بسم الله الرحمن الرحیم

جزوه

آزمایشگاه

کنترل خطی

نام استاد

مهندس

سینا محمدی

بهار۱۳۹۹

سیستم کنترلی:

در مطالعات سیستم کنترل کننده جهت تنظیم قدرت خروجی سیستم به کار میرود که جهت رسیدن به وضعیت تعادل قدرت بر اساس تغییرات ناگهانی و تولید ویا بار استفاده می شود. براساس کاهش انحراف فرکانس، عدم تعادل بین تولید و تقاضا در وضعیت تولید به وجود می آید که باید توسط کنترل کننده ها تنظیم شود. در این ترم به طراحی کنترل کننده برای سیستم های مختلف پرداخته می شود. اولین مبحث مورد بررسی، استفاده از کنترل کننده های کلاسیک خواهد بود.

کنترل کننده های کلاسیک:

منظور از طراحی کنترل کننده های کلاسیک برای یک سیستم، تعیین ضرایب Ki،Kd،Kp است که بسته به کاربرد و عملکرد مطلوب سیستم می تواند به شیوه های متفاوتی بیان شود. در این بخش ٤ ویژگی مهم پاسخ زمانی یک سیستم مورد بررسی قرار گرفته و معیار از مطلوبیت خروجی سیستم به واسطه ی آنها تعریف خواهد شد. این ویژگی ها عبارت انداز:زمان سعود، زمان نشست، بیشترین فراجهش، و انتگرال قدرمطلق خطا.

در ادامه تعریف مختصری ازهریک ازاین ویژگی ها بیان می شود و از روی آن ها معیار بهینگی پاسخ خروجی تعریف می شود.

زمان صعود:

زمان صعود، زمانی است که در طی آن پاسخ سیستم از ۱۰٪ مقدار نهایی خود به ۹۰٪ آن می رسد.

زمان نشست:

به زمانی اطلاق می شود که بعد ازآن زمان پاسخ سیستم در فاصله ی ۲٪ از پاسخ نهایی اش باقی می ماند.

بیشترین فراجهش:

ماکزیمم فراجهش به صورت تفاضل دو پاسخ y_{max}(مقدار بیشینه پاسخ) و y_{ss} (حد نهایی پاسخ) تعریف می شود.

انتگرال قدرمطاق خطا:

به صورت معادله ی $I_{AE} = \int_{0}^{\infty} |e_{(t)}|\, dt$ تعریف می شود.

به خاطر پیاده سازی زمان گسسته در محاسبه این انتگرال حد بالایی آن تا یک حدمعین (معمولا تا ۳برابر زمان نشست) در نظر گرفته می شود که جواب قابل قبول برای این انتگرال به دست می آید. بیش از۹۰٪ از کنترل کننده های امروزی، کنترل کننده از نوع کلاسیک اند. علت این امر آسانی و قابل فهم بودن این نوع کنترل کننده ها و کاربرد آسان آن است. کنترل کننده ی هر سیستم به عنوان مغز فرمان دهنده در آن سیستم مهم ترین نقش را در راستای عملکرد مطلئب آن دارد. برای کنترل موثر هر سیستم باید آن سیستم درک و مدل سازی گردد. اکثر فرایندهای کنترلی، مطابق با استفاده از نظریه فیدبک منفی برای رساندن خروجی به یک سطح ویا نگهداشتن آن در یک محدوده ی مشخص می باشد. بدین صورت سیگنال خطا یا اختلاف بین خروجی فرایند و ورودی آن برای کنترل فرایند استفاده می شوند. در این میان کنترل کننده نقش مهمی دارد. برای دست یابی به پاسخ سریع با درصد فراجهش کم در سیستم های حلقه بسته، در ابتدای پاسخ یعنی زمانی که سیگنال ورودی تازه اعمال شده است باید از سیگنال کنترل کننده وکنترلی سریع استفاده نمود ولی هنگامی که پاسخ به مقدار مطلوب نزدیک شد، باید سیگنال کنترل را کوچک کرد. مهمترین عاملی که در طراحی ها مدنظر قرار می گیرد درصد فراجهش است.

این مقدار از آن جهت اهمیت دارد که مقدار زیاد آن باعث وارد شدن زیان به سیستم می شود. اما در بسیاری از سیستم ها برای سرعت بخشیدن به پاسخ سیستم مجبوریم مقداری فراجهش را قبول کنیم. بنارایـن هدف اغلب طراحی ها کاهش درصد فراجهش بدون لطمه خوردن به پاسخ سیستم است. متداول ترین نوع کنترل کننده کلاسیک، کنترل کننده PID می باشد که دارای ۳بخش تناسبی،انتگرال گیرو مشتق گیر است. فرم کلی به صورت معادله ی

$$U_{(t)} = K_p\, e_{(t)} + K_i \int_{\cdot}^{t} \left|e_{(t)}\right| \mathrm{dt} + K_d \frac{d_{e(t)}}{dt}$$

می باشد. در این حالت ضریب K_p جهت افزایش سرعت پاسخ سیستم و افزایش دقت به کار می رود و اگر ضریب K_p بسیار بزرگ باشد، سرعت انحراف بیشتر می شود و ممکن است سیستم به حالت ناپایداری برسد. ضریب K_d جهت افزایش و بهبود عملکرد دینامیکی سیستم به کار می رود. همچنین ضریب K_i جهت حذف خطای حالت ماندگار به کار می رود ولی اگر مقدار آن بیش از حد زیاد باشد باعث پدیده ی اشباع در سیستم شده و ناپایداری را در سیستم به همراه دارد. اغلب جهت تنظیم این ضرایب در کنترل کننده های PID از روش صحیح و خطا و زیگلرنیکولن استفاده می شود.

تنظیم پارامترهای کنترل کننده های کلاسیک(PID):

در این روش مقادیر انتگرال گیر و مشتق گیر را برابر صفر قرار داده و K_p را طوری تنظیم می کنیم که پاسخ مطلوب بدون در نظر گرفتن خطای نهایی به دست آید. سپس K_p را افزایش می دهیم و t_d را طوری تنظیم می کنیم که فراجهش سیستم کاهش یابد. آنگاه K_i را طوری تنظیم می کنیم که مقدار خطای حالت ماندگار به حداقل برسد. این مراحل را تا حدی ادامه می دهیم که K_p به اندازه کافی و ممکن بزرگ باشد. ایراد اصلی این کنترل

کننده ها، مدت زمان زیاد جهت تنظیم پارامترها و سختی آن جهت تنظیم نهایی که باید به صورت بهینه باشد است. مراحل کار در روش زیگلرنیکولز به شرح زیر است:

۱-K_p را تاحدی افزایش می دهیم تا سیستم دچار یک نوسان کند شود. اینGainراK_u می نامیم .

۲-زمان بین دو پیک خروجی سیستم را اندازه گرفته و آن را t_u می نامیم.

۳-مقادیرGainهای کنترل کننده را از روی جدول زیر تنظیم می کنیم.

	K_p	T_i	T_d
P	۰,۵K_u		
PI	۰,۴۵K_u	$\frac{t_u}{۱.۲}$	
PID	۰,٦K_u	$\frac{t_u}{۱.۲}$	$\frac{t_u}{۸}$

به طور کلی ۳روش برای تعریف سیستم در متلب وجود دارد:

۱-استفاده از تابع TF (حوزه فرکانس)

۲-استفاده از ساختارZPK(حوزه فرکانس)

۳-استفاده از فضای حالت(حوزه زمان)

در درس اول، به تعریف سیستم بااستفاده از تابعTFمی پردازیم. تابعTFدر متلب به صورت زیر تعریف می شود:

Sys=tf(num,den). دراین رابطه num ریشه های صورت تابع تبدیل یا به عبارتی دیگر صفرهای سیستم اند. Den نیز بیانگر قطب های سیستم است.

مثال : $\frac{١}{s+١}$ =

```
>> num=[١];

>> den=[١ ١];

>> sys=tf(num,den)
```

sys = $\frac{١}{s+١}$

مثال : $\frac{١}{٢s^{٢}+٢s+١}$

```
>> num=[١];

>> den=[٢ ٢ ١];

>> sys٢=tf(num,den)

sys٢ =

       ١

  ---------------

  ٢ s^٢ + ٢ s + ١
>> sys٣=sys*sys٢




sys٣ =

            ١

  -----------------------

  ٢ s^٣ + ٤ s^٢ + ٣ s + ١
```

روش دیگر و راحت تر این است که ابتدا s را تعریف کرده و سپس تابع تبدیل را به همان شکل که در صورت مسئله می باشد در متلب تایپ می کنیم.

مثال: $\frac{١}{s+١} =$

```
>> s=tf('s');
>> sys=١/(s+١)
sys =
   ١
  -----
  s + ١
```

مثال: $\frac{١}{٢s^{٢}+٢s+١} =$

```
>> s=tf('s');
>> sys=١/(٢*s^٢+٢*s+١)
sys =
        ١
  ---------------
  ٢ s^٢ + ٢ s + ١
```

مثال: $\frac{١٠+s}{s^{٢}+٥s+٤} =$

نکته: برای نوشتن به روش اول، ابتدا عبارت باید استاندارد سازی شود.

```
>> sys=tf([١ ١٠],[١ ٥ ٤])
sys =
    s + ١٠
  -------------
  s^٢ + ٥ s + ٤
```

ولی روش دوم نیازی به استاندارد سازی عبارت ندارد.

```
>> s=tf('s');

>> sys=(۱۰+s)/(s^۲+۵*s+۴)

sys =

  s + ۱۰

 -------------

 s^۲ + ۵ s + ۴
```

ریشه یابی تابع تبدیل(به دست آوردن صفر و قطب سیستم):

```
>> s=tf('s');

>> sys=(۱۰*s+۱۰۰)/s^۲*(s+۱)*(s+۳);

>> p۱=[۱۰ ۱۰۰];

>> p۲=[۱ ۴ ۳ ۰ ۰];

>> r۱=roots(p۱)

r۱ =

  -۱۰

>> r۲=roots(p۲)

r۲ =

     ۰

     ۰

   -۳

   -۱
```

مثال: $G_{(s)}=\frac{۱}{(s+۱)(s^{۲}+۱)}=$

```
>> s=tf('s');
```

```
>> sys=١/(s+١)*(s^٢+١);
>> p١=[١];
>> p٢=[١ ١ ١ ١];
>> r١=roots(p١)
r١ =
   Empty matrix: ٠-by-١
>> r٢=roots(p٢)
r٢ =
  -١,٠٠٠٠ + ٠,٠٠٠٠i
  -٠,٠٠٠٠ + ١,٠٠٠٠i
  -٠,٠٠٠٠ - ١,٠٠٠٠i
```

تابع ZPK:

تابع ZPK در متلب به صورت SYS=ZPK(Z,P,K) تعریف می شود که در آن Z صفرهای سیستم و P قطب های سیستم و K بهره سیستم می باشد.

مثال: $\frac{١}{s+١} =$

```
>> sys=zpk([١],[-١],[١])
sys =
  (s-١)
  -----
  (s+١)
```

مثال: $\frac{(٤s)(s-١)(s+٧)}{(s+١)(s+٥)(s^٢-٢s+٢)} =$

نکته: ریشه های صورت مشخص می باشد(٠,١,-٧)=z

```
>> s=tf('s');
```

```
>> sys=١/(s+١)*(s+٥)*(s^٢-٢*s+٢);
>> p١=[١ ٤ -٥ ٢ ١٠];
>> r١=roots(p١)
r١ =
  -٥,٠٠٠٠ + ٠,٠٠٠٠i
   ١,٠٠٠٠ + ١,٠٠٠٠i
   ١,٠٠٠٠ - ١,٠٠٠٠i
  -١,٠٠٠٠ + ٠,٠٠٠٠i
>> sys=zpk([٠ ١ -٧],[-٥ ١+i ١-i -١],[٤])
sys =
       ٤ s (s-١) (s+٧)
  --------------------------
  (s+٥) (s+١) (s^٢ - ٢s + ٢)
```

مثال: $H_{(S)}=\begin{bmatrix} \frac{١}{S-٠.٣} \\ \frac{٢(S+٠.٥)}{(S-٠.١+J)(S-٠.١-J)} \end{bmatrix}=$

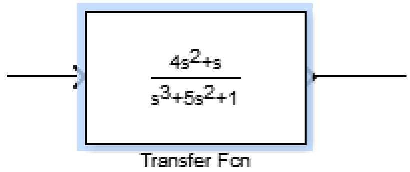
```
>> z={[];-.٥};
>> p={.٣;[.١-i .١+i]};
>> k=[١;٢];
>> h=zpk(z,p,k)

h =
          ١
   ١:  -------
       (s-٠,٣)
```

```
          ٢ (s+٠,٥)
   ٢:  --------------------
       (s^٢ - ٠,٢s + ١,٠١)
```

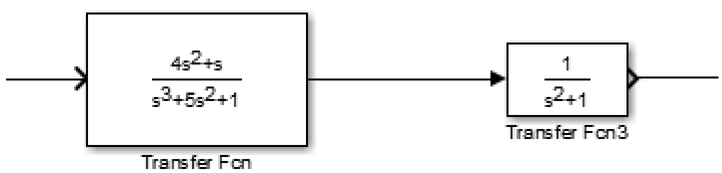
مثال: $H_{(s)} = \begin{bmatrix} \frac{-١}{s} & \frac{٣(s+٥)}{(s+١)^٢} \\ \frac{٢(s^٢-٢s+٢)}{(s-١)(s-٢)(s-٣)} & ٠ \end{bmatrix} =$

```
>> z={[] [-٥];[١-i ١+i] []};
>> p={٠ [-١ -١];[١ ٢ ٣] ٠};
>> k=[-١ ٣;٢ ٠];
>> h=zpk(z,p,k)
h =
       -١
   ١:  --
       s
       ٢ (s^٢ - ٢s + ٢)
   ٢:  ------------------
       (s-١) (s-٢) (s-٣)


       ٣ (s+٥)
   ١:  -------
       (s+١)^٢
   ٢:  ٠
```

مثال: برای هریک از سیستم های زیر تابع تبدیل را به دست آورید.




```
>> num=[٤ ١ ٠];
>> den=[١ ٥ ٠ ١];
>> sys=tf(num,den)
sys =
    ٤ s^٢ + s
  ---------------
  s^٣ + ٥ s^٢ + ١
```

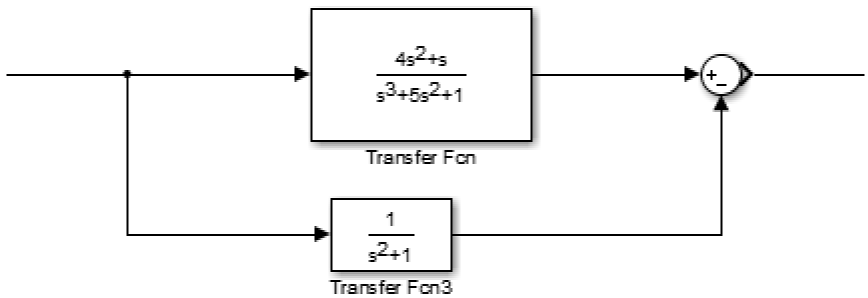



```
>> numg١=[٤ ١ ٠];
>> deng١=[١ ٥ ٠ ١];
>> numg٢=[١];
>> deng٢=[١ ٠ ١];
>> [nums,dens]=series(numg١,deng١,numg٢,deng٢);
>> sys=tf(nums,dens)
sys =
```

