ضمیمه و راهنما : دستورات متلب

- **ماتریس**

تعریف

```
a = [ 1 2 3 4 5; 2 3 4 5 6; 3 4 5 6 7; 4 5 6 7 8]
-----
a =
    1    2    3    4    5
    2    3    4    5    6
    3    4    5    6    7
    4    5    6    7    8
```

انتخاب عدد در ردیف m م و ستون n م

```
a(2,5)
-----
ans =  6
```

انتخاب یک ستون مشخص

```
v = a(:,4)
-----
v =
    4
    5
    6
    7
```

انتخاب چند ستون از ستون mم تا ستون nم

```
a(:,m:n)
a(:, 2:3)
-----
ans =
    2    3
    3    4
    4    5
    5    6
```

انتخاب بخشی از ماتریس

```
sa = a(2:3,2:4)
-----
sa =
    3    4    5
    4    5    6
```

حذف ستون یا ردیف خاص

```
a(: , 5)=[]
-----
a =
    1    2    3    4
    2    3    4    5
    3    4    5    6
    4    5    6    7
```

محاسبه معکوس ماتریس

```
a = [ 1 2 3; 2 3 4; 1 2 5]
inv(a)
-----
ans =
  -3.5000   2.0000   0.5000
   3.0000  -1.0000  -1.0000
  -0.5000        0   0.5000
```

جمع و تفریق ماتریس ها

```
a = [ 1 2 3 ; 4 5 6; 7 8 9];
b = [ 7 5 6 ; 2 0 8; 5 7 1];
c = a + b
d = a – b
-----
c =
    8    7    9
    6    5   14
   12   15   10
d =
   -6   -3   -3
    2    5   -2
    2    1    8
```

ضرب ماتریس ها

```
a = [ 10 12 23 ; 14 8 6; 27 8 9];
b = 2;
e = a * b
-----
e =
   20   24   46
   28   16   12
   54   16   18
```

محاسبه ترانهاده

```
a = [ 10 12 23 ; 14 8 6; 27 8 9]
b = a'
-----
a =
   10   12   23
   14    8    6
   27    8    9
b =
   10   14   27
   12    8    8
   23    6    9
```

محاسبه دترمینان

```
a = [ 1 2 3; 2 3 4; 1 2 5]
det(a)
-----
ans = -2
```

ادغام ماتریس ها

```
a = [ 10 12 23 ; 14 8 6; 27 8 9]
b = [ 12 31 45 ; 8 0 -9; 45 2 11]
c = [a, b]
d = [a; b]
-----
a =
    10    12    23
    14     8     6
    27     8     9
b =
    12    31    45
     8     0    -9
    45     2    11
c =
    10    12    23    12    31    45
    14     8     6     8     0    -9
    27     8     9    45     2    11
d =
    10    12    23
    14     8     6
    27     8     9
    12    31    45
     8     0    -9
    45     2    11
```

رنک ماتریس

```
a = [ 10 12 23 ; 14 8 6; 27 8 9];
n=rank(a)
-----
n =
   3
```

بردار ها (v) و مقادیر ویژه ماتریس (d)

```
a = [ 10 12 23 ; 14 8 6; 27 8 9];
[v,d]=eig(a)
------
v =
  -0.6457   -0.6997    0.1674
  -0.3972    0.2376   -0.8951
  -0.6521    0.6738    0.4132


d =
  40.6089         0         0
        0  -16.2202         0
        0         0    2.6113
```

مقادیر ویژه عناصر روی قطر اصلی ماتریس d و بردار های ویژه ی متناظر هرکدام ستون های ماتریس v است

- **ماتریس های خاص**

ماتریس صفر

```
zeros(5)
-----
ans =
    0    0    0    0    0
    0    0    0    0    0
    0    0    0    0    0
    0    0    0    0    0
    0    0    0    0    0
```

ماتریس یک

```
ones(4,3)
-----
ans =
    1    0    0    0
    0    1    0    0
    0    0    1    0
    0    0    0    1
```

ماتریس واحد

```
eye(4)
-----
ans =
    1    0    0    0
    0    1    0    0
    0    0    1    0
    0    0    0    1
```

