemler

MATLAB her değişkeni matris olarak algılayıp işlem yapmaktadır. Bu özellik bize çoğu yerde kolaylık sağlamaktadır.

**MATLAB’de dört farklı aritmetik işlem ( +, -, \*, / ) bulunmaktadır. Bu işlemlerin matris işlemleri olduğu unutulmamalıdır. Bunun yanında matris elemanları arasında birebir işlemler yapabilmek için ( .\*, ./, .^ ) operatörleri kullanılmalıdır. Yani,**

$x = [x_1, x_2, \dots, x_n]$      $y = [y_1, y_2, \dots, y_n]$     için;  
 $x.*y = [x_1y_1, x_2y_2, \dots, x_ny_n]$     olur.

#### B) Kompleks Sayılar

Matlab karmaşık sayıları da desteklemektedir. İmajiner kısım  $i$  ya da  $j$  ile temsil edilmektedir.

» $z=3 + 4i$	% 4ten sonra * işaretine ihtiyaç yoktur.
» $\text{conj}(z)$	% $z'$ in konjugesini hesaplar.
» $\text{angle}(z)$	% $z'$ in faz açısını hesaplar.
» $\text{real}(z)$	% gerçel kısmı hesaplar.
» $\text{imag}(z)$	% imajiner kısmı hesaplar.
» $\text{abs}(z)$	% $z'$ in mutlak değerini hesaplar.

### C) Dizi Oluşturma

Matlab'de tüm dizilerin indisleri birden başlamaktadır.  $x(1)$  dizinin ilk elemanını temsil eder. Herhangi bir dizi tanımlamak için C dilindeki gibi öncesinden bir ön tanımlama yapmak gerekmemektedir.  $x = [1 \ 3 \ 5 \ 7]$ ; şeklinde yazılarak istenilen dizi oluşturulabilir. Bununla birlikte dizi oluşturmak için farklı notasyonlar da bulunmaktadır.  $X = 1:2:7$  yazıldığı takdirde birden yediye kadar ikili adımlarla ilerleyen bir dizi oluşturulmaktadır.

### D) Bellek Ataması

zeros, ones komutları kullanılarak bellek ataması yapılabilir. Böylece işlemlerin daha hızlı olmasını sağlanır.

» $A = \text{zeros}(3,5);$
» $B = \text{ones}(10,2);$

### E) Özel Karakterler ve Fonksiyonlar

Matlab'de sıkça karşılaşılan bazı özel karakterler ve fonksiyonlar aşağıdaki gibidir.

Pi	: $\pi$ (3.14...)
NaN	: sayı değil (not-a-number). 0/0 gibi belirsizlikleri ifade eder.
Inf	: $+\infty$
%	: yorum yazabilmek için gerekli işarettir.
'	: matrisin transpozunu alır.
sqrt	: karekök almayı sağlar.
length	: vektörün uzunluğunu gösterir.
size	: matris boyutlarını verir.
find	: sıfır olmayan vektör elemanlarının indislerini gösterir.
rand	: 0-1 arasında rasgele sayı üretir.
exp	: üstel fonksiyon

### F) Kontrol Komutları

if, switch, while, for gibi komutlar ile matlab'de kontrol akışı sağlanır. **for ve while döngü komutlarıdır.** Döngü için belirtilen kriterler sağlandığı sürece belli bir komut dizisinin tekrar ederek çalışmasını sağlar. **if ve switch ise durum kontrol komutlarıdır.** Farklı durumlar için karşılaştırmalar yaparak hangi kodun çalıştırılacağına karar verirler.

FOR döngüsü:
» $X=0;$

```

» for i=1:10           % i değişkeni birer kez artar ve 10 olduğunda döngü sonlanır.
»     X=X+1;           % her seferinde X in değerini bir artırır.
» end
IF:
» x = -3;
» if x > 0
»     str='positive';
» elseif x < 0
»     str='negative';
» elseif x == 0
»     str='zero';
» else
»     str='error';
» end

```

Kodun açıklanması:

x = -3 değeri sıfırdan küçük olduğundan dolayı “str” değişkeni “negative” değerini alır.