٤ s^٢ + s

----

s^٥ + ٥ s^٤ + s^٣ + ٦ s^٢ + ١



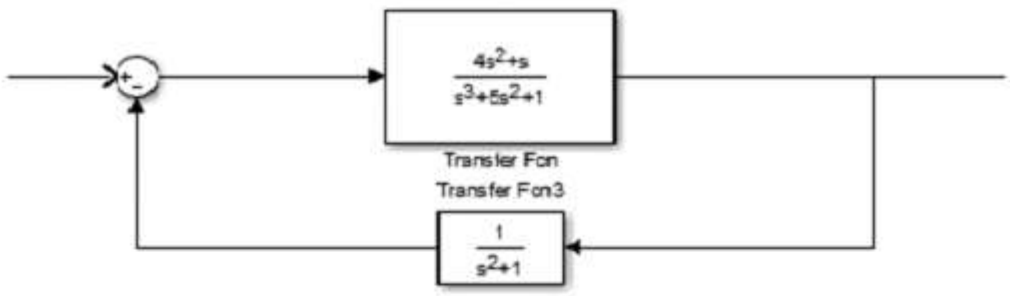

```
>> numg١=[٤ ١ ٠];
>> deng١=[١ ٥ ٠ ١];
>> numg٢=[١];
>> deng٢=[١ ٠ ١];
>> [nums,dens]=parallel(numg١,deng١,numg٢,deng٢);
>> sys=tf(nums,dens)
```

sys =

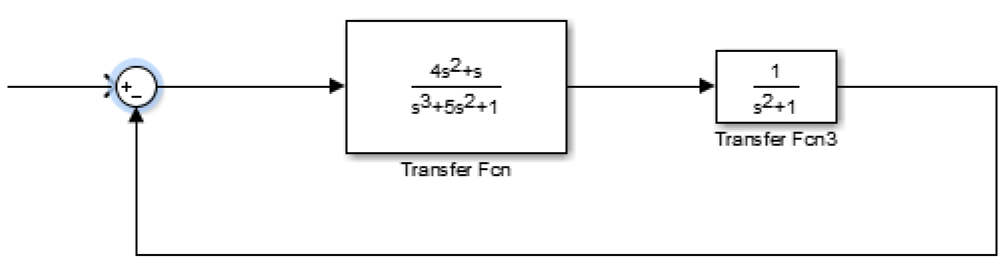
٤ s^٤ + ٢ s^٣ + ٩ s^٢ + s + ١

----

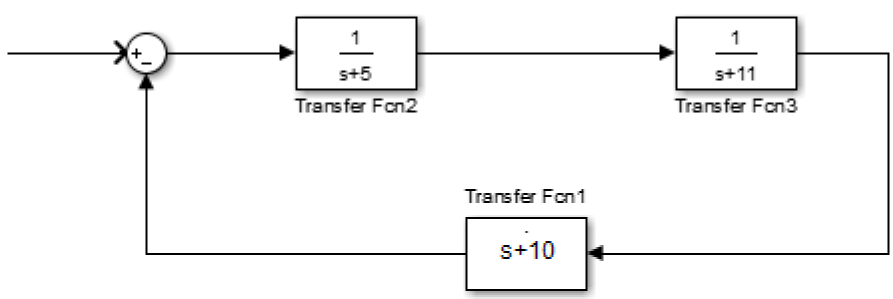
s^٥ + ٥ s^٤ + s^٣ + ٦ s^٢ + ١




```
>> numg١=[٤ ١ ٠];
>> deng١=[١ ٥ ٠ ١];
>> numg٢=[١];
>> deng٢=[١ ٠ ١];
>> [nums,dens]=feedback(numg١,deng١,numg٢,deng٢);
>> sys=tf(nums,dens)
sys =
      ٤ s^٤ + s^٣ + ٤ s^٢ + s
  ---------------------------------
  s^٥ + ٥ s^٤ + s^٣ + ١٠ s^٢ + s + ١
```

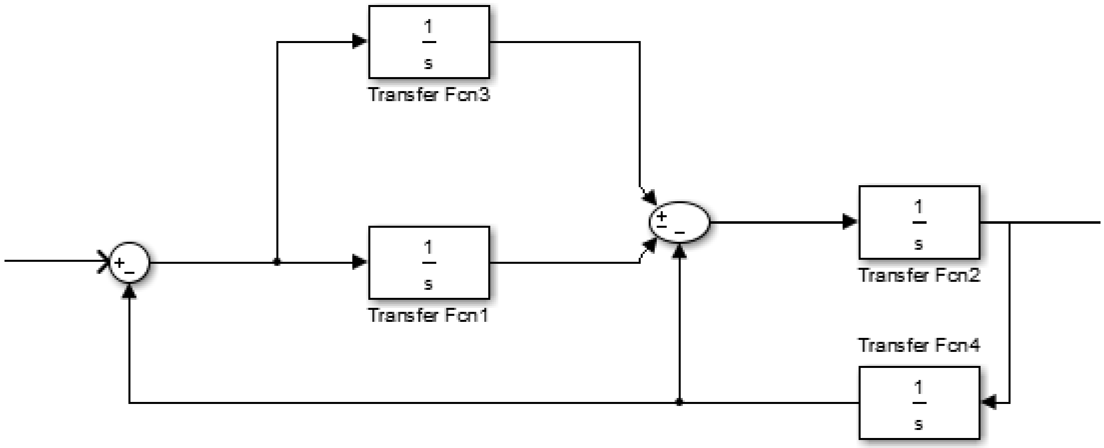


```
>> numg١=[٤ ١ ٠];
>> deng١=[١ ٥ ٠ ١];
>> numg٢=[١];
>> deng٢=[١ ٠ ١];
>> [nums,dens]=series(numg١,deng١,numg٢,deng٢);
>> [nums,dens]=cloop(nums,dens,-١);
>> sys=tf(nums,dens)
sys =
              ٤ s^٢ + s
  ----------------------------------
  s^٥ + ٥ s^٤ + s^٣ + ١٠ s^٢ + s + ١
```



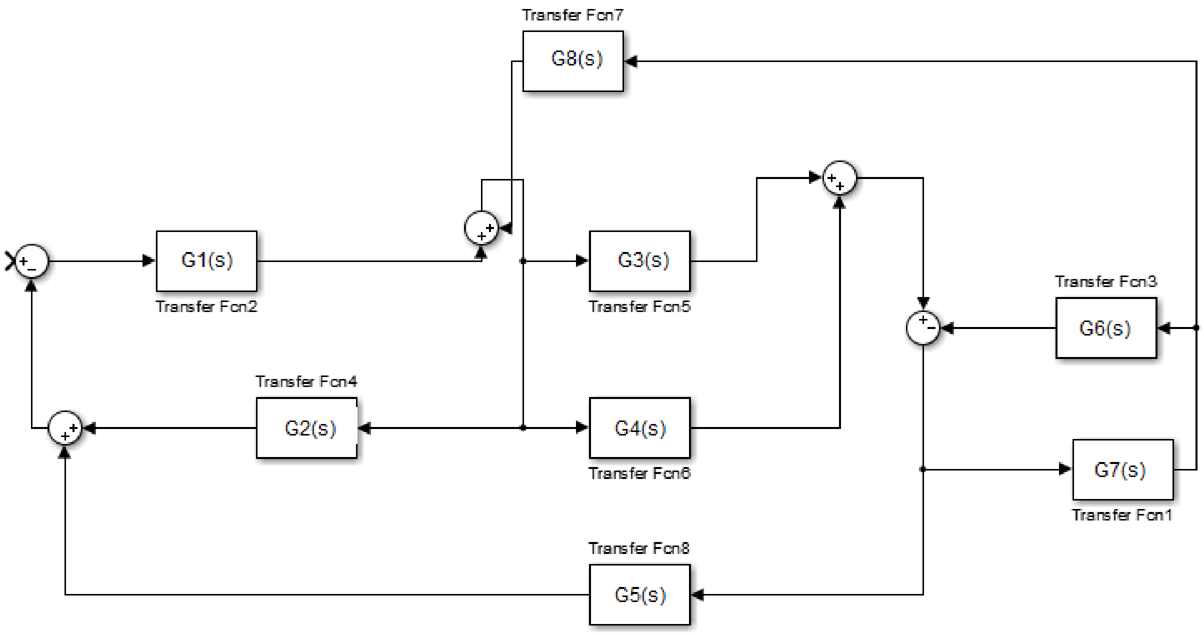

```
>> numg١=[١];
>> deng١=[١ ٥];
>> numg٢=[١];
>> deng٢=[١ ١١];
```

```
>> numg٣=[١ ١٠];
>> deng٣=[١];
>> [nums,dens]=series(numg١,deng١,numg٢,deng٢);
>> g=tf(nums,dens)
g =
        ١
  ---------------
  s^٢ + ١٦ s + ٥٥
>> h=tf(numg٣,deng٣)
h =
  s + ١٠
>> [num,den]=feedback(g,h)
num =
        ١
  ---------------
  s^٢ + ١٧ s + ٦٥
den =
     ٠
```




```
>> numg١=[١];
>> deng١=[١ ٠];
>> [nums,dens]=parallel(numg١,deng١,numg١,deng١);
>> e=tf(nums,dens)
e =
 ٢ s
 ---
 s^٢
>> [nums١,dens١]=feedback(numg١,deng١,numg١,deng١);
>> n=tf(nums١,dens١)
n =
    s
 -------
 s^٢ + ١
>> numg٢=[٢];
>> deng٢=[١ ٠];
>> numg٣=[١ ٠];
>> deng٣=[١ ٠ ١];
```

```
>> [nums٢,dens٢]=series(numg٢,deng٢,numg٣,deng٣);
>> g=tf(nums٢,dens٢)
g =
    ٢ s
  -------
  s^٣ + s
>> h=tf(numg١,deng١)
h =
  ١
  -
  s
>> [nums٣,dens٣]=feedback(g,h)
nums٣ =
       ٢ s^٣
  ---------------
  s^٥ + s^٣ + ٢ s^٢
```



$G١=\frac{١}{S+٧}$ $G٢=\frac{١}{S^٢+٦S+٥}$ $G٣=\frac{١}{S+٨}$ $G٤=\frac{١}{S}$

$G٥=\frac{٧}{S+٣}$ $G٦=\frac{١}{S^٢+٧S+٥}$ $G٧=\frac{٥}{S+٥}$ $G٨=\frac{١}{S+٩}$ $G٩=١$

```
g١=tf([٠ ٠ ١],[٠ ١ ٧]);
g٢=tf([٠ ٠ ١],[١ ٦ ٥]);
g٣=tf([٠ ٠ ١],[٠ ١ ٨]);
g٤=tf([٠ ٠ ١],[٠ ١ ٠]);
g٥=tf([٠ ٠ ٧],[٠ ١ ٣]);
g٦=tf([٠ ٠ ١],[١ ٧ ٥]);
g٧=tf([٠ ٠ ٥],[٠ ١ ٥]);
g٨=tf([٠ ٠ ١],[٠ ١ ٩]);
g٩=tf([٠ ٠ ١],[٠ ٠ ١]);
t١=append(g١,g٢,g٣,g٤,g٥,g٦,g٧,g٨,g٩);

q =
```

```
  ١  -٢  -٥   ٩
  ٢   ١   ٨   ٠
  ٣   ١   ٨   ٠
  ٤   ١   ٨   ٠
  ٥   ٣   ٤  -٦
  ٦   ٧   ٠   ٠
  ٧   ٣   ٤  -٦
  ٨   ٧   ٠   ٠
>> inputs=٩;
>> outputs=٧;
>> ts=connect(t١,q,inputs,outputs)
ts =
   ١٠ s^٧ + ٢٩٠ s^٦ + ٣٣٥٠ s^٥ + ١٩٨٠٠ s^٤ + ٦٣٦٩٠ s^٣+ ١٠٨٩١٠ s^٢ +
٨,٨٩٥e٠٤ s + ٢,٧e٠٤
  -----------------------------------------------------------------
  s^١٠ + ٤٥ s^٩ + ٨٦٦ s^٨ + ٩٣٠٥ s^٧ + ٦,١١٥e٠٤ s^٦ + ٢,٥٣٣e٠٥ s^٥
+ ٦,٥٧e٠٥ s^٤ + ١,٠٢٧e٠٦ s^٣ + ٨,٩٠٩e٠٥ s^٢+ ٣,٦٢٦e٠٥ s + ٤,٢e٠٤
```

راه حل:

مرحله اول: تعریف توابع تبدیل هر یک از بلوک ها.

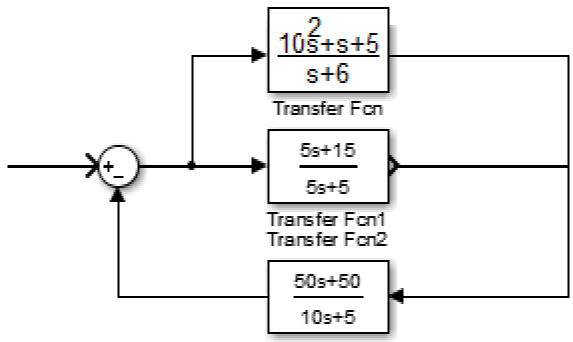
نکته: تابع G٩ به عنوان بلوک ورودی درنظر گرفته می شود.

مرحله دوم: با استفاده از دستور append یک بانک اطلاعاتی از بلوک های تبدیل می سازیم.

مرحله سوم: تعریف ماتریسی که نحوه اتصالات بلوک های مختلف به یکدیگر را تشریح کند. درایه اول هرکدام از سطرهای این ماتریس باید نشان دهنده شماره ی بلوکی باشد که می خواهیم اتصالات آن را شرح دهیم. درایه های بعدی هم نشان دهنده ی بلوک های ورودی به این بلوک است. دقت شود که بلوک

های ورودی را باید با علامتشان درنظر بگیریم. برای نمونه در همین مثال سطر پنجم به صورت (۶- ۴ ۳ ۵) تعریف شده است که در آن عدد اول نشان دهنده بلوک پنجم و سایر اعداد بلوک های ورودی به این بلوک اند.علامت بلوک ششم منفی است.

