تعریف تابع تبدیل:

برای تعریف تابع تبدیل یک سیستم در MATLAB چند روش وجود دارد:

۱- تعریف مخرج و صورت به صورت دو بردار چندجمله‌ای و استفاده از آنها در دستورهای دیگر

۲- تعریف مخرج و صورت به صورت دو بردار چندجمله‌ای و استفاده از تابع tf

۳- تعرف متغیر s با استفاده از تابع tf در حوزه لاپلاس و تعریف تابع

مثال: تابع $G(S)=\dfrac{1}{S^2+5S+2}$ را تعریف نمایید.

راه حل اول:

```
num = [1];
den = [1 5 2];
```

راه حل دوم:

```
num = [1];
den = [1 5 2];
G=tf(num,den)
```

راه حل سوم:

```
s = tf('s')
G = 1/(s^2+5*s+2)
```

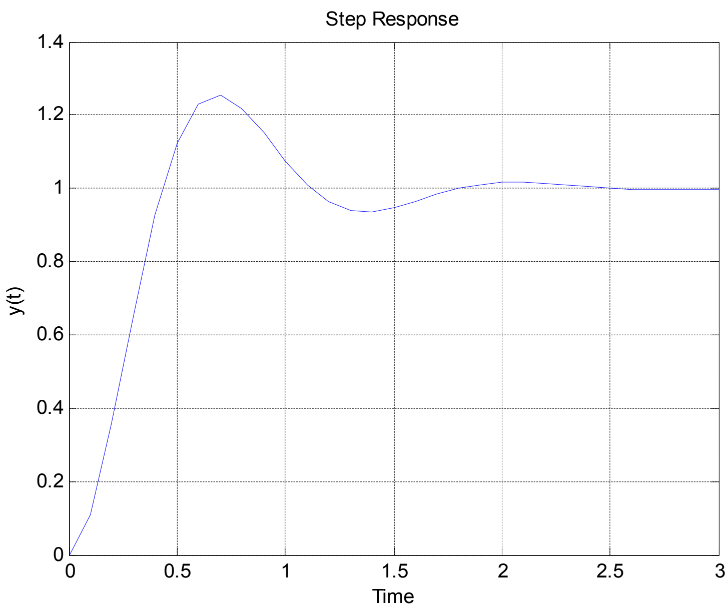
رسم پاسخ پله:

دستور step پاسخ پله واحد یک سیستم را که تابع تبدیل آن را دارید رسم می‌کند.

مثال: مطلوبست رسم پاسخ پله واحد سیستمی‌که تابع تبدیل آن $G(S)=\dfrac{25}{S^2+4S+25}$ می‌باشد.

حل: ابتدا تابع تبدیل را با روش اول تعریف می‌کنیم و سپس پاسخ پله آن رسم می‌نماییم.

```
num=25;
den=[1 4 25];
t=0:0.1:3;
y=step(num,den,t)
plot(t,y)
title('Step Response');
xlabel('Time');
ylabel('y(t)');
```



مثال: پاسخ پله واحد را برای سیستمی‌با تابع تبدیل زیر رسم کنید.

$$G(s)=\frac{1}{2s+1}$$

حال اگر یک المان تأخیری به مدار اضافه شود (e^{-s}) پاسخ پله واحد را رسم و با حالت قبل مقایسه کنید.

حل: با اضافه شدن المان تأخیری داریم:

$$H(s)=\frac{1}{2s+1}\,e^{-S}$$

تابع تبدیل با تأخیر را می‌توان به صورت زیر تعریف نمود:

```
num = [1];
den = [2 1];
G=tf(num,den,'iodelay',1)
```

یا اینکه می‌توان تأخیر e^{-S} را تقریب بزنیم. با استفاده از تقریب **Pade** داریم:

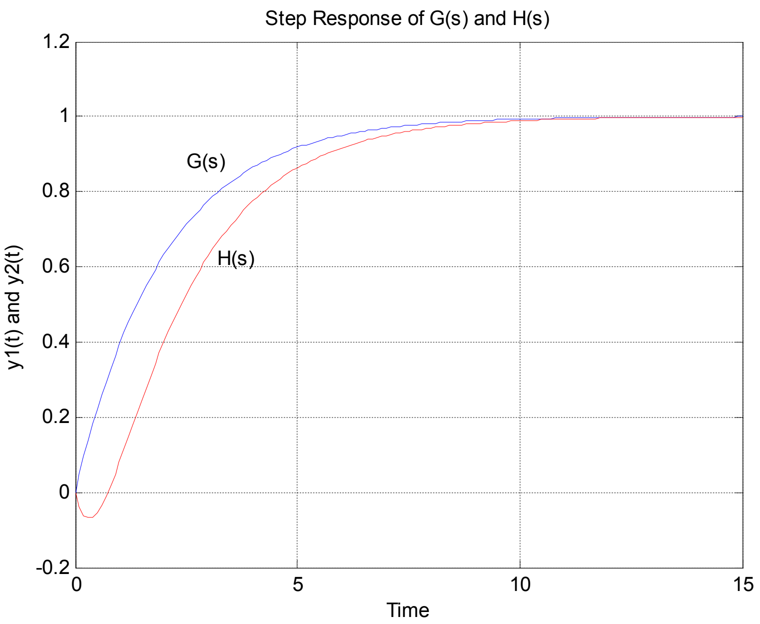
$$e^{-\alpha S}=\frac{1-\frac{\alpha}{2}S}{1+\frac{\alpha}{2}S}$$

پس:

$$H(S)=\frac{-S+2}{S^2+5S+2}$$

برای رسم G(**s**) و **H(s)** روی یک صفحه مختصات برنامه زیر را وارد کنید:

```
clear; clc
num1=1;
den1=[2 1];
t=[0:0.1:15];
y1=step(num1,den1,t);
plot(t,y1,'b')
gtext('G(s)')
hold on
num = [1];
den = [2 1];
G = tf(num,den,'iodelay',1)
y2=step(G,t);
plot(t,y2,'r')
gtext('H(s)')
grid
title('Step Response of G(s) and H(s)');
xlabel('Time');
ylabel('y1(t) and y2(t)');
```



توضیحات برنامه :

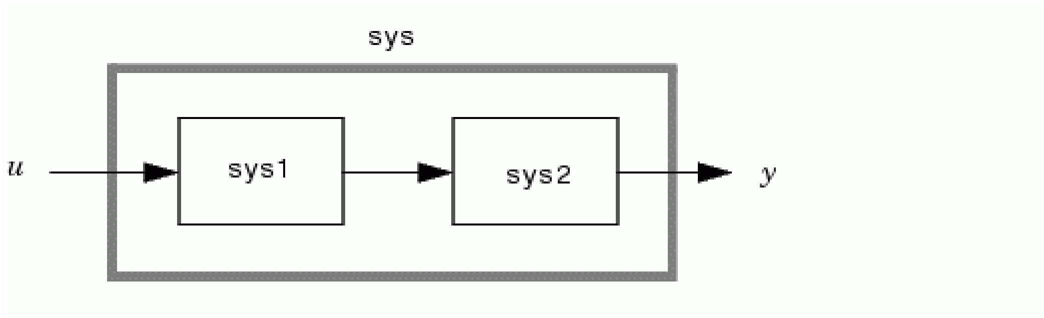
دستور **Plot** در سطر ۶ منحنی را با رنگ آبی رسم می‌کند آرگومان سوم مربوط به این دستور رنگ منحنی را مشخص خواهد کرد.

دستور **gtext** روی منحنی برچسب گذاری می‌کند به این ترتیب که وقتی برای اجرای این فایل **delay1** را تایپ کردید منحنی اول رسم می‌شود و یک علامت + روی منحنی ظاهر خواهد شد. شما با حرکت موس می‌توانید بر چسب مورد نظر را روی منحنی قرار دهید. برای این منظور روی موس کلیک کنید بعد از این عمل منحنی دوم نیز رسم خواهد شد با روش ذکر شده قبلی منحنی دوم را نیز بر چسب دهید.