ماتریس رندوم (با اعداد اتفاقی)

```
rand(3, 5)
-----
ans =
  0.8147   0.9134   0.2785   0.9649   0.9572
  0.9058   0.6324   0.5469   0.1576   0.4854
  0.1270   0.0975   0.9575   0.9706   0.8003
```

ماتریس جادویی (مجموع اعداد هر ردیف با مجموع اعداد هر ستون و قطر برابر است)

```
magic(4)
-----
ans =
  16    2    3   13
   5   11   10    8
   9    7    6   12
   4   14   15    1
```

- **ماتریس قطری**

Example :

```
v = [2 1 -1 -2 -5];
D = diag(v)
-----
D = 5×5

     2     0     0     0     0
     0     1     0     0     0
     0     0    -1     0     0
     0     0     0    -2     0
     0     0     0     0    -5
```

Example :

```
v = [2 1 -1 -2 -5];
D1 = diag(v,1)
-----
D1 = 6×6

     0     2     0     0     0     0
     0     0     1     0     0     0
     0     0     0    -1     0     0
     0     0     0     0    -2     0
     0     0     0     0     0    -5
     0     0     0     0     0     0
```

- **جملات شرطی**

```
if <expression 1>
  % Executes when the expression 1 is true
  <statement(s)>

elseif <expression 2>
  % Executes when the boolean expression 2 is true
  <statement(s)>

else
  %  executes when the none of the above condition is true
  <statement(s)>
End
```

Example :

```
a = 100;
  if a == 10
    % if condition is true then print the following
    fprintf('Value of a is 10\n' );
  elseif( a == 20 )
    % if else if condition is true
    fprintf('Value of a is 20\n' );
  elseif a == 30
    % if else if condition is true
    fprintf('Value of a is 30\n' );
  else
    % if none of the conditions is true '
    fprintf('None of the values are matching\n');
  fprintf('Exact value of a is: %d\n', a );
  end
```

```
-----
None of the values are matching
Exact value of a is: 100
```

- **حلقه ها**

حلقه ی for

```
for index = values
  <program statements>
        ...
End
```

Example :

```
for a = 10:
  fprintf('value of a: %d\n', a);
end
-----
value of a: 10
value of a: 11
value of a: 12
value of a: 13
value of a: 14
value of a: 15
value of a: 16
value of a: 17
value of a: 18
```

Example :

```
for a = 1.0: -0.1: 0.0
  disp(a)
end
-----
1
0.90000
0.80000
0.70000
0.60000
0.50000
0.40000
0.30000
0.20000
0.10000
0
```

خروج از حلقه با دستور break

```
for a=10:100
  fprintf('value of a: %d\n', a);
  a = a + 1;
    if( a > 12)
      % terminate the loop using break statement
      break;
    end
end
-----
value of a: 10
value of a: 11
value of a: 12
```

حلقه while

```
while <expression>
  <statements>
End
```

Example :

```
a = 10;
% while loop execution
while( a < 20 )
  fprintf('value of a: %d\n', a);
  a = a + 1;
end
```

```
-----
value of a: 10
value of a: 11
value of a: 12
value of a: 13
value of a: 14
value of a: 15
value of a: 16
value of a: 17
value of a: 18
value of a: 19
```

حلقه های تو در تو

Example :

```
for i = 2:50
  for j = 2:50
    if(~mod(i,j))
      break; % if factor found, not prime
    end
  end
  if(j > (i/j))
    fprintf('%d is prime\n', i);
  end
end
```

```
-----
2 is prime
3 is prime
5 is prime
7 is prime
11 is prime
13 is prime
17 is prime
19 is prime
23 is prime
29 is prime
31 is prime
37 is prime
41 is prime
43 is prime
47 is prime
```

دستورات جبری

حل معادلات خطی

```
solve(equation, variable)
```

Example :

```
solve('v-u-3*t^2=0', 'v')
-----
ans =
   3*t^2 + u
```

حل معادلات با درجات بالا

```
eq = 'x^4 - 7*x^3 + 3*x^2 - 5*x + 9 = 0';
s = solve(eq);
% converting the roots to double type
disp('Numeric value of first root'), disp(double(s(1)));
disp('Numeric value of second root'), disp(double(s(2)));
disp('Numeric value of third root'), disp(double(s(3)));
disp('Numeric value of fourth root'), disp(double(s(4)));
-----
Numeric value of first root
   6.6304
Numeric value of second root
   1.0598
Numeric value of third root
   -0.3451 - 1.0778i
Numeric value of fourth root
   -0.3451 + 1.0778i
```