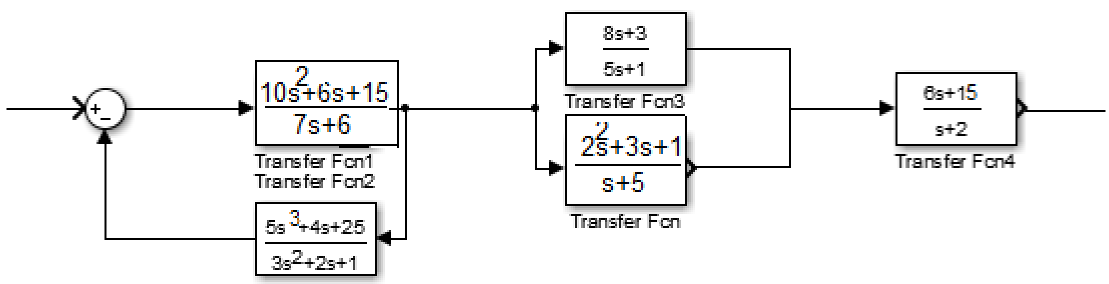
مثال)




```
>> numg۱=[۱۰ ۱ ۵];
>> deng۱=[۱ ۶];
>> sys=tf(numg۱,deng۱);
>> numg۲=[۵ ۱۵];
>> deng۲=[۵ ۵];
>> sys۱=tf(numg۲,deng۲);
>> a=parallel(sys,sys۱)
a =
  ۵۰ s^۳ + ۶۰ s^۲ + ۷۵ s + ۱۱۵
  ----------------------------
      ۵ s^۲ + ۳۵ s + ۳۰

>> numg۳=[۵۰ ۵۰];
```

```
>> deng٣=[١٠ ٥];
>> sys٢=tf(numg٣,deng٣);
>> b=feedback(a,sys٢)
b =
  ٥٠٠ s^٤ + ٨٥٠ s^٣ + ١٠٥٠ s^٢ + ١٥٢٥ s + ٥٧٥
 ----------------------------------------------
 ٢٥٠٠ s^٤ + ٥٥٥٠ s^٣ + ٧١٢٥ s^٢ + ٩٩٧٥ s + ٥٩٠٠
```

مثال)




```
>> numg١=[١٠ ٦ ١٥];
>> deng١=[٧ ٦];
>> sys=tf(numg١,deng١);
>> numg٢=[٥ ٠ ٤ ٢٥];
>> deng٢=[٣ ٢ ١];
>> sys١=tf(numg٢,deng٢);
>> a=feedback(sys,sys١)
```

```
a =

      ٣٠ s^٤ + ٣٨ s^٣ + ٦٧ s^٢ + ٣٦ s + ١٥

  -----------------------------------------------------

  ٥٠ s^٥ + ٣٠ s^٤ + ١٣٦ s^٣ + ٣٠٦ s^٢ + ٢٢٩ s + ٣٨١

>> numg٣=[٨ ٣];

>> deng٣=[٥ ١];

>> sys٢=tf(numg٣,deng٣);

>> numg٤=[٢ ٣ ١];

>> deng٤=[١ ٥];

>> sys٣=tf(numg٤,deng٤);

>> b=parallel(sys٢,sys٣)

b =

  ١٠ s^٣ + ٢٥ s^٢ + ٥١ s + ١٦

  ---------------------------

      ٥ s^٢ + ٢٦ s + ٥

>> c=series(a,b)

c =

  ٣٠٠ s^٧ + ١١٣٠ s^٦ + ٣١٥٠ s^٥ + ٤٤٥٣ s^٤ + ٥٠٧٥ s^٣

           + ٣٢٨٣ s^٢ + ١٣٤١ s + ٢٤٠

  -----------------------------------------------------------------

  ٢٥٠ s^٧ + ١٤٥٠ s^٦ + ١٧١٠ s^٥ + ٥٢١٦ s^٤ + ٩٧٨١ s^٣

 + ٩٣٨٩ s^٢ + ١١٠٥١ s + ١٩٠٥

>> numg٥=[٦ ١٥];

>> deng٥=[١ ٢];
```

```
>> sys٤=tf(numg٥,deng٥);

>> d=series(c,sys٤)

d =

  ١٨٠٠ s^٨ + ١١٢٨٠ s^٧ + ٣٥٨٥٠ s^٦ + ٧٣٩٦٨ s^٥ + ٩٧٢٤٥ s^٤

   + ٩٥٨٢٣ s^٣ + ٥٧٢٩١ s^٢ + ٢١٥٥٥ s + ٣٦٠٠

  ---------------------------------------------------------------

  ٢٥٠ s^٨ + ١٩٥٠ s^٧ + ٤٦١٠ s^٦ + ٨٦٣٦ s^٥ + ٢٠٢١٣ s^٤

   + ٢٨٩٥١ s^٣ + ٢٩٨٢٩ s^٢ + ٢٤٠٠٧ s + ٣٨١٠
```

رابطه توابع تبدیل وتوابع فضای حالت:

معادله کلی فضای حالت به صورت زیر می باشد:

X=A_X+B_U

y=C_x+D

به عنوان مثال داریم:

A ماتریس: X=$\begin{bmatrix} ۰ & ۱ & ۰ \\ -۱ & -۱ & ۰ \\ ۱ & ۱ & -۲ \end{bmatrix}$, B ماتریس : X=$\begin{bmatrix} ۰ \\ ۱ \\ ۰ \end{bmatrix}$, Cماتریس: y =$\begin{bmatrix} ۰ & ۰ & ۱ \end{bmatrix}$

I=$\begin{bmatrix} ۱ & ۰ & ۰ \\ ۰ & ۱ & ۰ \\ ۰ & ۰ & ۱ \end{bmatrix}$

نکته:ماتریس I باید مطابق با ماتریس A باشد.

نکته:اگر ماتریس D را نداشته باشیم آن را برابرصفر قرار می دهیم.(D=[۰])

دستورات لازم جهت پیاده سازی فضای حالت و تابع تبدیل:

۱: SYMS S ۲:$\left[C(SI-A)^{-۱}\right]$ B*D ۳: W= [(S*I) -A]

٤ :(C*inv(w))*B+D ٥: sys=ss(A,B,C,D)

جهت رسم تابع پله از دستور Step ورسم تابع ضربه از دستور impulse استفاده می کنیم.

نکته: برای ایجاد یک بازه زمانی برای رسم توابع می توان مانند زیر عمل کرد:

T=[۰:۰,۰۱:٤۰]

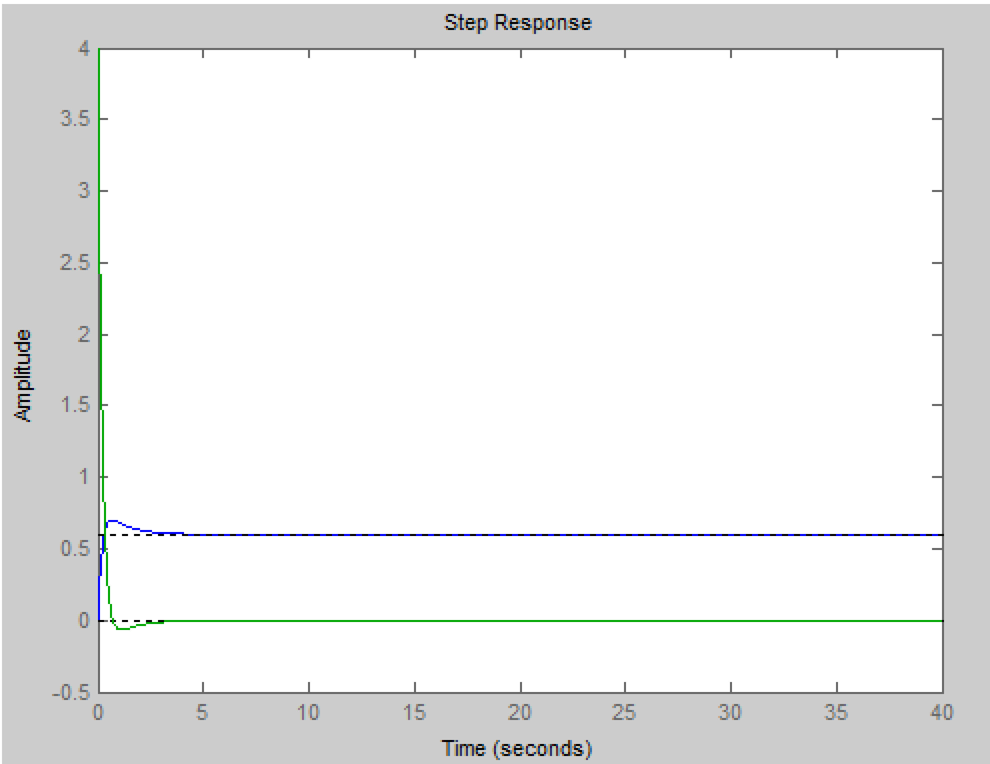
نکته:برای نمایش دو منحنی پله و ضربه روی یک نمودار از دستورhold all استفاده می کنیم.

مثال:تابع تبدیل را به همراه رسم ورودی پله و ضربه برای سیستم زیر به دست آورید.

$\begin{bmatrix} X'_۱ \\ X'_۲ \end{bmatrix}$ = $\begin{bmatrix} ۰ & ۱ \\ -٥ & -٦ \end{bmatrix}$ $\begin{bmatrix} x_۱ \\ x_۲ \end{bmatrix}$ +$\begin{bmatrix} ۰ \\ ۱ \end{bmatrix}$u y=$\begin{bmatrix} ۳ & ٤ \end{bmatrix}$ $\begin{bmatrix} x_۱ \\ x_۲ \end{bmatrix}$

```
>> a=[٠ ١;-٥ -٦];
>> b=[٠;١];
>> c=[٣ ٤];
>> d=[٠];
>> i=[١ ٠;٠ ١];
>> sys=ss(a,b,c,d)
sys =
  a =
       x١  x٢
   x١   ٠   ١
   x٢  -٥  -٦
  b =
       u١
   x١   ٠
   x٢   ١
  c =
       x١  x٢
   y١   ٣   ٤
  d =
       u١
   y١   ٠

>> g=tf(sys)
g =
    ٤ s + ٣
```

```
 -------------
 s^۲ + ٦ s + ٥
>> t=[۰:۰,۰۱:٤۰];
>> step(sys,t)
hold all
>> impulse(sys,t)
```

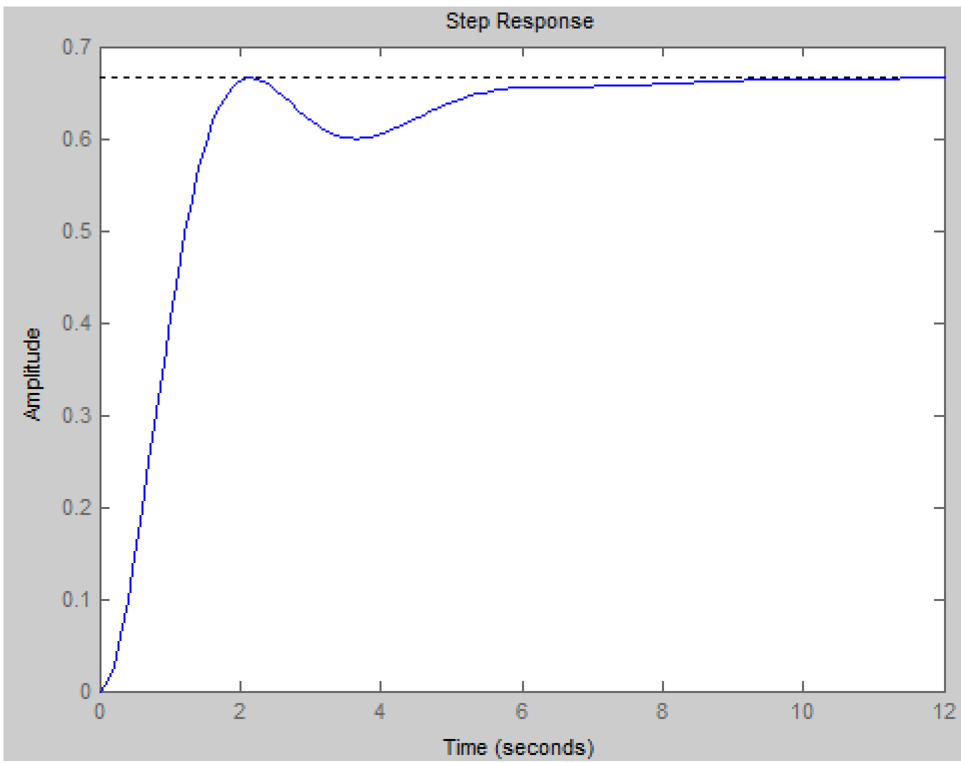


مثال:برای سیستمی با این تابع تبدیل حلقه باز، پاسخ پله حلقه بسته را رسم کنید.

$H_{(S)}=\frac{۳S+۲}{۲S^۳+٤S^۲+٥S+۱}$

```
>> num=[۳ ۲];
```

```
>> den=[٢ ٤ ٥ ١];
>> g=tf(num,den)
>> t=feedback(g,١)
t =
        ٣ s + ٢
  ----------------------
  ٢ s^٣ + ٤ s^٢ + ٨ s + ٣
>> step(t)
```

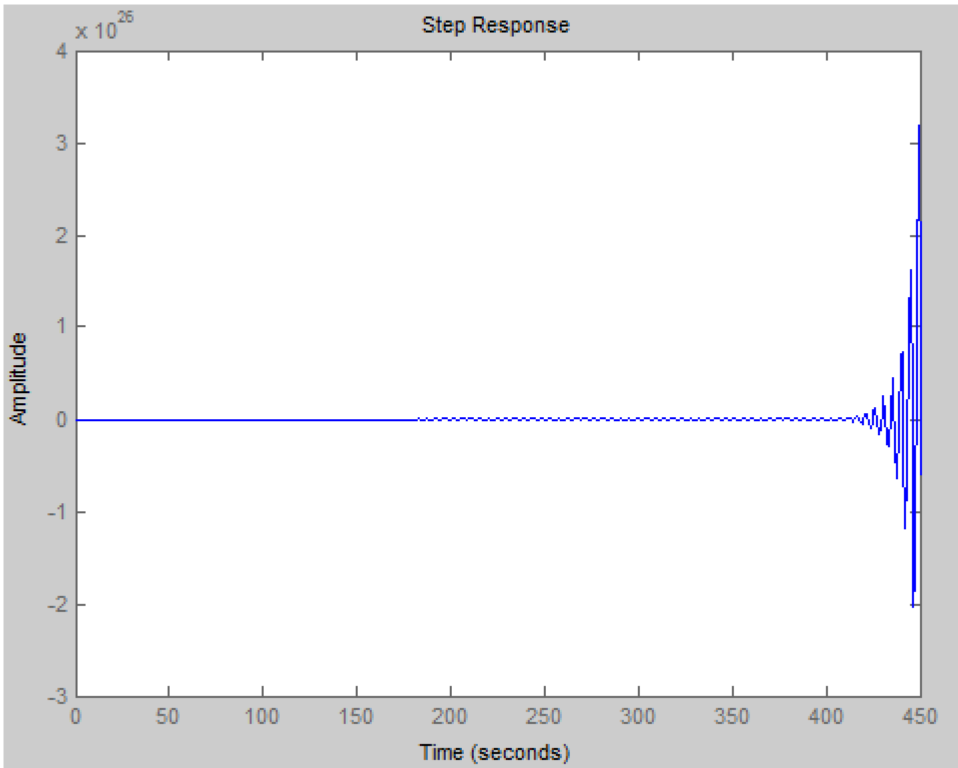


مثال:نمودار پله سیستم زیر را به دست آورید.

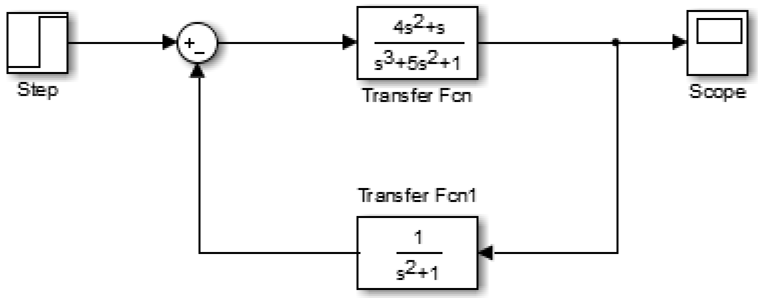
Sys=$\frac{٤s^{٤}+s^{٣}+٤s^{٢}+s}{s^{٥}+٥s^{٤}+s^{٣}+١٠s^{٢}+s+١}$

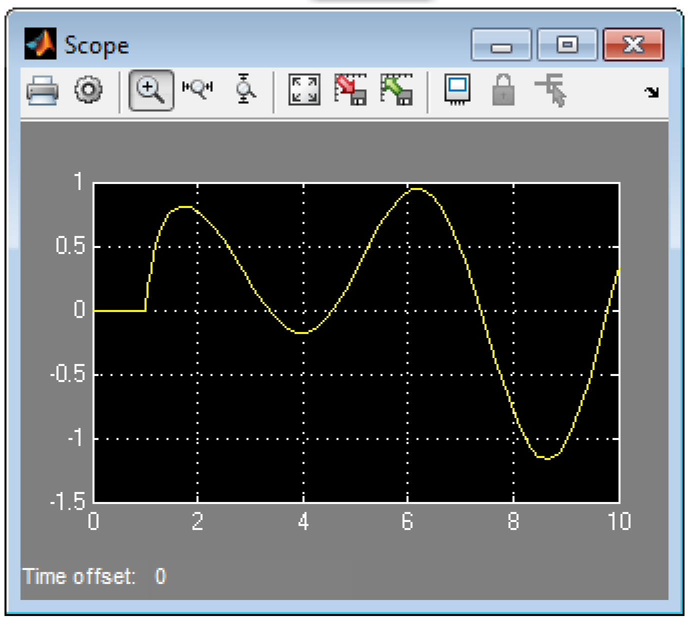
```
>> num=[٤ ١ ٤ ١ ٠];
```

```
>> den=[١ ٥ ١ ١٠ ١ ١];
>> g=tf(num,den);
>> step(g)
```



مثال:





تمرین:

۱-سیستم زیر را درنظر بگیرید.مطلوبست: الف) به دست آوردن تابع تبدیل این سیستم.