توجه: اغلب توابع تبدیلی که در سیستم های کنترل با آنها مواجه می‌شویم در نیمه راست صفحه قطب یا صفری ندارند. این دسته از توابع را توابع تبدیل با فاز می‌نیمم گویند. وقتی تابع تبدیلی در نیمه راست صفحه s صفر یا قطب یا هر دو را داشته باشد. تابع تبدیل غیر می‌نیمم فاز است یک نمونه از این توابع، تابع تبدیل **H(s)** ذکر شده در این مثال است که بدلیل وجود یک صفر در سمت راست صفحه **S** **(s=2)** افتی در منحنی بوجود آمده است.

دستور **series**

این دستور ماتریس توابع تبدیل دو سیستم را که با یکدیگر سری شده اند محاسبه می‌کند.



مثال) مطلوبست محاسبه ماتریسهای فضای حالت برای دو سیستم با توابع تبـدیل $G(S)=\dfrac{1}{S+1}$ و $D(S)=\dfrac{2}{5S+1}$ که با یکدیگر سری شده اند.

حل:

```
num1=1;
den1=[1 1];
num2=2;
den2=[5 1];
[A1,B1,C1,D1]=tf2ss(num1,den1);
[A2,B2,C2,D2]=tf2ss(num2,den2);
[A,B,C,D]=series(A1,B1,C1,D1,A2,B2,C2,D2)
```

جواب :

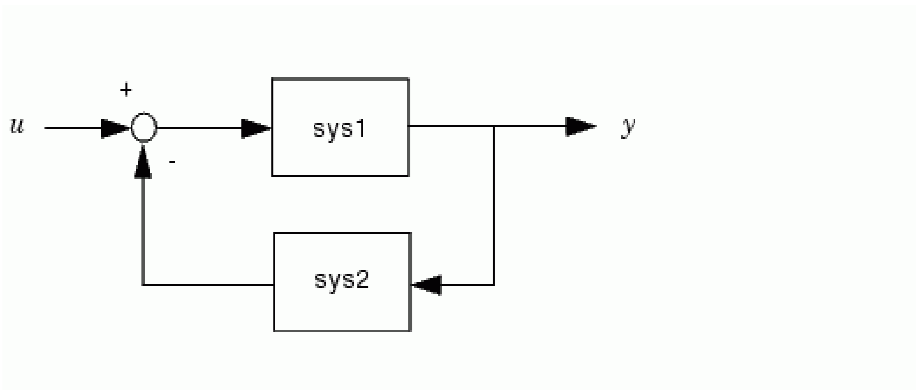
```
A = -0.2000    1.0000        B= 0        c=0.4000    0       D=0
         0   -1.0000           1
```

با اجرای دستور **ss2tf** برای ماتریسهای فضای حالت فوق تابع تبـدیل حاصلـضرب **D(S) , G(S)** را خواهید داشت.

$$D(S).G(S)=\frac{2}{5S^2+6S+1}$$

دستور **feedback**

این دستور برای محاسبه مدل LTI یک سیستم حلقه بسته با فیدبک به کار می‌رود.



مثال) اگر در سیستمی‌تابع تبدیل مسیر پیشرو $G(S)=\frac{1}{S+1}$ و تابع تبدیل مسیر فیدبک $D(S)=\frac{2}{5S+1}$ باشد ماتریس فضای حالت سیستم حلقه بسته فوق را بدست آورید.

حل:

```
num1 = 1;
den1 = [1 1];
sys1 = tf(num1,den1)
num2 = 2;
den2 = [5 1];
sys2 = tf(num2,den2)
cl = feedback(sys1,sys2)
```

جواب :

```
Transfer function:
    5 s + 1
---------------
5 s^2 + 6 s + 3
```

دستور tf2ss

این دستور تابع تبدیل یک سیستم را به فرم فضای حالت تبدیل می‌کند.