روش دیگر تعریف $x^4 - 7x^3 + 3x^2 - 5x + 9 = 0$: نوشتن ضرایب از بزرگترین توان تا کمترین در قالب آرایه

```
eq = [1, -7,  3, -5, 9];
```

حل دستگاه مختصات خطی

$$5x + 9y = 5$$

$$3x - 6y = 4$$

```
s = solve('5*x + 9*y = 5','3*x - 6*y = 4');
s.x
s.y
-----
ans =
   22/19
ans =
   -5/57
```

نوشتن معادلات و تعریف متغیرها

بست دادن معادلات

```
% define symbolic variables
x = sym ('x');
y = sym ('y');
z = sym ('z');

% expanding equations
expand((x-5)*(x+9))
expand((x+2)*(x-3)*(x-5)*(x+7))
-----
ans =
-45.0+x^2+(4.0)*x

ans =
210.0+x^4-(43.0)*x^2+x^3+(23.0)*x
```

تعریف متغیرها

فاکتورگیری و ساده سازی

```
syms x
syms y
factor(x^3 - y^3)
factor([x^2-y^2,x^3+y^3])
simplify((x^4-16)/(x^2-4))
-----
ans =
   (x - y)*(x^2 + x*y + y^2)
ans =
   [ (x - y)*(x + y), (x + y)*(x^2 - x*y + y^2)]
ans =
   x^2 + 4
```

محاسبه حد تابع

```
syms x
limit((x^3 + 5)/(x^4 + 7))
-----
ans =
   5/7
```

تعریف تابع تبدیل با استفاده از چند جمله ای

```
a=tf([5 6 1],[0 1])

a=
   s
------------
s^2 + 6 s + 5
```

تعریف تابع تبدیل با متغیر s

```
syms s;
a=s/(s*s+6*s+5)
-----
a=
s/(s^2 + 6*s + 5)
```

تعریف تابع تبدیل با قطب ها، صفرها و گین

```
h = zpk(0, [1-i 1+i 2], -2);
-----
h =

          -2 s
  --------------------
  (s-2) (s^2 - 2s + 2)
```

محاسبه لاپلاس تابع

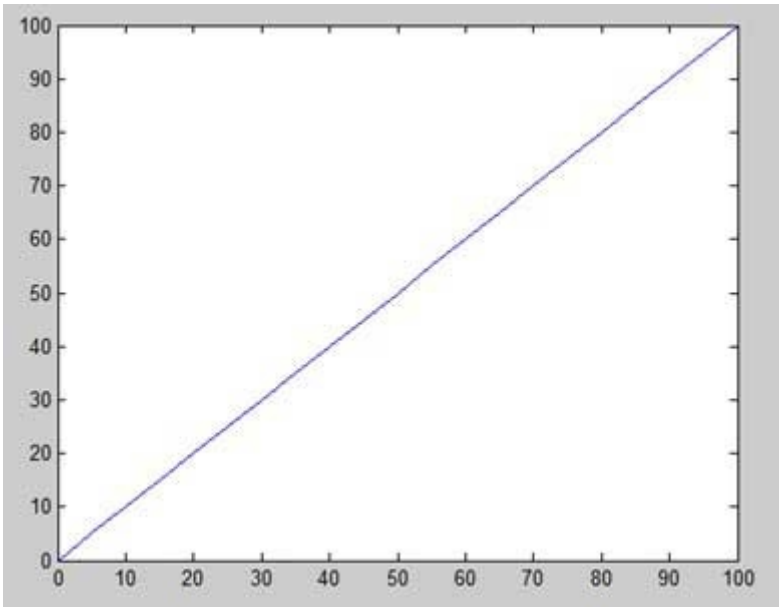
```
syms s t a b w

laplace(a)
laplace(t^2)
laplace(exp(-b*t))
laplace(sin(w*t))
-----
ans =
   1/s^2

ans =
   2/s^3

ans =
   1/(b + s)

ans =
   w/(s^2 + w^2)
```