ب) رسم پاسخ پله و ضربه ج)به دست آوردن بهره و صفر و قطب(Z,P,K)

$A=\frac{s+۱}{s^۲+۳s+۱}$

الف) >> num=[۱ ۱];

>> den=[۱ ۳ ۱];

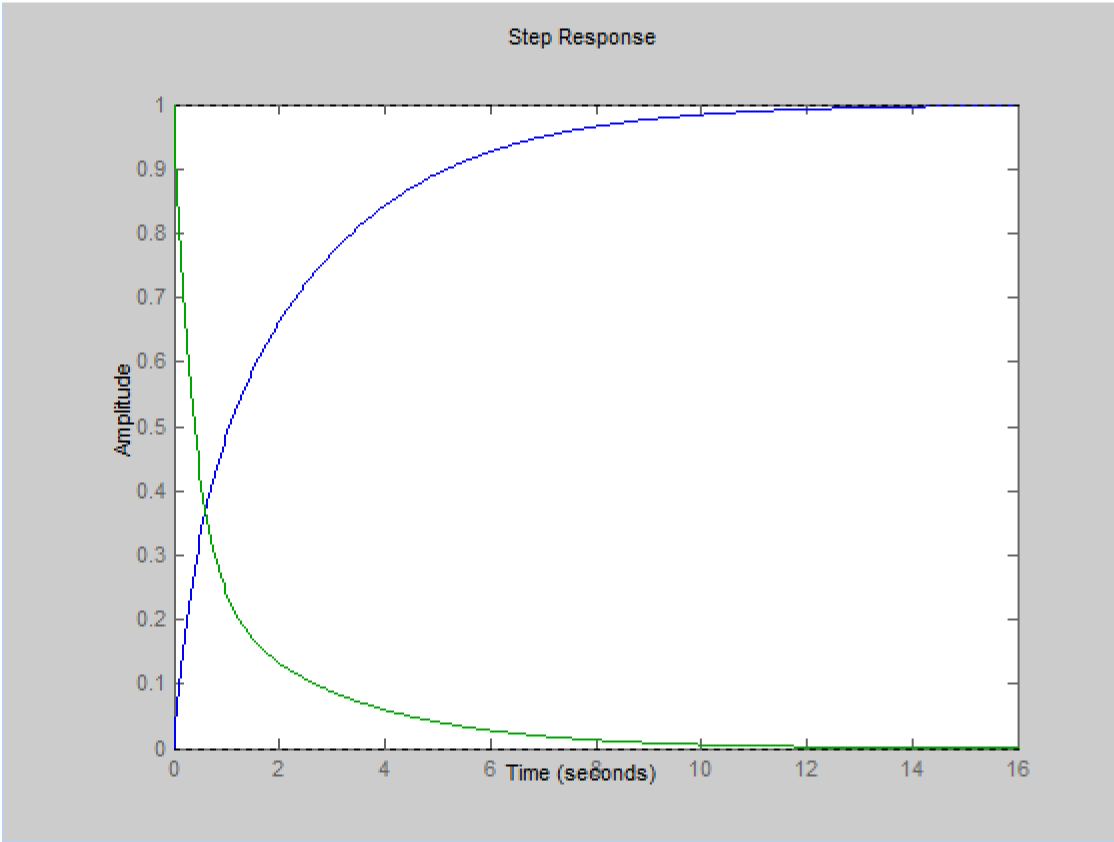
>> sys=tf(num,den)

```
sys =

      s + ۱
  -------------
  s^۲ + ۳ s + ۱
```

ب) >> step(sys)

>> hold all

>> impulse(sys)



ج) >> r=roots([۱ ۳ ۱]);

>> t=zpk([-۱],[r],[۱])

t =

(s+۱)

(s+۲,۶۱۸) (s+۰,۳۸۲)

۲-نمایش فضای حالت دو سیستم به صورت زیر است:

الف) $x_۱ = A_{x۱} + B_u$

$y = c\, x_۱ \qquad A=\begin{bmatrix} -۱ & ۰ & ۱ \\ ۱ & -۲ & ۰ \\ ۰ & ۰ & -۲ \end{bmatrix} \qquad B=\begin{bmatrix} ۰ \\ ۰ \\ ۱ \end{bmatrix} \qquad C=[۱ \quad ۱ \quad ۰]$

ب)$\dot{x}_۲$=A$x_۲$+B_u

$$Y=c\,x_۲ \qquad A=\begin{bmatrix} ۰ & ۱ & ۰ \\ -۱ & -۱ & ۰ \\ ۱ & ۰ & ۰ \end{bmatrix} \qquad B=\begin{bmatrix} ۰ \\ ۱ \\ ۰ \end{bmatrix} \qquad C=[۰ \quad ۰ \quad ۱]$$

مطلوبست:

الف)تعیین تابع تبدیل هر دو سیستم از روی معادلات فضای حالت

ب)تعیین صفر و قطب و بهره هر دو سیستم (Z,P,K)

ج)رسم پاسخ پله هر دو سیستم

سیستم اول:

الف: >> a=[-۱ ۰ ۱;۱ -۲ ۰;۰ ۰ -۲];

>> b=[۰;۰;۱];

>> c=[۱ ۱ ۰];

>> d=[۰];

>> syms s

>> i=eye(۳);

>> w=[(s*i)-a];

>> g(s)=[c*inv(w)]*b+d

g(s) =

۱/((s + ۱)*(s + ۲)) + ۱/((s + ۱)*(s + ۲)^۲)

>> sys=ss(a,b,c,d);

>> f=tf(sys)

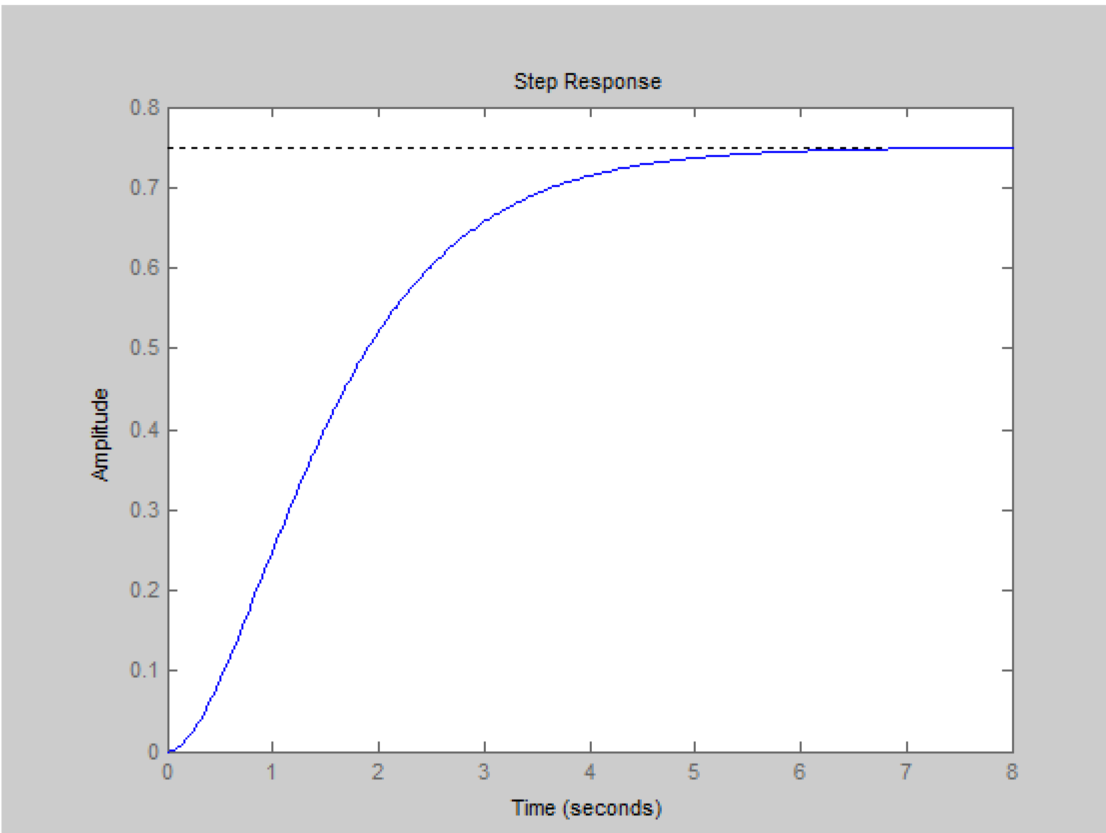
f =

```
        s + ۳
  ---------------------
  s^۳ + ۵ s^۲ + ۸ s + ٤
```

ب: >> r=roots([۱ ۵ ۸ ٤]);

```
>> e=zpk([-٣],[r],[١])
e =
     (s+٣)
  -------------
  (s+٢)^٢ (s+١)
```

ج: >> step(sys)



سیستم دوم:

الف: >> a=[٠ ١ ٠;-١ -١ ٠;١ ٠ ٠];

>> b=[٠;١;٠];

>> c=[٠ ٠ ١];

>> d=[٠];

>> i=eye(٣);

>> syms s

>> w=[(s*i)-a];

>> g(s)=(c*inv(w))*b+d

g(s) =

١/(s^٣ + s^٢ + s)

>> sys=ss(a,b,c,d);

>> f=tf(sys)

f =

١

----

s^٣ + s^٢ + s

ب: >> r=roots([١ ١ ١ ٠]);

>> h=zpk([],[r],[١])

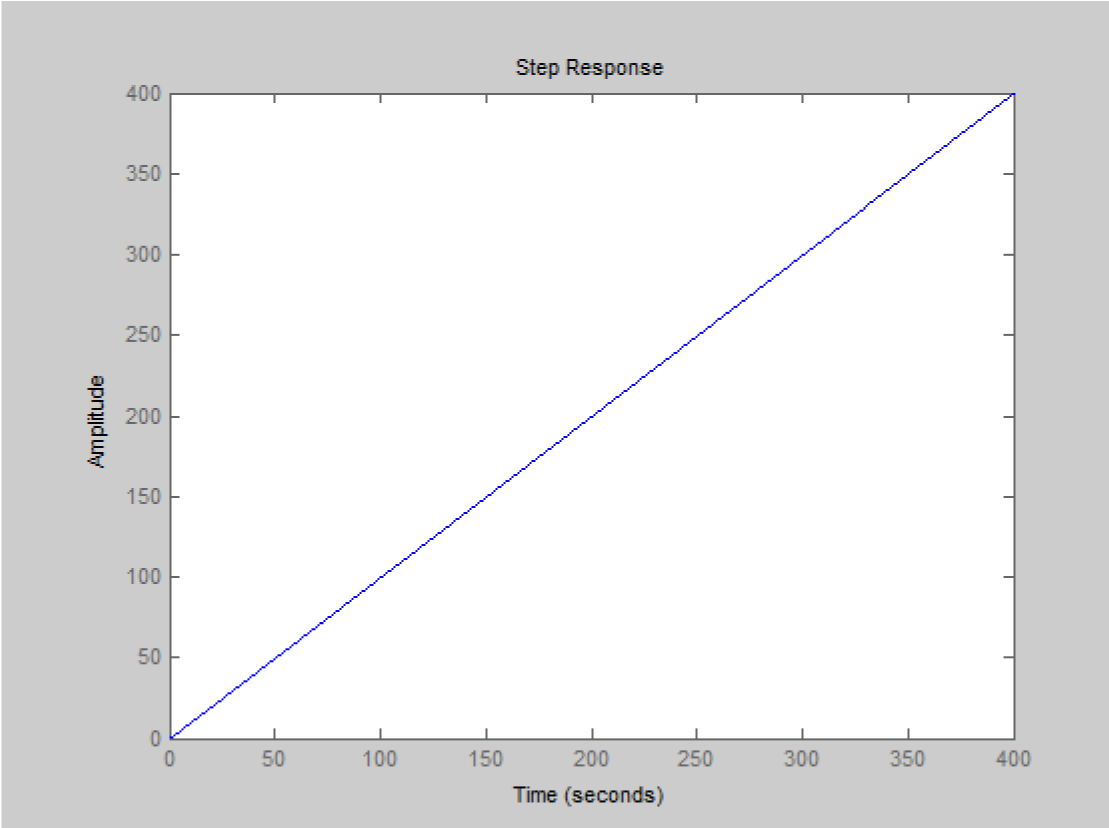
h =

١

----

s (s^٢ + s + ١)

ج: >> step(sys)



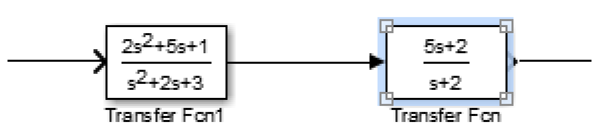
۳-تابع تبدیل زیر را درنظر بگیرید. مطلوبست:نمایش فضای حالت این سیستم.

$H_{(s)}=\frac{s+۶}{s^{۲}+۵s+۶}$

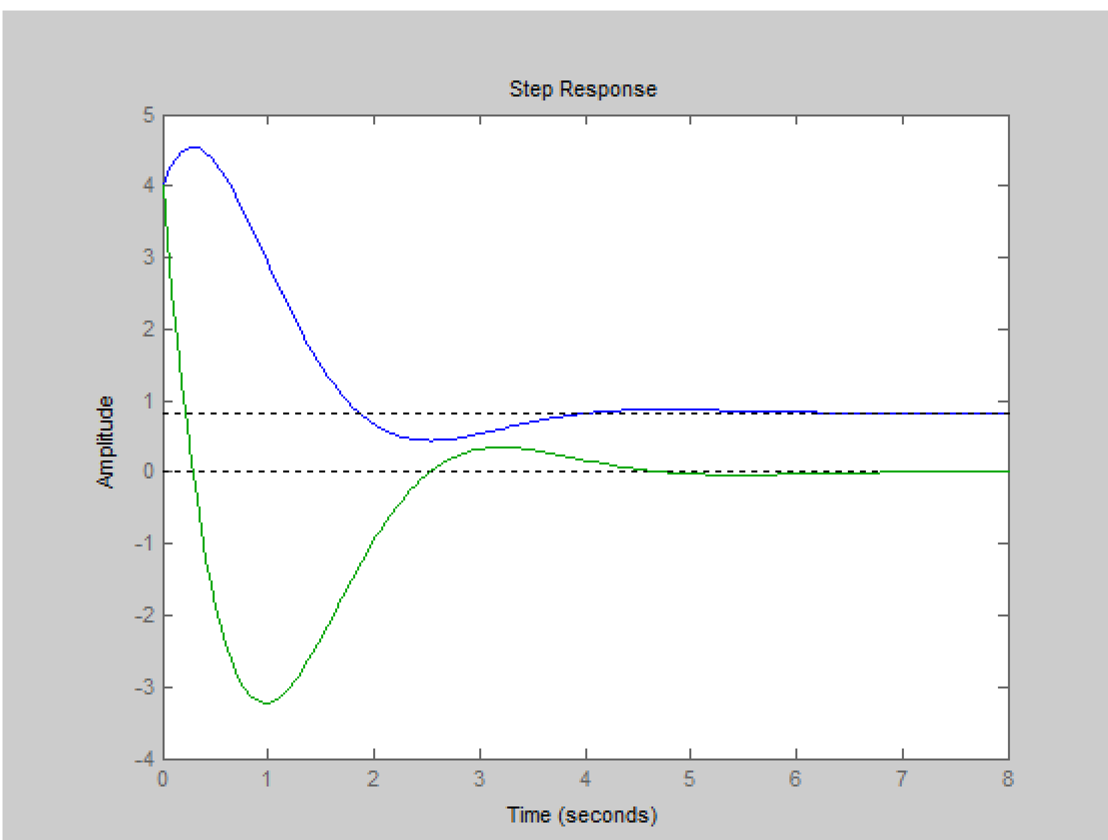
```
>> num=[۱ ۶];

>> den=[۱ ۵ ۶];

>> sys=tf(num,den)

sys =

      s + ۶

  -------------

s^۲ + ۵ s + ۶
```

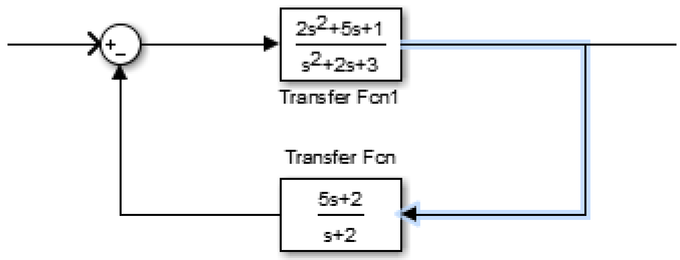
```
>> e=ss(sys)
e =
  a =
      x۱  x۲
   x۱  -۵  -۳
   x۲   ۲   ۰
  b =
      u۱
   x۱   ۲
   x۲   ۰
  c =
       x۱   x۲
   y۱  ۰,۵  ۱,۵
  d =
      u۱
y۱   ۰
```

٤-چنانچه دو تابع $G_{(s)} = \frac{۲s^۲+۵s+۱}{s^۲+۲s+۳}$ و $H_{(s)} = \frac{۵s+۲}{s+۲}$ موجود باشند؛ حاصل فیدبک، اتصال سری و موازی این دو تابع تبدیل را بیابید. همچنین در محیط سیمولینگ نیز طراحی شود. در پایان، حالت های سیستم به ورودی پله و ضربه را در هر دو حالت (سیمولینگ و Mفایل) بررسی کنید.