مثال) ماتریسهای فضای حالت برای تابع تبدیل $G(S)=\frac{1}{S+1}$ بدست آورید.

```
num=1;
den=[1 1];
[A,B,C,D]=tf2ss(num,den)
```

جواب:

```
A=-1      B=1      C=1      D=0
```

دستور ss2tf

این دستور ماتریسهای فضای حالت یک سیستم را به فرم تابع تبدیل بر می‌گرداند. فرمت کلی دستور بصورت **[num,den] = ss2tf (a,b,c,d,iu)** می‌باشد که در آن **iu** شماره ورودی است.

مثال) تابع انتقال یک سیستم را به ازای **d=0,c=1,b=1,a=-1** بدست آورید.

حل:

```
A=-1;
B=1;
C=1;
D=0;
[num,den]=ss2tf(A,B,C,D,1)
```

جواب :

```
num=1
den=1 1
```

دستور ss2zp

توسط این دستور با داشتن معادلات فضای حالت، مقادیر صفر و قطب تابع تبدیل محاسبه خواهد شد.
فرمت این دستور به صورت **[Z,P,K]=ss2zp(a,b,c,d,iu)** است که **iu** شماره ورودی و **K** گین تابع تبدیل است.

مثال) مقادیر صفر و قطب سیستمی‌با معادلات فضای حالت زیر را بدست آورید.

حل :

```
A=[-1 0;1 -0.2];
B=[1 ; 0];
C=[0 0.4];
D=0;
[Z,P,K]=ss2zp(A,B,C,D,1)
```

جواب :

```
Z = [ ]
P=
-0.2
-1
K= 0.4
```

رسم مکان هندسی ریشه‌های یک سیستم:

دستور rlocus مکان هندسی ریشه های معادله مشخصه یک سیستم خطی را بر حسب تغییرات k رسم می‌کند.

$$H(S) = 1 + \frac{num(s)}{den(s)} = 0$$

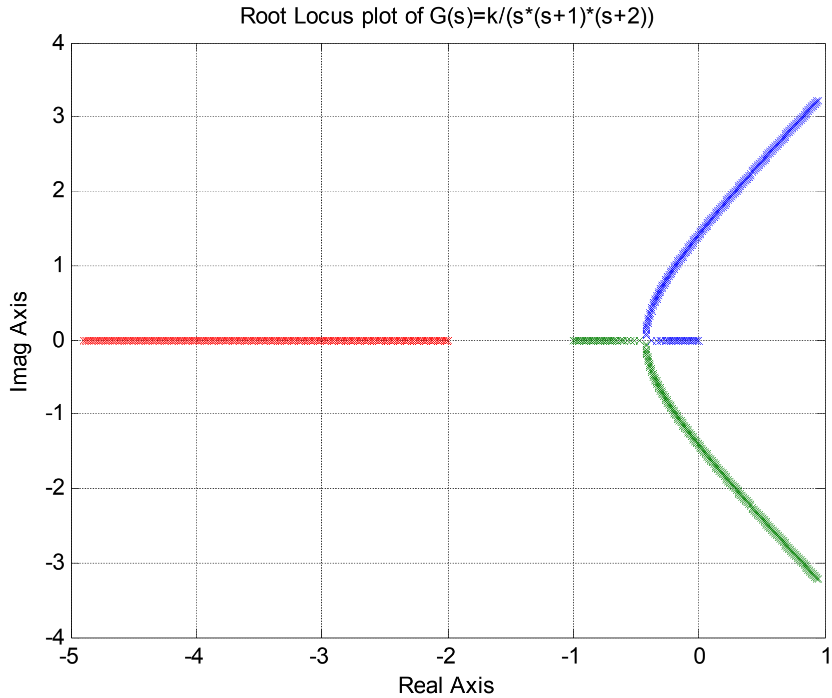
فرمت این دستور به شکل **rlocus(num,den,k)** است. اگر سیستمی‌با فیدبک منفی داشته باشیم آنگاه:

$$H(S) = \frac{D(S).G(S)}{1 + D(S).G(S)}$$

که معادله مشخصه سیستم فوق بصورت **1+DG=0** است.