لاپلاس معکوس

```
syms s t
ilaplace(1/s^3)
-----
ans =
   t^2/2
```

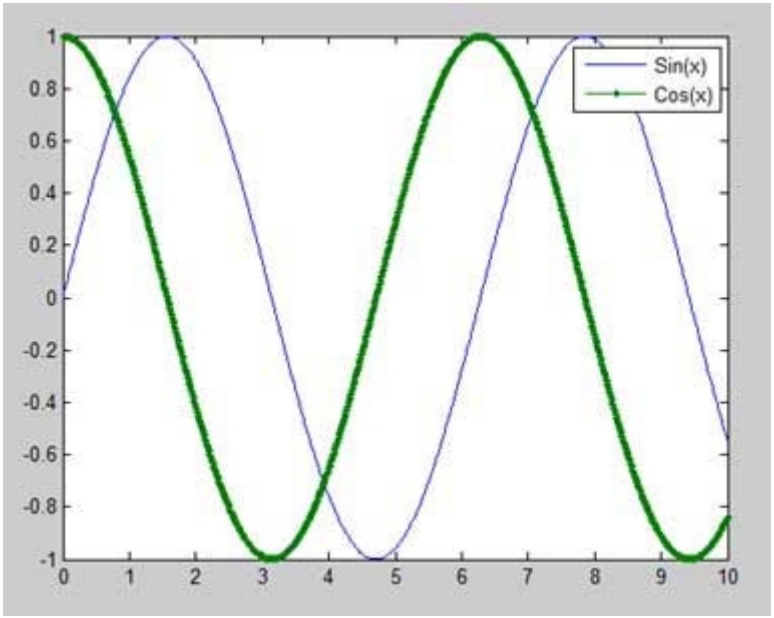
- **رسم نمودار**

```
x = [0:5:100];
y = x;
plot(x, y)
-----
```



رسم دو نمودار همزمان و نام گذاری

```
x = [0 : 0.01: 10];
y = sin(x);
g = cos(x);
plot(x, y, x, g),
legend('Sin(x)', 'Cos(x)')
-----
```



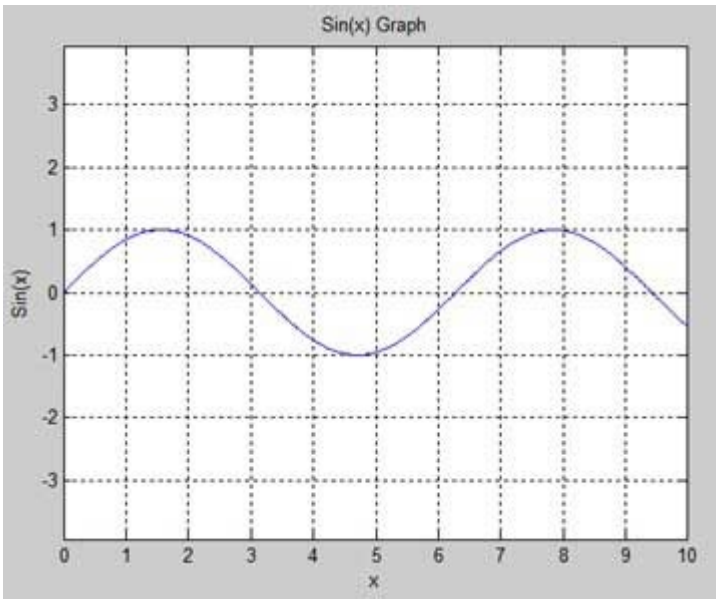
اضافه کردن عنوان (title)

اضافه کردن عنوان محورها (xlabel – ylabel)

تقسیم بندی نمودار (grid on)

نمودار مربعی (axis equal)

```
x = [0:0.01:10];
y = sin(x);
plot(x, y),
xlabel('x'),
ylabel('Sin(x)'),
title('Sin(x) Graph'),
grid onو
axis equal
-----
```



تعریف محدوده محورها

axis ([xmin xmax ymin ymax])