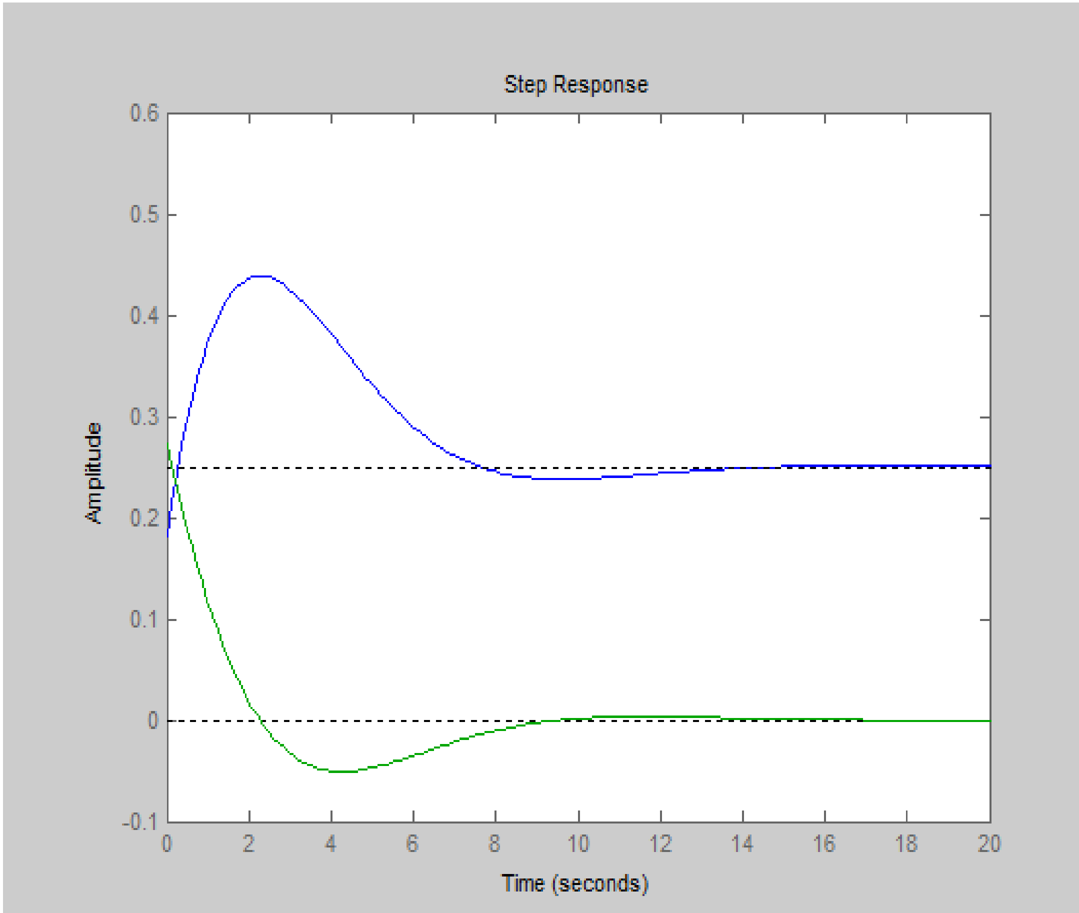


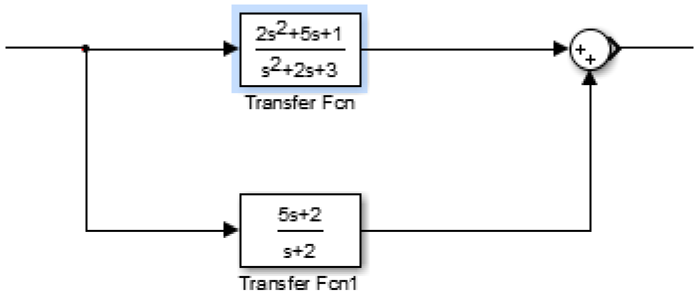
```
>> numg١=[٢ ٥ ١];

>> deng١=[١ ٢ ٣];

>> numg٢=[٢ ٥];

>> deng٢=[١ ٢];

>> [num,den]=series(numg١,deng١,numg٢,deng٢);

>> sys=tf(num,den)

sys =

 ١٠ s^٣ + ٢٩ s^٢ + ١٥ s + ٢

 -------------------------

   s^٣ + ٤ s^٢ + ٧ s + ٦

>> step(sys)

>> hold all

>> impulse(sys)
```



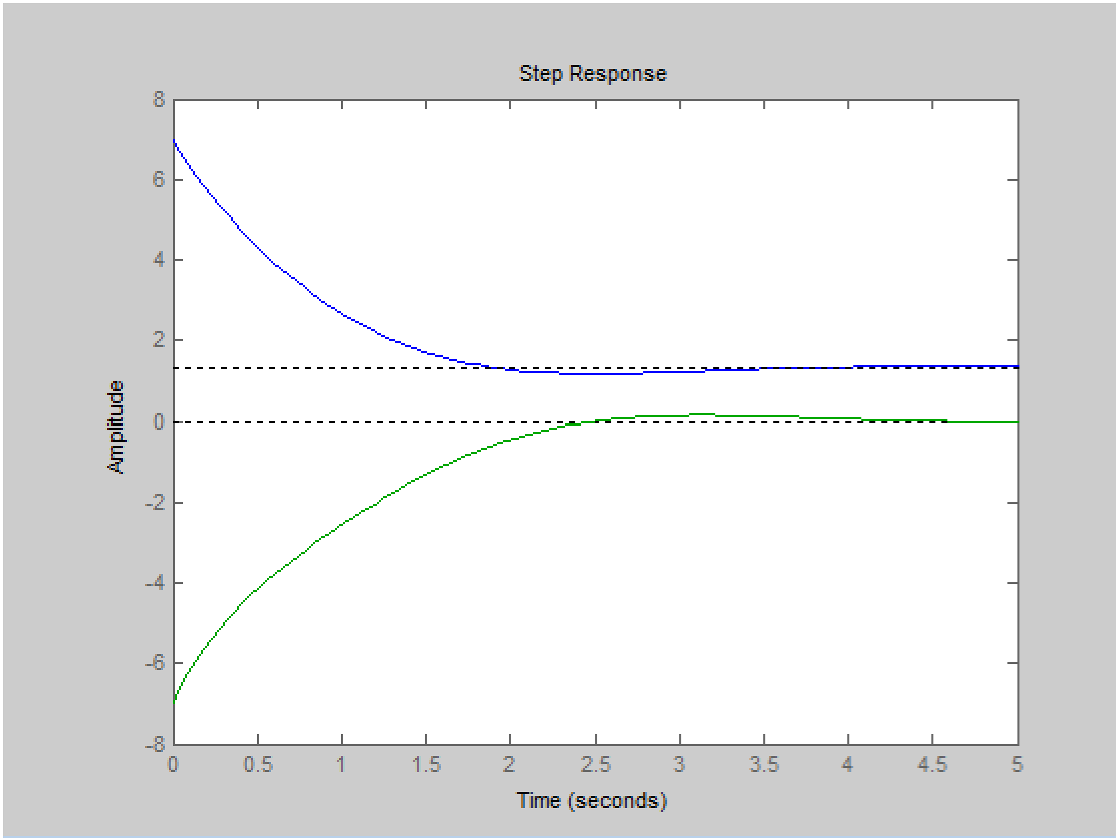


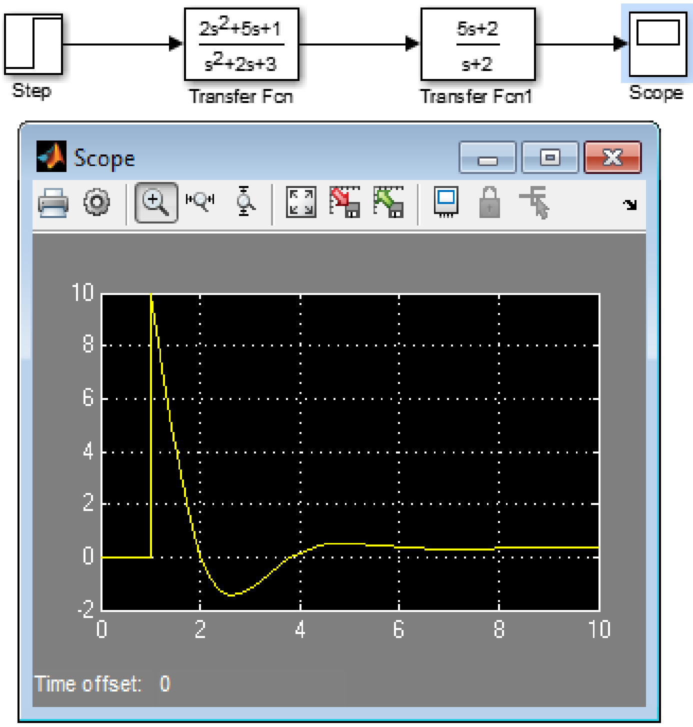

```
>> numg١=[٢ ٥ ١];
>> deng١=[١ ٢ ٣];
>> numg٢=[٥ ٢];
>> deng٢=[١ ٢];
>> [num,den]=feedback(numg١,deng١,numg٢,deng٢);
>> sys=tf(num,den)
sys =
   ٢ s^٣ + ٩ s^٢ + ١١ s + ٢
  --------------------------
  ١١ s^٣ + ٣٣ s^٢ + ٢٢ s + ٨
>> step(sys)
>> hold all
>> impulse(sys)
```

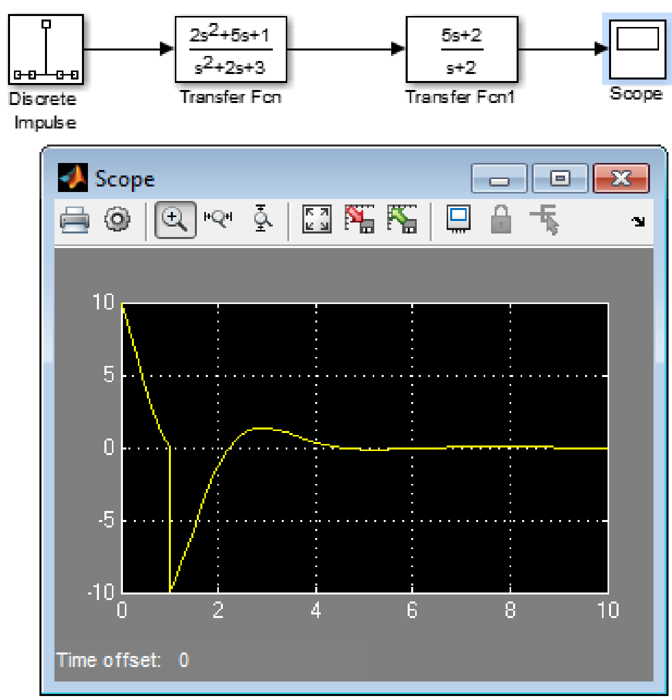

Step Response
Amplitude
Time (seconds)
0.6
0.5
0.4
0.3
0.2
0.1
0
-0.1
0
2
4
6
8
10
12
14
16
18
20

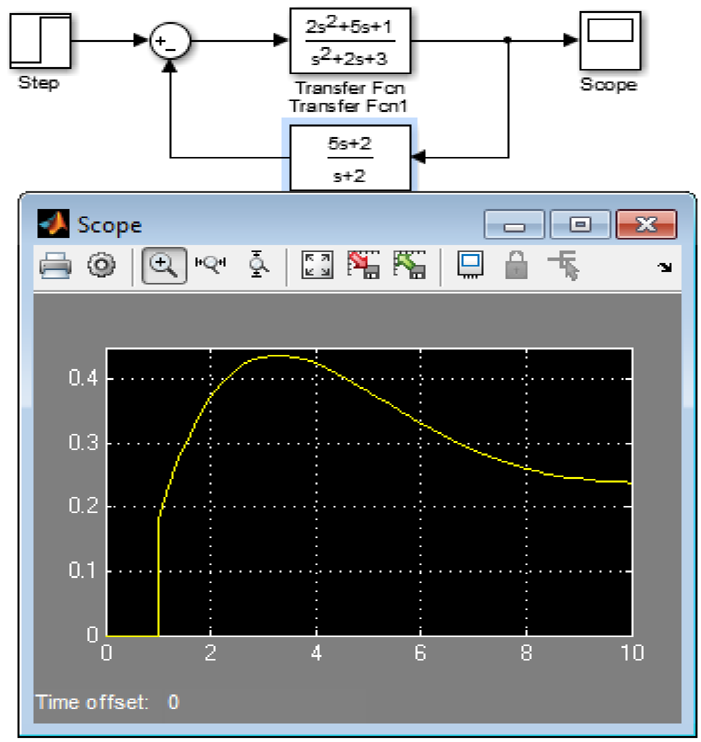


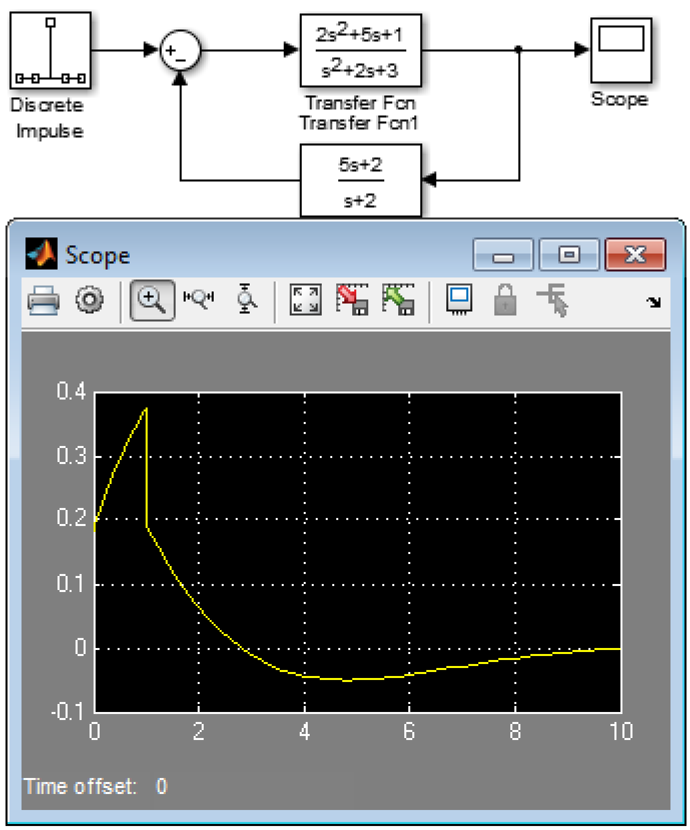

```
>> numg١=[٢ ٥ ١];
>> deng١=[١ ٢ ٣];
>> numg٢=[٥ ٢];
>> deng٢=[١ ٢];
>> [num,den]=parallel(numg١,deng١,numg٢,deng٢);
>> sys=tf(num,den)
sys =
 ٧ s^٣ + ٢١ s^٢ + ٣٠ s + ٨
 -------------------------
   s^٣ + ٤ s^٢ + ٧ s + ٦

>> step(sys)
>> hold all
>> impulse(sys)
```