مثال) برای یک سیستم کنترل خطی توابع تبدیل مسیر پیشرو **D(S)=1** و $G(S) = \frac{K}{S(S+1)(S+2)}$ می‌باشد. مطلوبست رسم مکان هندسی ریشه ها بازای مقادیر مختلف **K**

```
% rlocus1
num=1;
den=[1 3 2 0];
k=0:.01:50;
r=rlocus(num,den,k);
plot(r,'x')
grid
title('Root Locus plot of G(s)=k/(s*(s+1)*(s+2))')
xlabel('Real Axis')
ylabel('Imag Axis')
```



توضیحات برنامه

توجه کنید که هر چه فاصله ای **K** کوچکتر باشد شکل مکان هندسی پیوسته تر خواهد شد. این برنامه را با نام **rlocus1.m** در **Login** خودتان ذخیره کنید و سپس برای اجرای این برنامه تایپ کنید **[r,k] = rlocus1(50)** عدد **MAX** دلخواه است و بستگی به این دارد که تا چه حدی می‌خواهیم **K** را افزایش دهیم. حالا **K** را مرتباً افزایش دهید مثلا از 50 به 100 یا به 150 و نتیجه را ببینید همانطور که ملاحظه می‌کنید با افزایش **K** مکان هندسی در قسمت بیشتری از ناحیه سمت راست صفحه **S** پیشروی می‌کند.

دستور abs

این دستور برای محاسبه اندازه یک چند جمله ای بر حسب **S** بکار می‌رود.

دستور angle

این دستور برای محاسبه زاویه یک چند جمله ای بر حسب **S** بکار می‌رود. اندازه این زاویه بر حسب رادیان خواهد بود.

مثال) دامنه و فاز تابع تبدیل زیر را برای **s = 4 j** بدست آورید.

$$G(s)=\frac{s+1}{(s+2)(s+3)}$$

حل :

```
s=4j;
abs((s+1)/((s+2)*(s+3)))
```

جواب :

```
ans =
    0.1844
```

```
angle((s+1)/((s+2)*(s+3)))
```

جواب :

```
ans =
   -0.7086
```

مثال) در مثال **rlocus1**

الف) حداکثر **K** را که باعث ناپایداری می‌شود بدست آورید.

ب) مقدار **K** را طوری بدست آورید که $\xi = 0.5$ شود.

ج) مقادیر ریشه ها را به ازای $\xi = 0.5$ محاسبه کنید.

حل :

الف)

$$G(S)=K\frac{1}{S(S+1)(S+2)}$$

مقدار **K** که باعث ناپایداری می‌شود همان مقداری است که به ازای آن، مکان هندسی محور موهومی‌را قطع می‌کند. هنگام رسم مکان هندسی بطور تقریبی این نقطه 1.4j است. چون این نقطه روی مکان هندسی می‌باشد. باید در شرط اندازه صدق کند پس :

$$|D.G|=1$$

$$\left|\frac{K}{S(S+1)(S+2)}\right|=1$$

$$K=|S(S+1)(S+2)|$$

با استفاده از دستور **abs** داریم :

```
s=1.4j;
abs(s*(s+1)*(s+2))
```

جواب : **5.9**

پس حداکثر **K** که باعث ناپایداری می‌شود $k \cong 6$ خواهد بود.

ب)

می‌دانیم $\sin\theta = \xi$ که در این صورت $\theta = 30^{\circ}$ خواهد شد.

حالا باید حطی با زاویه 30° نسبت به محور موهومی‌و از مبدا مختصات روی مکان هندسی رسم کنید.
محل برخورد این خط با مکان هندسی ریشه مورد نظر خواهد بود که به ازای این ریشه **K** را محاسبه خواهیم کرد.