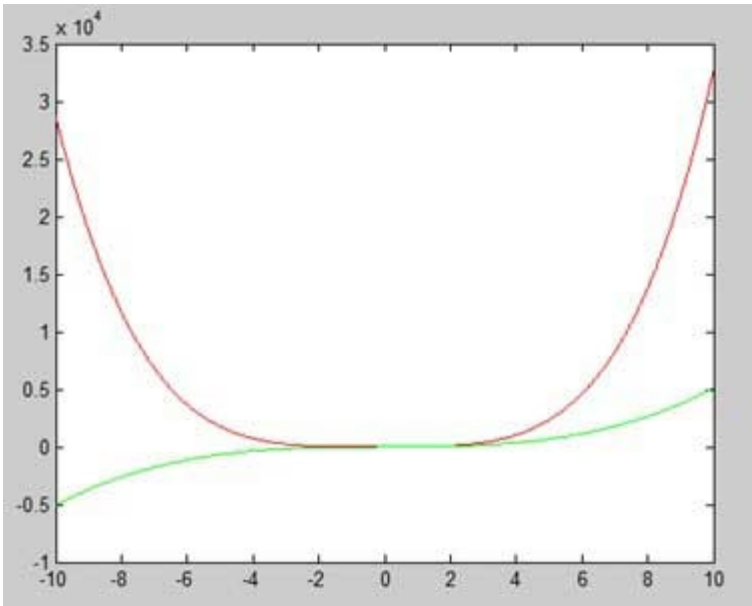
Example :

```
x = [0 : 0.01: 10];
y = exp(-x).* sin(2*x + 3);
plot(x, y), axis([0 10 -1 1])
```

انتخاب رنگ نمودار

Code	Color
w	White
k	Black
b	Blue
r	Red
c	Cyan
g	Green
m	Magenta
y	Yellow

```
x = [-10 : 0.01: 10];
y = 3*x.^4 + 2 * x.^3 + 7 * x.^2 + 2 * x + 9;
g = 5 * x.^3 + 9 * x + 2;
plot(x, y, 'r', x, g, 'g')
-----
```



رسم همزمان چند نمودار با دستور hold on

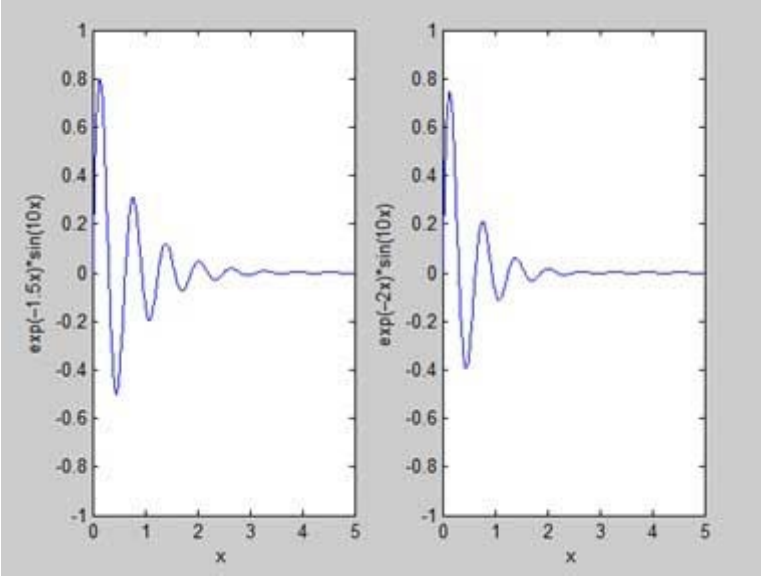
```
x = [0 : 0.01: 10];
y = sin(x);
g = cos(x);
plot(x, y),
hold on,
plot(x, y),
legend('Sin(x)', 'Cos(x)')
```