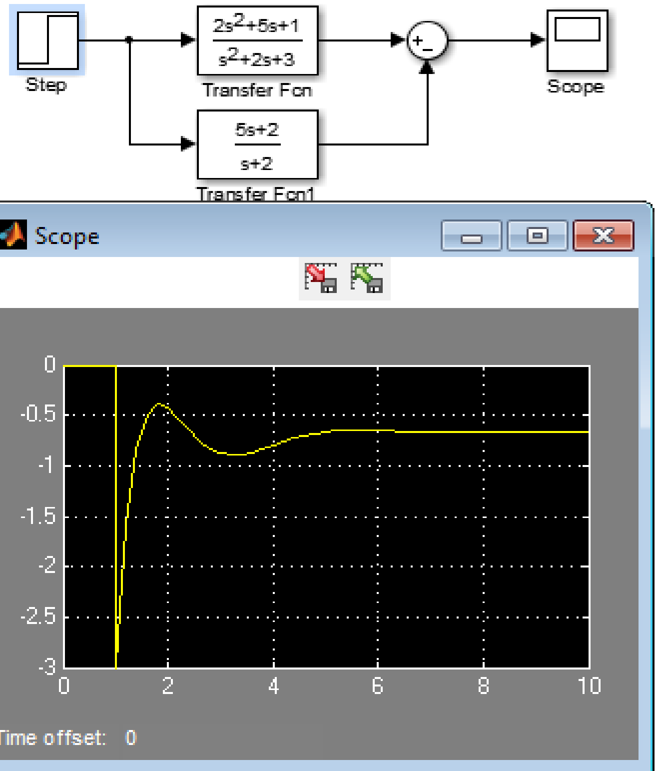

Step Response
Amplitude
Time (seconds)
8
6
4
2
0
-2
-4
-6
-8
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5

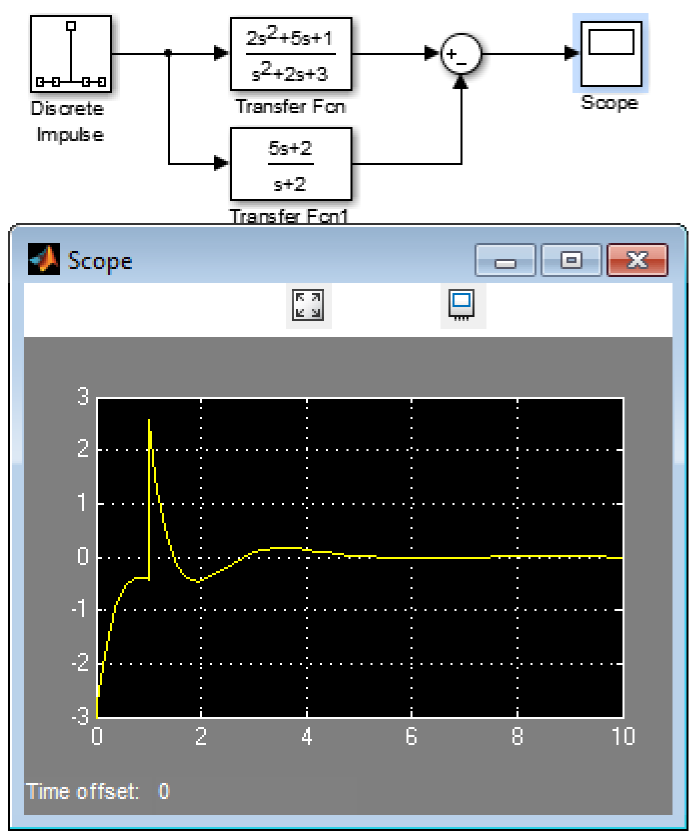

Step
2s2+5s+1
s2+2s+3
Transfer Fcn
5s+2
s+2
Transfer Fcn1
Scope
Scope
10
8
6
4
2
0
-2
0
2
4
6
8
10
Time offset: 0


Discrete
Impulse
2s2+5s+1
s2+2s+3
Transfer Fcn
5s+2
s+2
Transfer Fcn1
Scope
Scope
10
5
0
-5
-10
0
2
4
6
8
10
Time offset: 0


Step
2s2+5s+1
s2+2s+3
Transfer Fcn
Transfer Fcn1
Scope
5s+2
s+2
Scope
0.4
0.3
0.2
0.1
0
0
2
4
6
8
10
Time offset: 0

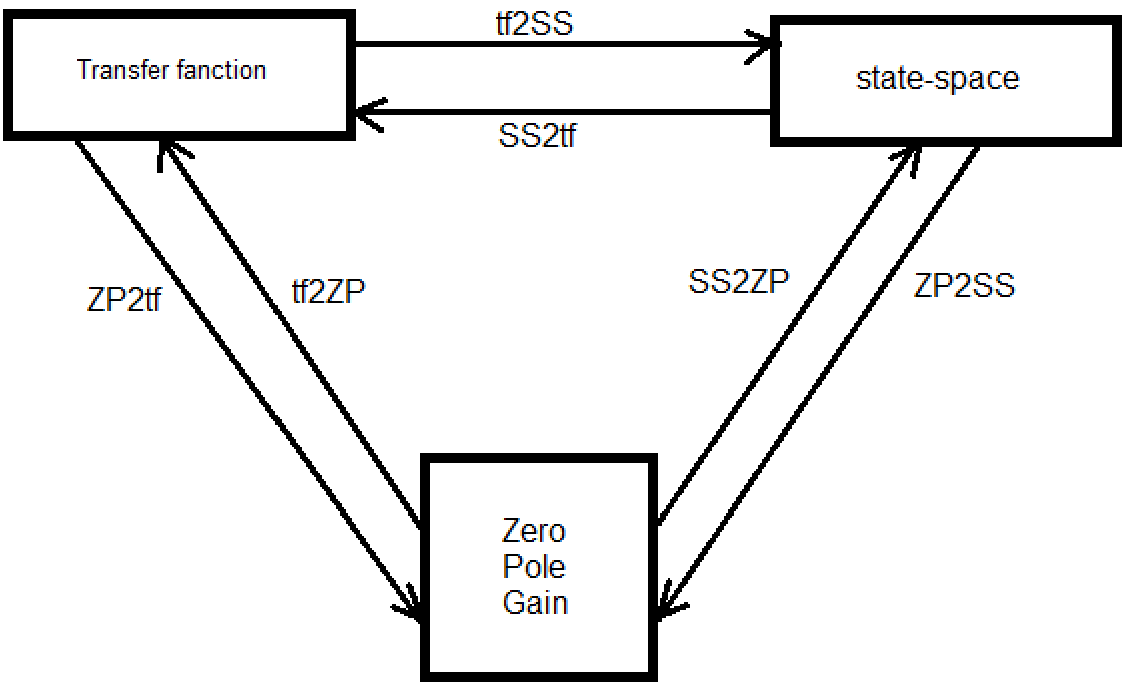

Discrete
Impulse
2s2+5s+1
s2+2s+3
Transfer Fcn
Transfer Fcn1
Scope
5s+2
s+2
Scope
0.4
0.3
0.2
0.1
0
-0.1
0
2
4
6
8
10
Time offset: 0


Step
2s2+5s+1
s2+2s+3
Transfer Fcn
5s+2
s+2
Transfer Fcn1
Scope
Scope
Time offset: 0


Discrete Impulse
2s2+5s+1
s2+2s+3
Transfer Fcn
5s+2
s+2
Transfer Fcn1
Scope
Scope
Time offset: 0

تبدیل مدل ها به یکدیگر:

تبدیل فضای حالت به تابع تبدیل	$[num, den]$=SS۲tf(A,B,C,D)
تبدیل صفر و قطب به فضای حالت	$[A, B, C, D]$=ZP۲SS(Z,P,K)
تبدیل فضای حالت به صفر و قطب	$[Z, P, K]$=SS۲ZP(A,B,C,D)
تبدیل تابع تبدیل به فضای حالت	$[A, B, C, D]$=tf۲SS(num,den)
تبدیل تابع تبدیل به صفر و قطب	$[Z, P, K]$=tf۲ZP(num,den)
تبدیل صفر و قطب به تابع تبدیل	$[num, den]$=ZP۲tf(Z,P,K)



پاسخ فرکانسی سیستم ها:

چنانچه از قبل، یک تابع تبدیل مانند $H_{(s)}$ تعریف شده باشد، می توان با استفاده از دستورات زیر نمودار پاسخ فرکانسی آن سیستم را رسم کرد.

rlocus(G):رسم مکان هندسی ریشه ها

nyquist(G) :رسم دیاگرام نایکوئیست سیستم

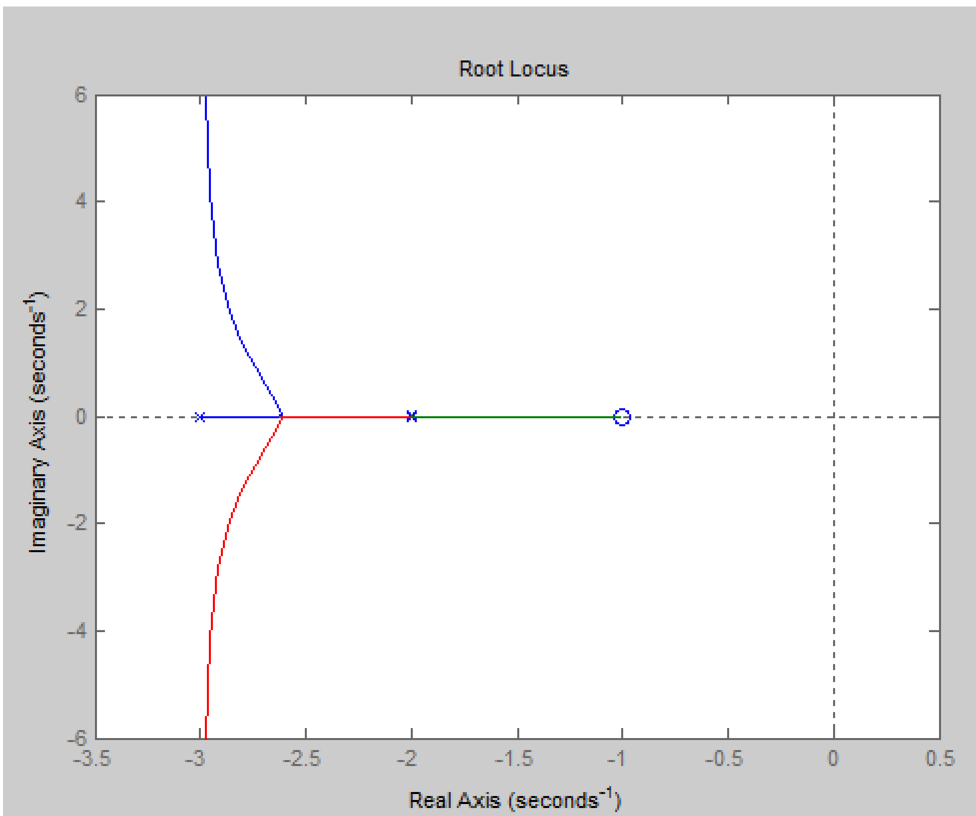
bode(G) :رسم دیاگرام بد سیستم

Nichols(G):رسم دیاگرام نیکولز سیستم

مثال:مکان هندسی ریشه های سیستم زیر را رسم کنید.

$$H_{(s)}=\frac{s+۱}{(s+۲)^{۲}(s+۳)}$$

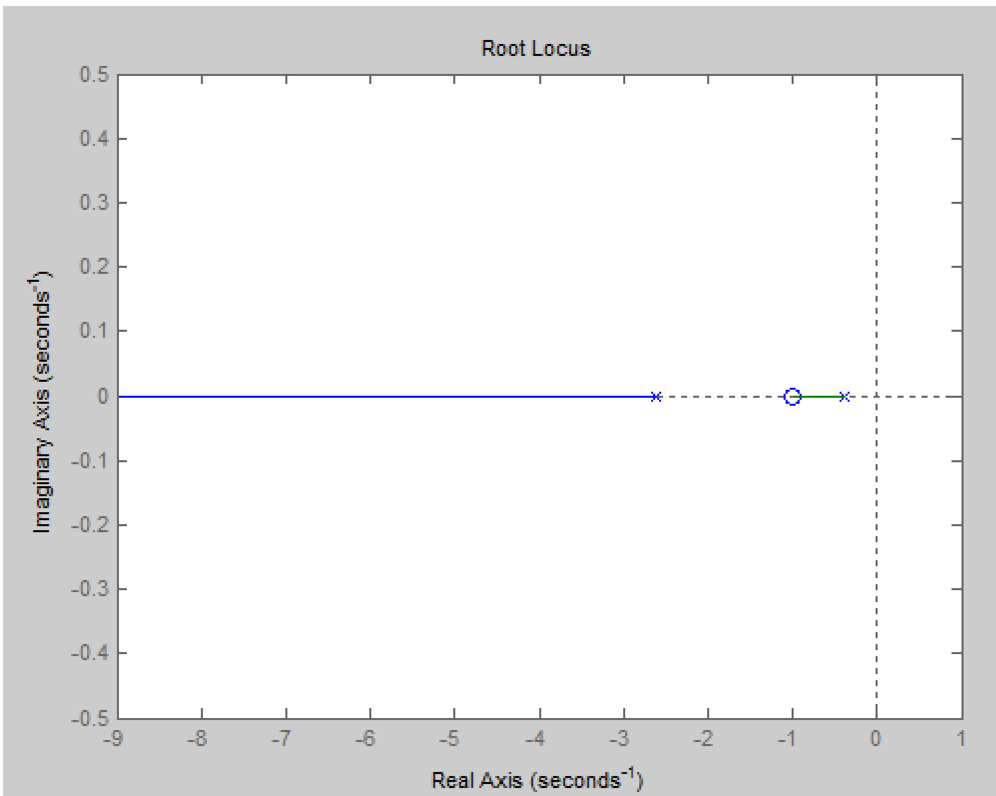
```
>> s=tf('s');
>> d=(s+۱)/(((s+۲)^۲)*(s+۳));
>> rlocus(d)
```



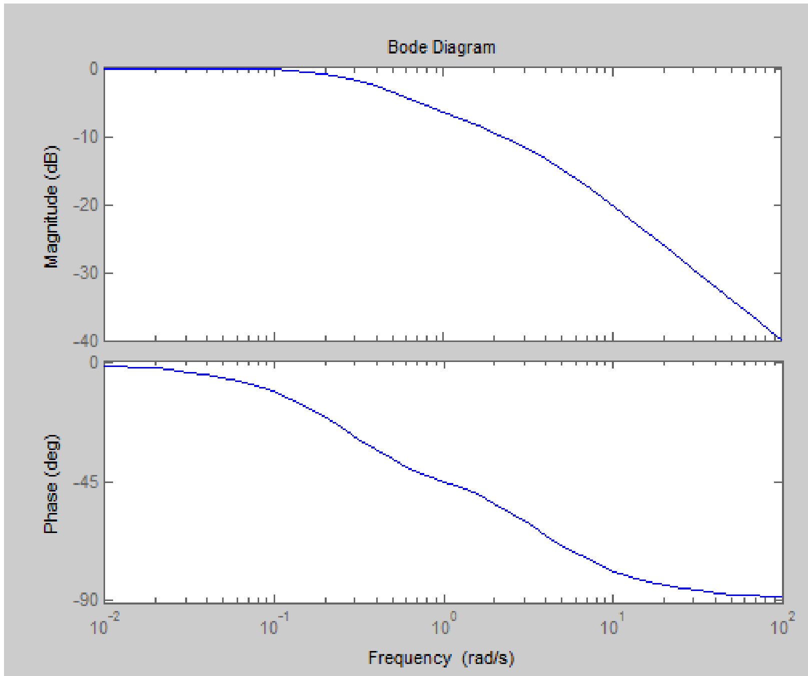
مثال)مکان هندسی ریشه ها، دیاگرام بد ونایکوئیست سیستم زیر را به دست آورید.