ابتدا باید خط را رسم کنید. برای این منظور نقطه ای را روی محور موهومی‌در نظر بگیرید (نقطه دلخواه است) سپس با توجه به منحنی صفحه بعد داریم :

$$tg30^{\circ} = \frac{A}{5}$$

A=-2.88

برنامه مربوط به **rlocus1** را اجرا کنید مکان هندسی را با خطوط ضخیم رسم نکنید یعنی بجای دو خط برنامه زیر

```
r=rlocus(num,den,k);
plot(r,'x')
```

یک خط برنامه زیر را وارد کنید.

```
plot(rlocus(num,den,k));
```

در ادامه برنامه فوق برنامه زیر را برای رسم خط روی مکان هندسی تایپ کنید :

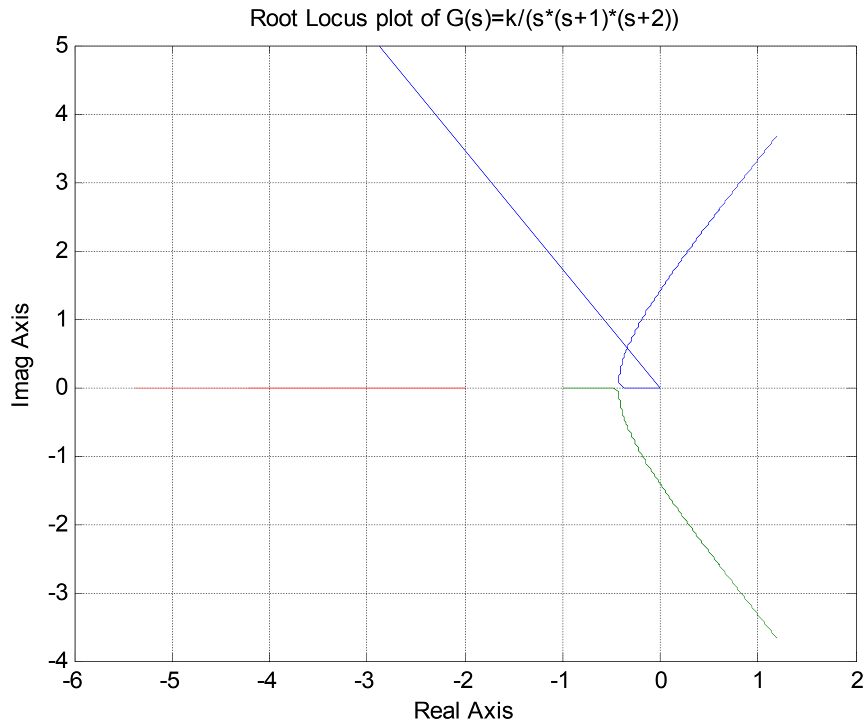
```
hold on
 plot([0 -2.88],[0 5]
```

محل تقاطع خط با مکان هندسی $S = -0.35 \pm 0.57j$ است با استفاده از شرط اندازه داریم :

```
s=-0.35+0.57j;
 abs(s*(s+1)*(s+2))
```

جواب : 1.038

پس به ازای **k=1.038**، $\xi = 0.5$ خواهد شد.



ج)

راه حل اول :

$$S^3 + 3S^2 + 2S + K = 0$$
$$S^3 + 3S^2 + 2S + 1.038 = 0$$

با استفاده از دستور **roots** ریشه ها بدست می‌آید.

S1=-0.33+0.57j
S2=-.033-0.57j
S3=-2.3

راه حل دوم :

در برنامه **rlocus1** محدوده **K** چنین تعریف کنید :

K=0:0.01:1.038

که بدین ترتیب مکان هندسی تا **k=1.38** رسم خواهد شد و **S1** ، **S2** و **S3** بطور تقریبی قابل تعیین هستند. همانطور که می‌بینید روش اول دقیق تر است.