WHILE döngüsü:

```

» x = -10;
» while x < 0
»     x = x+2;
»     if x == -2
»         break;
»     end
» end

```

Kodun açıklanması:

x değeri -10 ile başlatılır ve ikişer artırılarak -2'ye kadar ilerler. -2 olduğunda ise “break” komutu ile döngü sonlandırılır.

#### Karşılaştırma Operatörleri:

```

<=   küçük eşitse
<    küçükse
>=   büyük eşitse
>    büyükse
==   eşitse
~=   eşit değilse

```

#### Lojik Operatörler

```

&    VE
|    VEYA
~    DEĞİL

```

### G) Değişken Kaydetme

MATLAB’de “save” komutu ile değişkenler kaydedilir. Bu değişkenler “load” komutu ile tekrar geri çağırılabilir. Uzun bir işlem yükü sonucu oluşan çıktılar kaydedilerek, daha sonra o çıktıların ulaşmak için zaman kaybedilmemiş olur.

```

» x = sqrt(16/3);
» y = floor(x);
» save(var.mat','x');
» clear
» load var.mat

```

## H) Çizim İşlemleri

Matlabda en basit grafik çizdirme komutu **plot** komutudur. **Plot** komutu, iki boyutlu doğru (çizgi) grafiği çizdirir. Örneğin X ile Y, iki aynı boyutlu vektör ve X'deki sayılar x-ekseni (absis) üzerinde Y'deki sayılar y-ekseni (ordinat) üzerinde olsun. **Plot** komutu X in her noktasına karşılık gelen Y değerlerini çizdirir. Diğer bir deyişle, ( X(1),Y(1)), (X(2),Y(2)), (X(3),Y(3) ) vb. noktalar çizdirilir ve daha sonra tüm bu noktalar birleştirilir. **Plot** komutunun nasıl çalıştığını bir örnek üzerinde görelim. Önce iki basit vektör oluşturalım:

```

» x_nok = [1 2 3 4 5];
» y_nok = [25 0 20 5 15];

```

Daha sonra bu iki vektörü çizdirmek için komut satırına şu komutu yazalım:

» <b>plot(x_nok, y_nok)</b>	% eksenler lineer ölçeklenirler.
» <b>semilogx(x, y)</b>	% x eksenini 10 tabanında logaritmik ölçeklidir.
» <b>semilogy(x, y)</b>	% y eksenini 10 tabanında logaritmik ölçeklidir.
» <b>loglog(x, y)</b>	% Her iki eksen de 10 tabanında logaritmik ölçeklidir.

<b>xlabel ('text') :</b> Grafiğin x-eksenini adlandırır. Genelde text, veri adı ve/veya birimini gösterir. <b>ylabel ('text') :</b> Grafiğin y-eksenini adlandırır. Genelde text, veri adı ve/veya birimini gösterir. <b>title ('text') :</b> Grafiğe başlık verir.
---

Bazen özellikle çizgi grafiklerde grafiğin daha anlaşılır olması için, diğer bir deyişle okunabilirliğini artırmak ve görünümünü daha belirgin yapmak için kılavuz çizgileri (yani alt zemin ızgarası) eklemek gerekebilir. Bu işlem için komut satırına

» **grid on** (veya sadece **grid** yazılır)

Aynı düzlem üzerinde ve aynı eksen takımını kullanarak tek bir grafik penceresinde birden fazla grafiği (çizimi) göstermek için **subplot** komutu kullanılır. Genel formatı:

» **subplot(m,n,p)**

şeklinde. Burada *m* satır, *n* sütun sayısını gösterir ve *m x n* 'lik matris düzeninde grafik çizilebilir. Bu komut dikdörtgen şeklinde *m\*n* grafik alanı oluşturur. *p* ise 1 ile *m\*n* arasında olmalıdır. *p*, satırda soldan sağa doğru sütunda yukarıdan aşağı doğru belirleme yapar yani grafikler bu sıraya göre yerleştirilir.