رسم چند نمودار باهم در m ردیف n تایی

```
subplot(m, n, p)
```

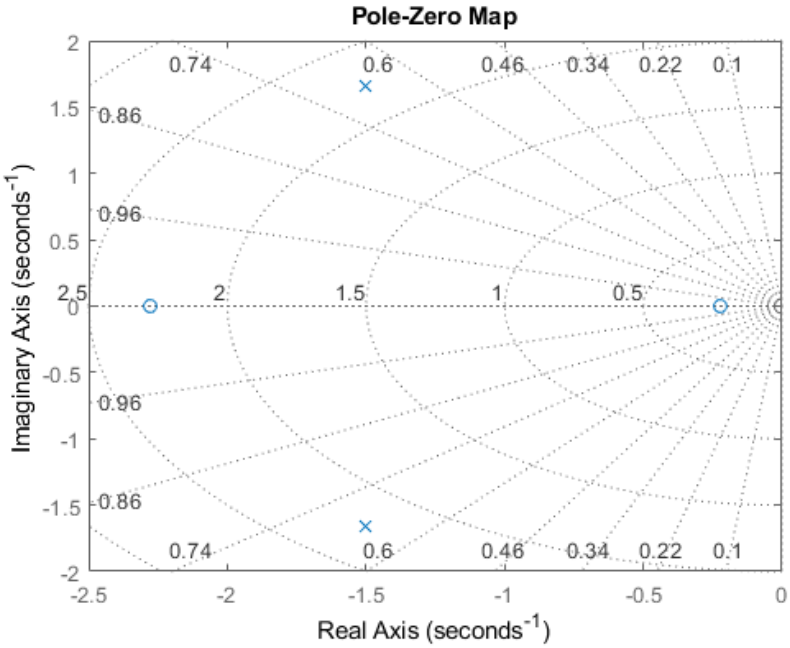
p نشان دهنده ی جایگاه هر نمودار بین بقیه است

```
x = [0:0.01:5];
y = exp(-1.5*x).*sin(10*x);
subplot(1,2,1)
plot(x,y), xlabel('x'),ylabel('exp(–1.5x)*sin(10x)'),axis([0 5 -1 1])
y = exp(-2*x).*sin(10*x);
subplot(1,2,2)
plot(x,y),xlabel('x'),ylabel('exp(–2x)*sin(10x)'),axis([0 5 -1 1])
-----
```



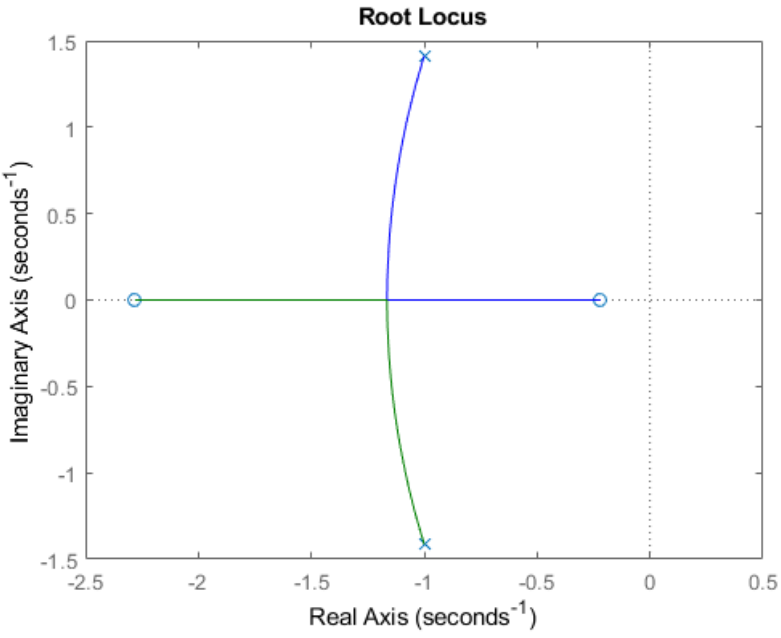
رسم مکان هندسی قطب ها و صفرها

```
H = tf([2 5 1],[1 3 5]);
pzmap(H)
grid on
-----
```



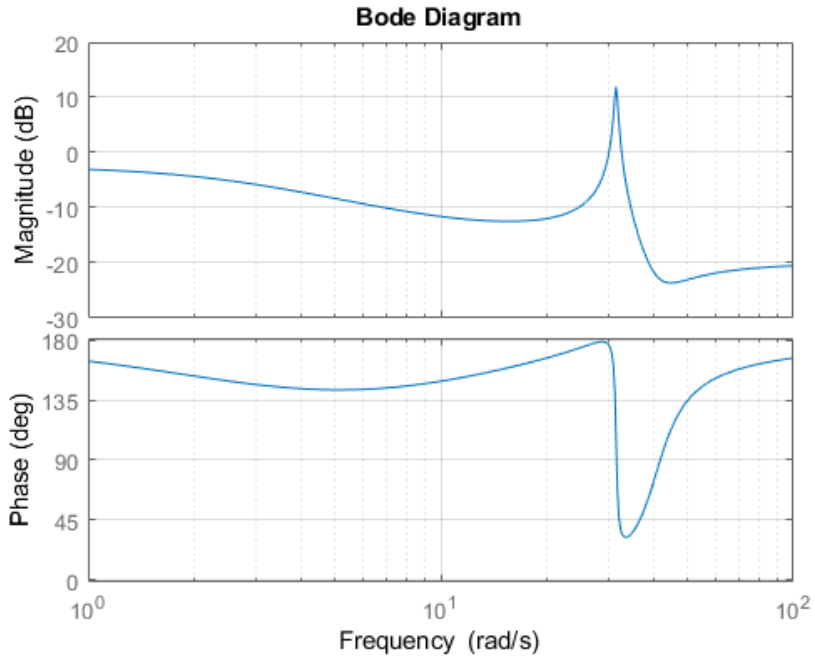
رسم نمودار Root Locus

```
h = tf([2 5 1],[1 2 3]);
rlocus(h)
-----
```