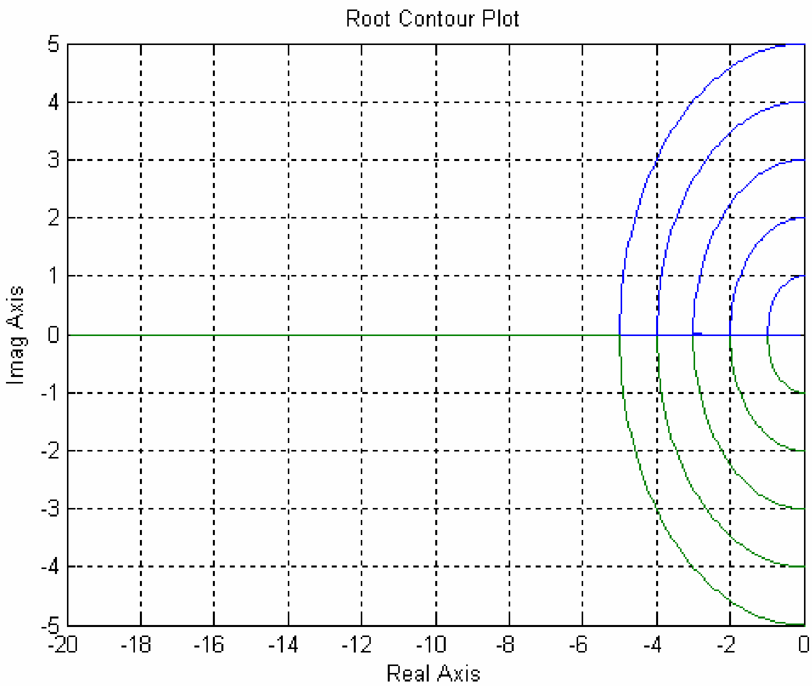
تمرین) مکان هندسی را برای $G(S) = K\dfrac{S+1}{S^3 + pS^2}$ به ازای **p=20** و **p=9** و **p=12** رسم کنید.

(**D(s)=1**)

مثال) **Root Contuor** را برای $G(S)=\dfrac{aS}{S^2+K}$ و **D(S)=1** رسم کنید.

حل:

```
% rlocus2
k=[1 4 9 16 25];
a=0:0.1:20;
num=[1 0];
for i=1:5
    den=[1 0 k(i)];
    plot(rlocus(num,den,a));
    hold on
end
grid
title('Root Contour Plot')
xlabel('Real Axis')
ylabel('Imag Axis' )
```



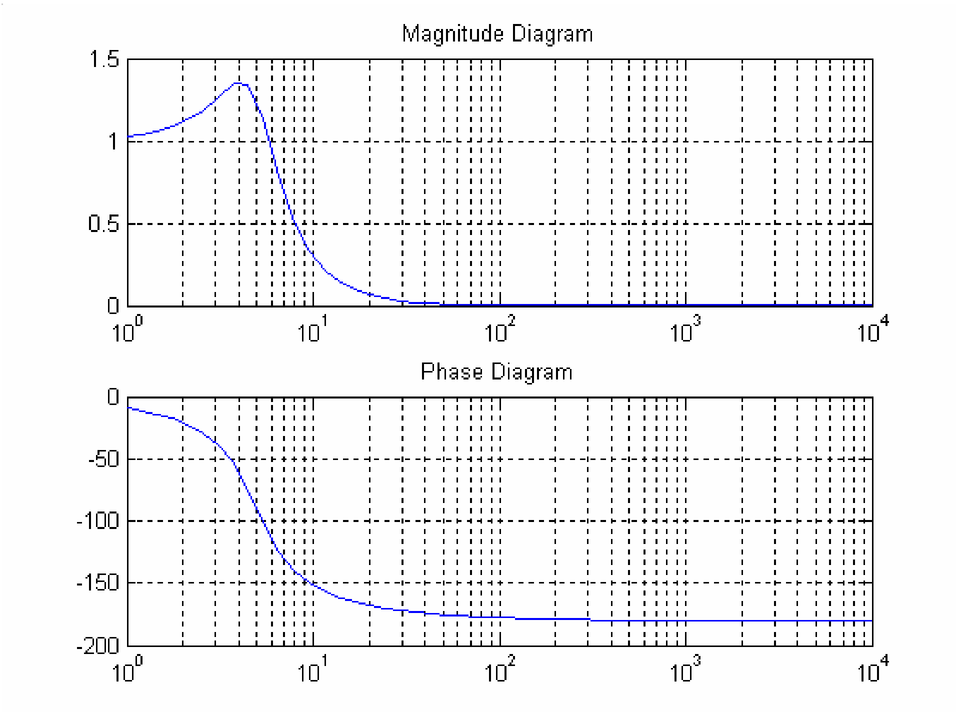
رسم منحنی Bode

دستور bode برای رسم اندازه و فاز تابع تبدیل یک سیستم بکار می‌رود. فرمت این دســتور بـه شـکل **[mag,phase]=bode(num,den,w)** است.