$$H_{(s)}=\frac{s+۱}{s^{۲}+۳s+۱}$$

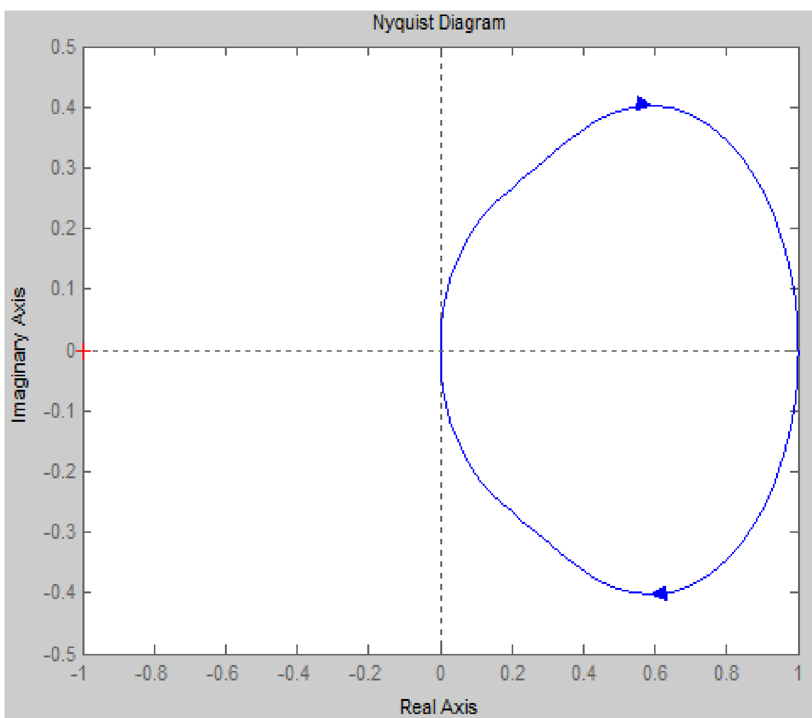
```
>> num=[١ ١];
>> den=[١ ٣ ١];
>> sys=tf(num,den);
>> rlocus(sys)
```



>> bode(sys)



>> nyquist(sys)



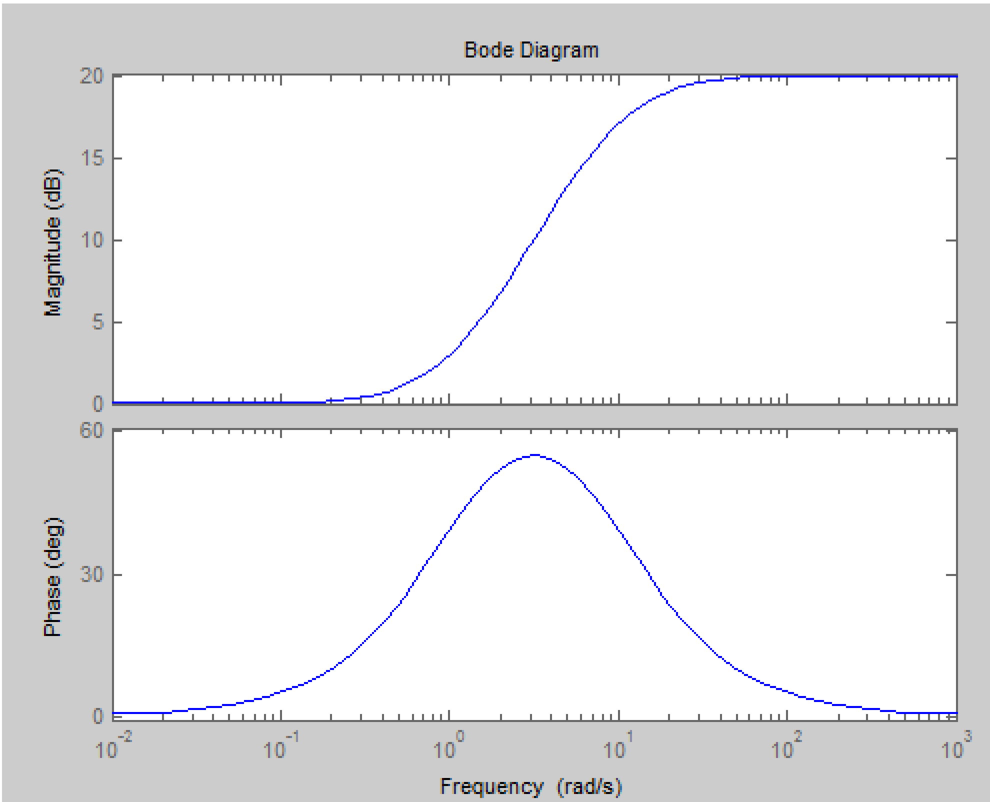
مثال)دیاگرام بد سیستم های زیر را رسم کنید.

$$H_{(s)}=\frac{10(s+1)}{s+10}$$

>> s=tf('s');

>> k=(10*(s+1))/(s+10);

>> bode(k)

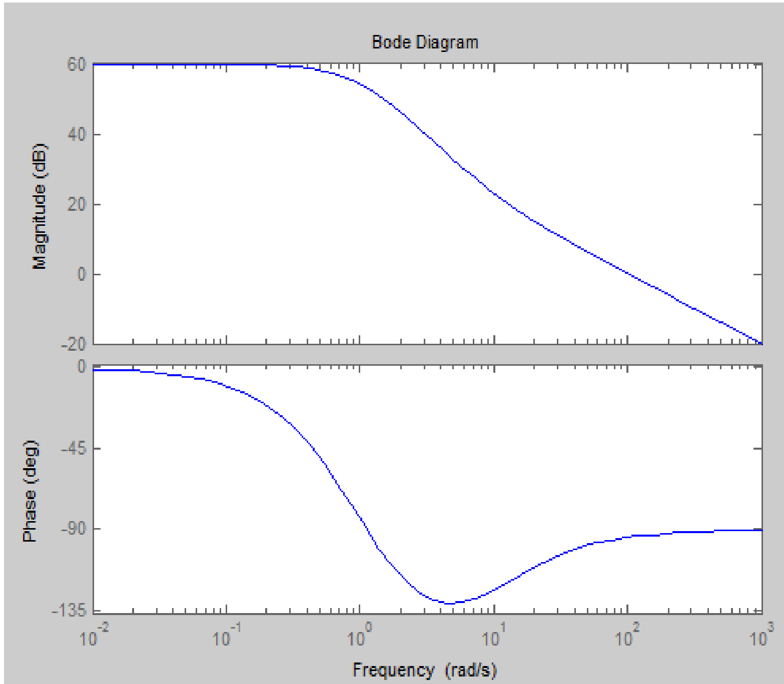


$$H_{(s)}=\frac{100(s+10)}{(s+1)^2}$$

>> s=tf('s');

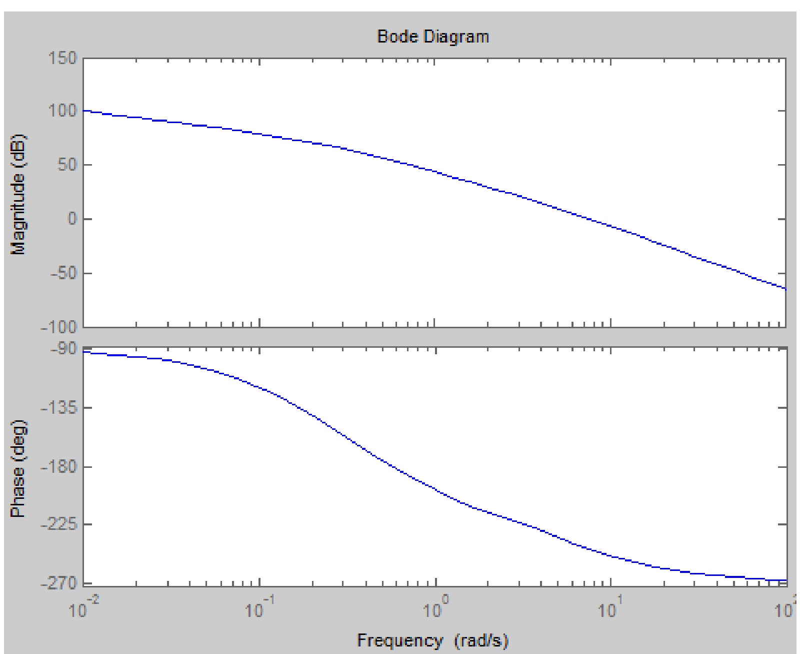
>> o=(100*(s+10))/((s+1)^2);

>> bode(o)



$$H_{(s)}=\frac{٥٠٠(s+٢)}{s(s+١)(s^{٢}+٥s+١)}$$

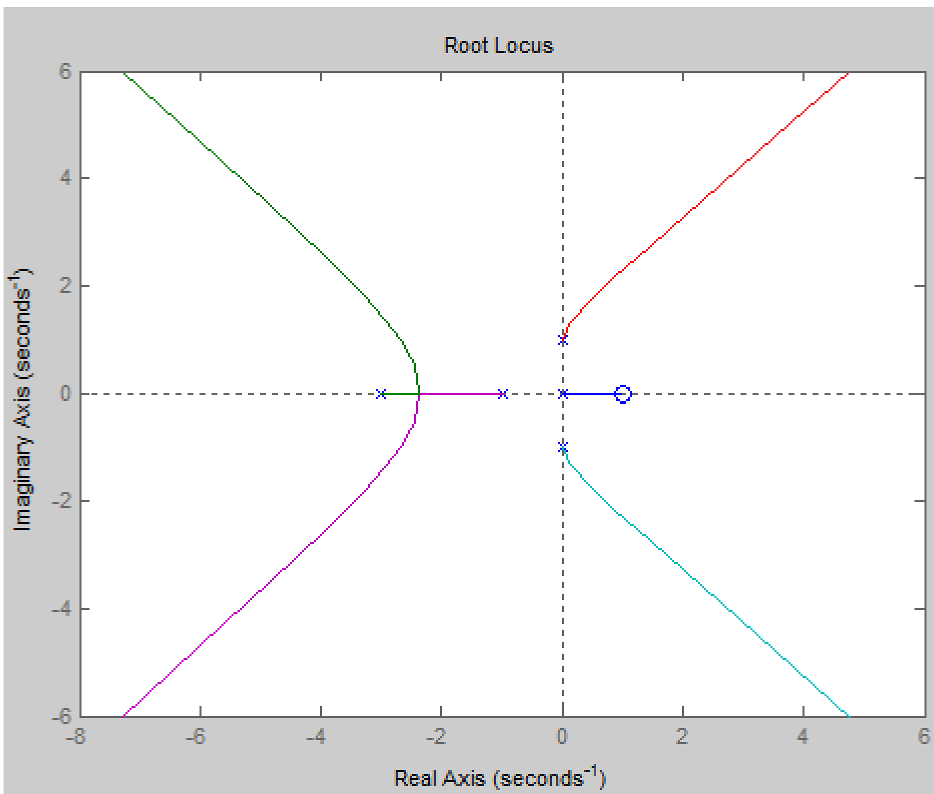
```
>> s=tf('s');
>> q=(٥٠٠*(s+٢))/(s*(s+١)*(s^٢+٥*s+١));
>> bode(q)
```



مثال) مکان هندسی ریشه های سیستم های زیر را به دست آورید.

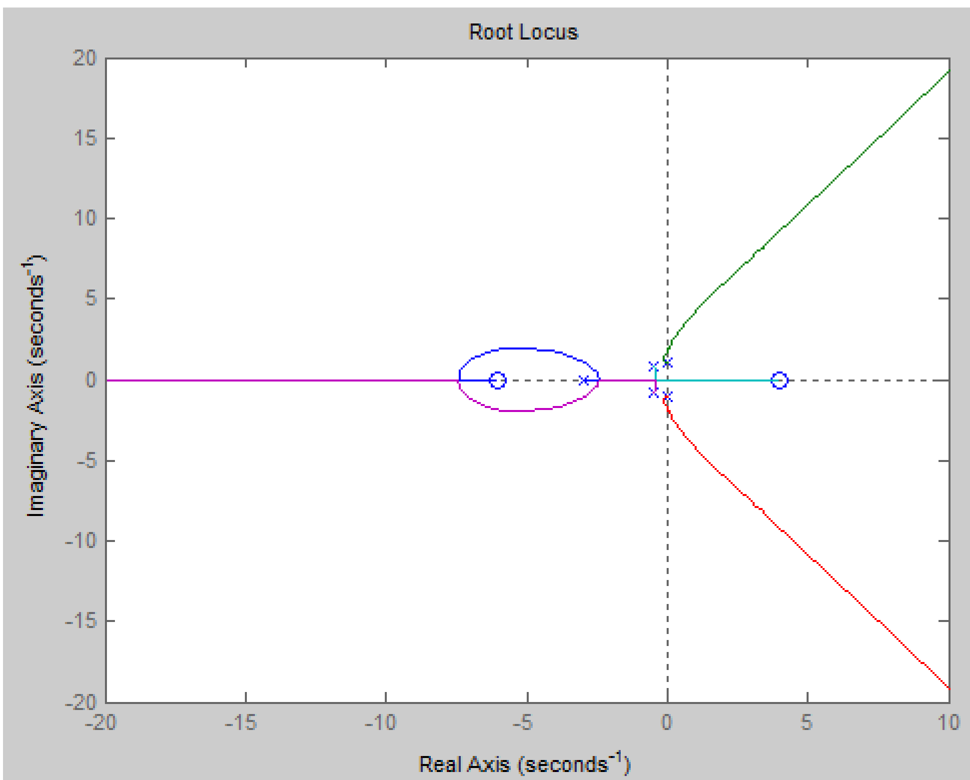
$$H_{(s)}=\frac{(s-۱)}{s(s+۱)(s^{۲}+۱)(s+۳)}$$

```
>> s=tf('s');
>> m=(s-۱)/(s*(s+۱)*((s^۲)+۱)*(s+۳));
>> rlocus(m)
```

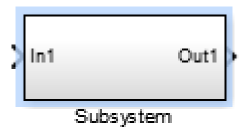


$H_{(s)}=\frac{(٦+s)(s-٤)}{(s^٢+١)(s+٣)(s^٢+s+١)}$

```
>> s=tf('s');
>> b=((٦+s)*(s-٤))/(((s^٢)+١)*(s+٣)*((s^٢)+s+١));
>> rlocus(b)
```



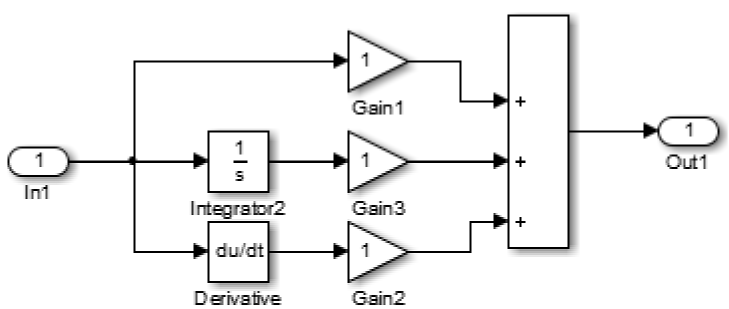
کنترل کننده PID



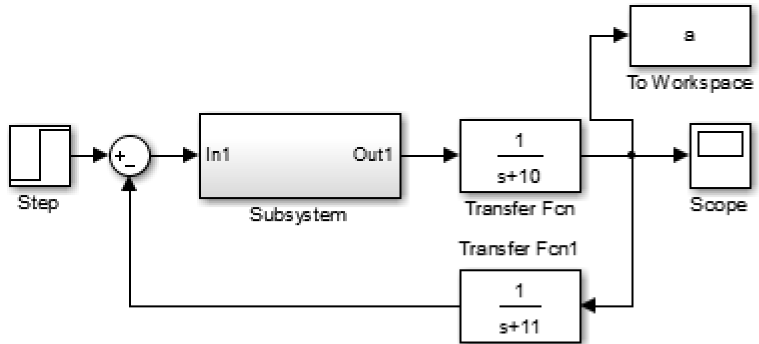
همان طور که در ابتدای جزوه توضیحاتی در مورد کنترل کننده های PID بیان شده است اکنون می خواهیم آن ها در محیط سیمولینگ بررسی کنیم.