رسم نمودار بود

```
H = tf([-0.1,-2.4,-181,-1950],[1,3.3,990,2600]);
bode(H,{1,100})
grid on
------
```



گسسته سازی (با روش های zoh – foh – impulse – tustin – matched – least squares)

```
sysd = c2d(sys,Ts,method)
```

Example :

```
H = tf([1 -1],[1 4 5],'InputDelay', 0.3);
Hd = c2d(H,0.1,'tustin')
-----
Hd =
            0.03918 z^2 - 0.004124 z - 0.0433
  z^(-3) * ---------------------------------
                 z^2 - 1.629 z + 0.6701
```

تعریف سیستم با ماتریس های فضای حالت

```
A = blkdiag([-0.1 0.4; -0.4 -0.1],[-1 5; -5 -1]);
B = [1; zeros(3,1)];
C = [1 0 1 0];
D = 0;

sys = idss(A,B,C,D)
```

تبدیل فضای حالت به تابع تبدیل متناظر

```
A = blkdiag([-0.1 0.4; -0.4 -0.1],[-1 5; -5 -1]);
B = [1; zeros(3,1)];
C = [1 0 1 0];
D = 0;
[a,b]=ss2tf(A,B,C,D);
H=tf(a,b)
-----
H =

         s^3 + 2.1 s^2 + 26.2 s + 2.6
  ---------------------------------------
  s^4 + 2.2 s^3 + 26.57 s^2 + 5.54 s + 4.42
```

تبدیل تابع تبدیل به ماتریس های فضای حالت

```
a = [1 2 3];
b = [1 0.4 1];
[A,B,C,D] = tf2ss(a,b)
-----
A =
   -0.4000   -1.0000
    1.0000         0

B =
     1
     0

C =
    1.6000    2.0000


D =
     1
```

محاسبه ماتریس مشاهده پذیری

```
a=[1 1;4 -2];
c=[1 0;0 1];
ob=obsv(a,c)
-----
ob =

     1     0
     0     1
     1     1
     4    -2
```

بررسی مشاهده پذیری (نتیجه 0 به معنای مشاهده پذیری است)

```
unob=length(a)-rank(ob)
-----
unob = 0
```

محاسبه ماتریس کنترل پذیری

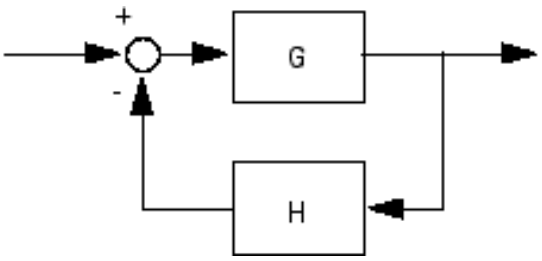
```
A = [1  1;4 -2];
B = [1 -1;1 -1];
Co = ctrb(A,B)
-----
Co =

     1    -1     2    -2
     1    -1     2    -2
```

بررسی کنترل پذیری

```
unco = length(A) - rank(Co)
-----
unco = 1
```

حلقه فیدبک

```
sys = feedback(G,H)
```



محاسبه گین فیدبک K برای قرار دادن قطب های سیستم (A,B) در محل های p

```
A;[0 1 ;0 1-]=
B;[0 ;0.9] =
P=[-10 -15];
K=place(A,B,[-10 -15])
-----
K=
166.6667  26.6667
```