مثال) دیاگرام bode را برای سیستمی‌با تابع تبدیل $G(S)=\dfrac{25}{S^2+4S+25}$ رسم کنید.

حل :

```
num=25;
den=[1 4 25];
w=logspace(0,4);
[mag,phase]=bode(num,den,w);
subplot(211);
semilogx(w,mag)
grid
title('Magnitude Diagram')
subplot(212);
semilogx(w,phase)
grid
title('Phase Diagram')
```



توضیحات برنامه

W=logspace(0,4) محور افقی را از 100 تا 10^4 بصورت لگاریتمی‌تقسیم بندی می‌کند.

Semilogx(w,mag) مانند دســتور plot اسـت بـا ایـن تفـاوت کـه **semilogx** درجـه بنـدی محـور مختصات را بصورت نیمه لگاریتمی‌انجام می‌دهد. یعنی محور **x** را لگاریتمی‌و محور **y** را خطـی درجـه بندی می‌کند.

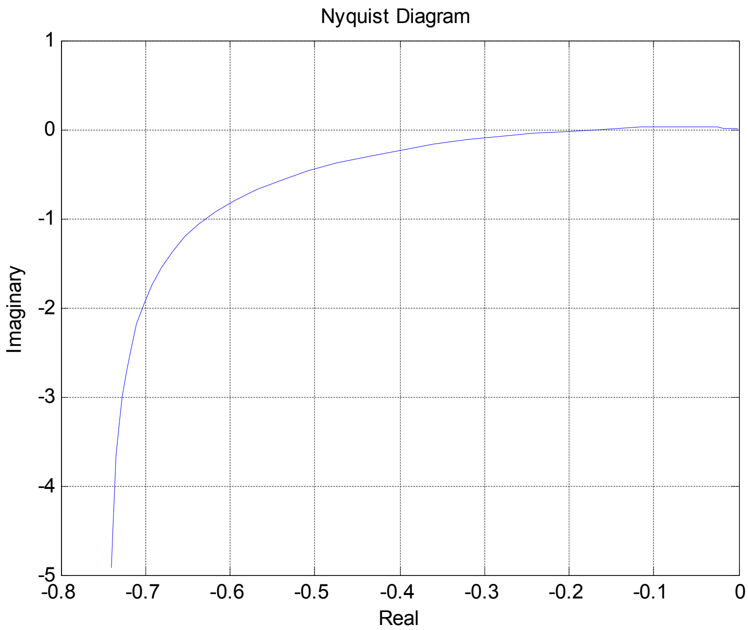
رسم منحنی نایکویست:

دستور nyquist منحنی نایکویست یک سیستم LTI را رسم می‌کند. فرمت این دستور بصورت **[re,im]=nyquist(num,den,w)** است.

مثال) منحنی **nyquist** برای تابع تبدیل $D(S).G(S)=\frac{K}{S(S+1)(S+2)}$ به ازای **k=1** را رسم و پایداری سیستم حلقه بسته را بررسی نمایید.

حل:

```
num=1;
den=[1 3 2 0];
w=logspace(-1,1);
re,im]=nyquist(num,den,w);
plot(re,im);
grid
title('Nyquist Diagram');
xlabel('Real');
ylabel('Imaginary');
```



توجه: همانطور که ملاحظه می‌کنید به ازای **K=1** منحنی **nyquist** نقطه (**-1+0j**) را شامل نمی‌شود و سیستم پایدار است.