<b>Örnek:</b> » figure(1) » x=-pi:pi/10:pi; » y1=cos(x); » plot(x,y1,'r+'); » xlabel('x');	» subplot(2,1,1), » plot(x,y2,'b--'); » xlabel('x'); » ylabel('y'); » title('y =sinx Grafiği'); » subplot(2,1,2),
---	--

<pre> » ylabel('y'); » title('y =cosx Grafiği'); » figure(2) » y2=sin(x); </pre>	<pre> » plot(x,y2,'r*'); » xlabel('x'); » ylabel('y'); » title('y =sinx Grafiği'); </pre>
--	---

#### IV. DENEYİN YAPILIŞI

!!! Deneyde yazılacak tüm kodların MATLAB'ın editör penceresinde düzenli bir şekilde yazılması ve kayıt edilmesi beklenmektedir. Aksi belirtilmediği takdirde komut satırında işlem yapılmayacaktır.

- 1) Bu kısımda matrislerde toplama, çıkarma, çarpma, iç çarpım (dot product) işlemleri yapılacaktır. Bunun dışında `sqrt`, `abs`, `sum`, `mean`, `floor`, `ceil`, `round`, `zeros`, `ones`, `size` komutları incelenecektir.

$$A = \begin{pmatrix} 1 & 2 \\ 4 & 5 \\ 7 & 8 \end{pmatrix} \quad B = \begin{pmatrix} 1 & 3 \\ 5 & 7 \\ 9 & 11 \end{pmatrix}$$

$k = 10$ ;

- $A+B=?$
  - $k*(A-B)=?$
  - A matrisi ile B matrisinin transpozunu çarpınız.
  - A ve B matrisleri arasında iç çarpım yapınız.
  - A matrisinin satırlarının ortalamasını A\_m vektörüne yazdırınız.
  - B matrisinin sütunlarının toplamını B\_s dizisine atayınız.
  - A matrisinin elemanlarını tüm eleman değerleri '1' olan X matrisi ile toplayınız.
  - A\_m vektörünün elemanlarını tam sayı olacak şekilde yuvarlayınız.
  - A\_m vektörünün elemanlarını yukarı yuvarlayınız.
  - A\_m vektörünün elemanlarını aşağı yuvarlayınız.
  - A ve B' matrislerinin çarpımı sonucu elde edilen matrisin boyutlarını bulunuz.
- 2)  $\text{lin} = [1 \ 2 \ \dots \ 20]$  vektörünü oluşturunuz. Ardından bu vektörü  $\text{lin} = [1 \ -2 \ 3 \ -4 \ \dots \ -20]$  biçimine (çift elemanlar negatif) dönüştürünüz.
- 3) Eleman değerleri 11-30 arasında olan  $10 \times 3$ 'lük bir x matrisi oluşturunuz. Ardından x matrisinin 20'den büyük değerleri için 1, diğerleri içinse 0 değerini alan bir y matrisi oluşturunuz. (İpucu: **rand** komutunun kullanımını inceleyiniz)
- 4)  $3a + 6b + 4c = 1$   
 $a + 5b = 2$   
 $7b + 7c = 3$

şeklinde verilen denklemi matris halinde tanımlayarak a, b, c değişkenlerini hesaplayınız.

- 5) Eleman değerleri 1’den 21’e kadar birer atlayarak ilerleyen bir **x** dizisi oluşturunuz.  **$y = e^x$**  olmak üzere **y** dizisini hesaplayarak çizdiriniz. Aynı pencereye y eksenini logaritmik olacak biçimde ikinci bir grafik daha çizdiriniz. Şekillerin başlık ve eksen isimlerini ekleyiniz.
- 6) **a** ve **b** şıklarında verilen fonksiyonları iki ayrı figure’de **subplot** komutu kullanarak çizdiriniz. **x**’i **linspace** komutu ile  $[-\pi, \pi]$  aralığında 100 noktali oluşturunuz

a)  $y = 10x$  ve  $y = x^2$

b)  $\cos(x)$  ve  $\exp(-5x) \cdot \sin(2x)$