برای تعیین ضرایب K_pوK_iوK_dمدار زیر را که متناسب با ویژگی های کنترل کننده PID می باشد را در بلوک دیاگرام Subsystem قرار می دهیم.

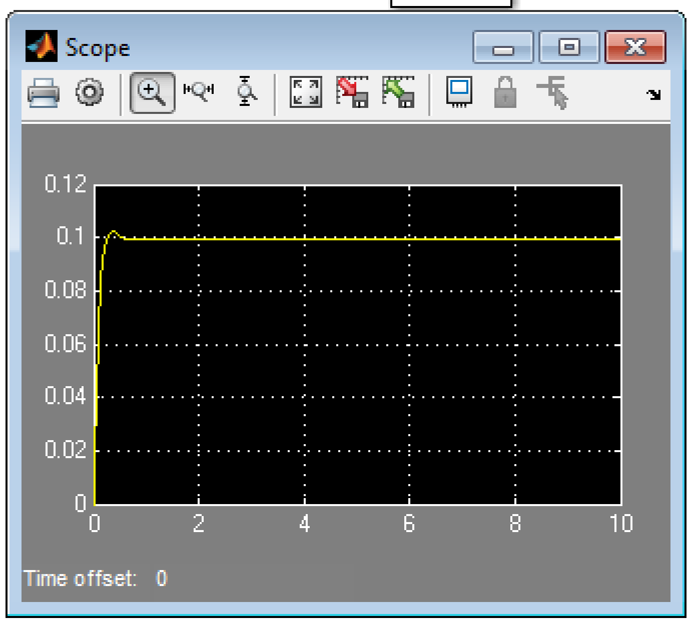
در این مدار باید با تنظیم دستی gainها به شکل موج مورد نظر برسیم.



مثال)

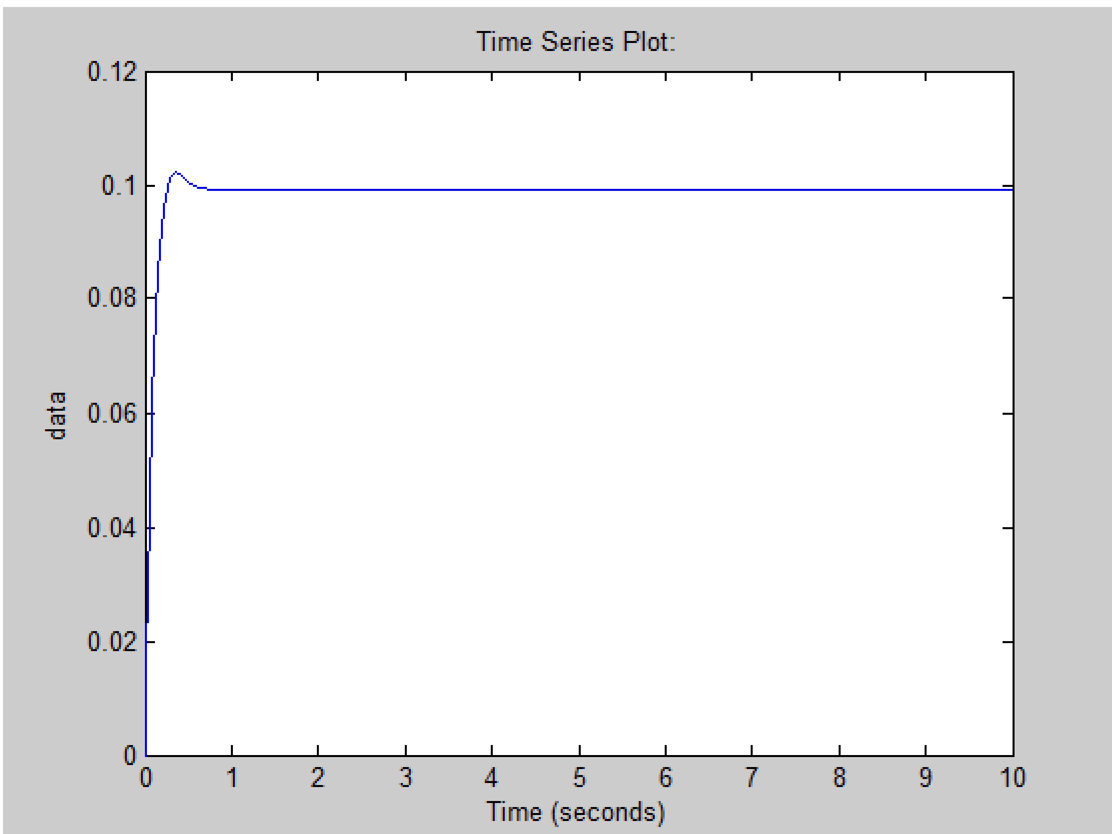


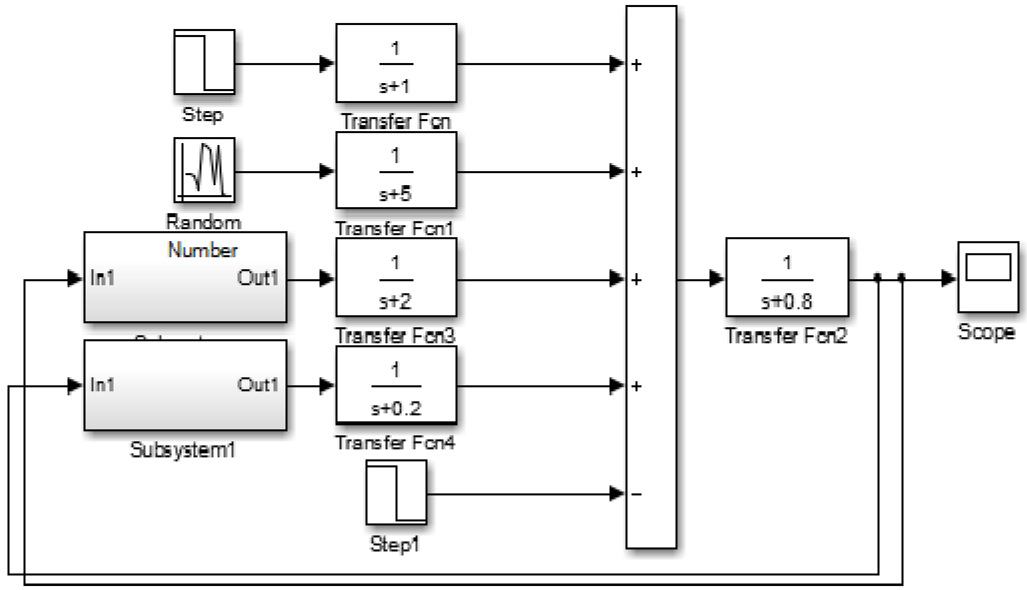
a
To Workspace
Step
In1
Out1
Subsystem
1
s+10
Transfer Fcn
Scope
Transfer Fcn1
1
s+11

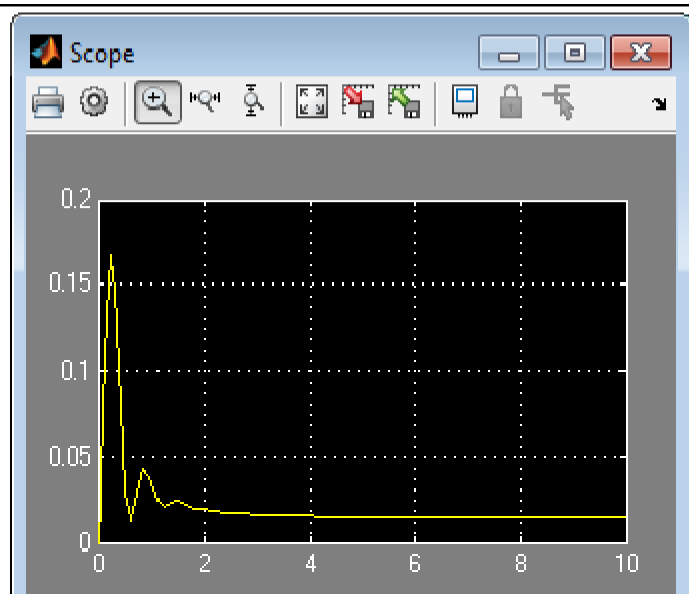


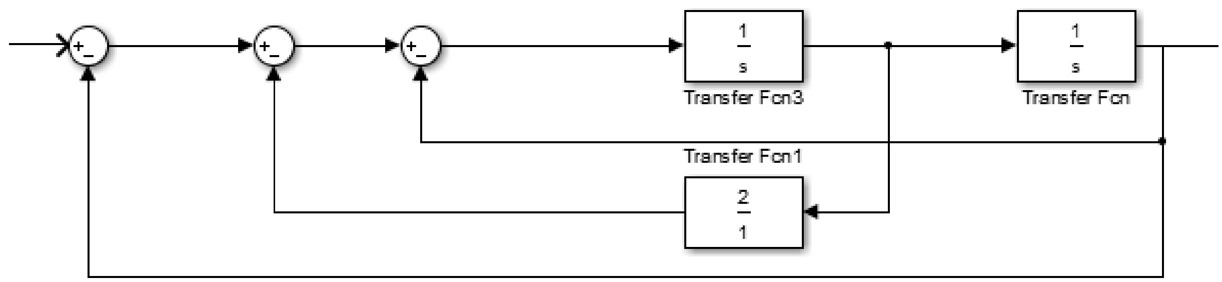
Scope
0.12
0.1
0.08
0.06
0.04
0.02
0
0
2
4
6
8
10
Time offset: 0

```
>> plot(a)
```

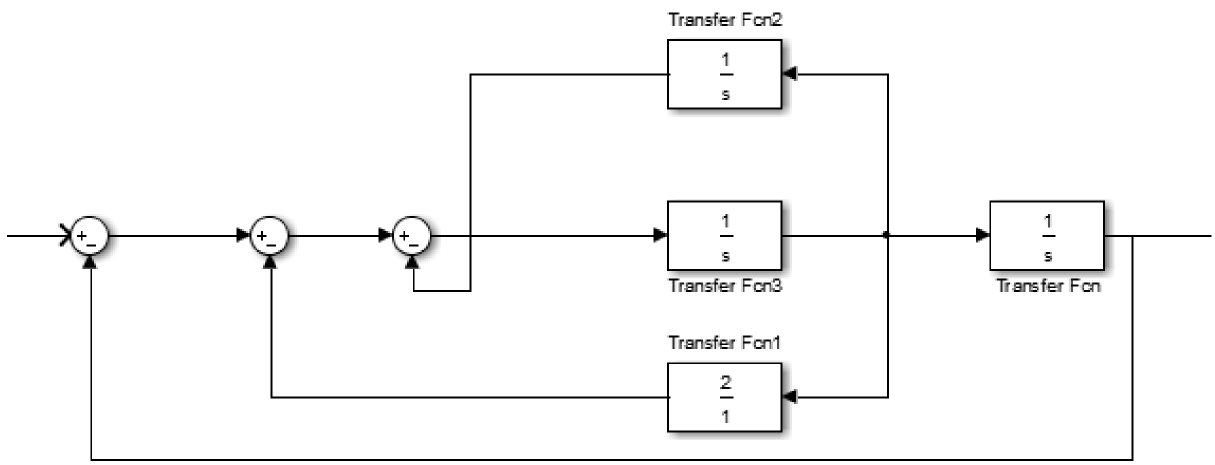

Time Series Plot:
data
Time (seconds)
0.12
0.1
0.08
0.06
0.04
0.02
0
0
1
2
3
4
5
6
7
8
9
10


Step
1
s+1
Transfer Fcn
Random
Number
1
s+5
Transfer Fcn1
In1
Out1
1
s+2
Transfer Fcn3
In1
Out1
Subsystem1
1
s+0.2
Transfer Fcn4
Step1
+
+
+
+
-
1
s+0.8
Transfer Fcn2
Scope


Scope
0.2
0.15
0.1
0.05
0
0
2
4
6
8
10



ساده سازی سیستم:

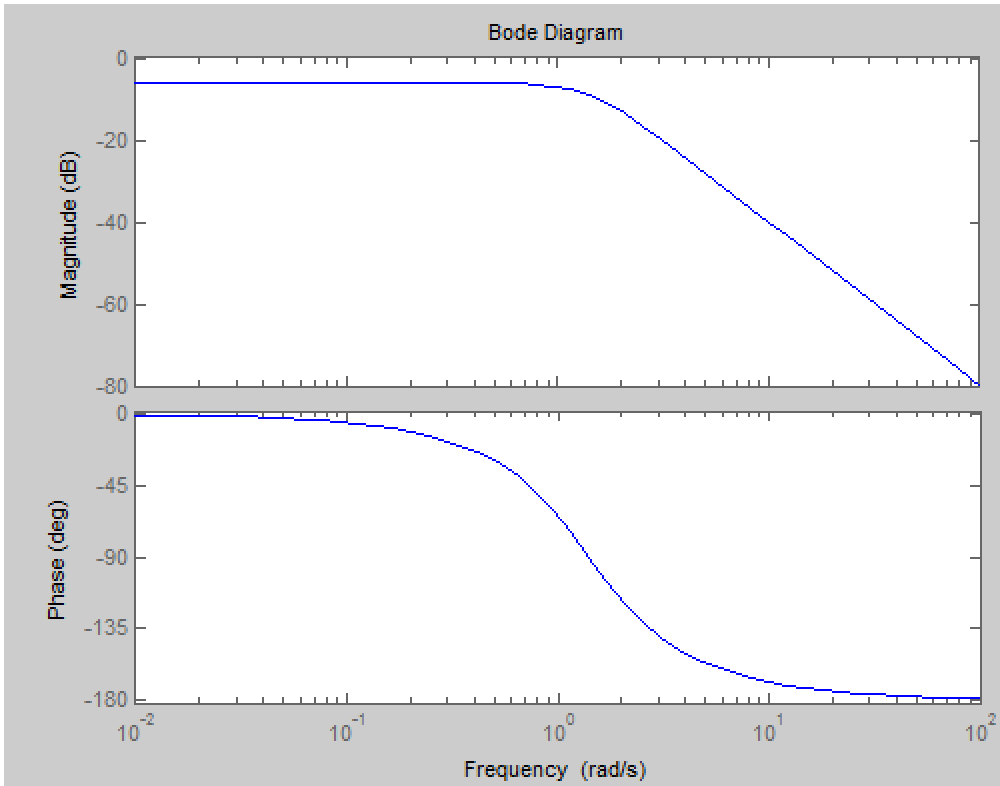



```
>> numg١=[١];
deng١=[١ ٠];
numg٢=[٢];
deng٢=[١];
[nums,dens]=feedback(numg١,deng١,numg١,deng١);
c=tf(nums,dens)
c =
    s
  -------
  s^٢ + ١
```

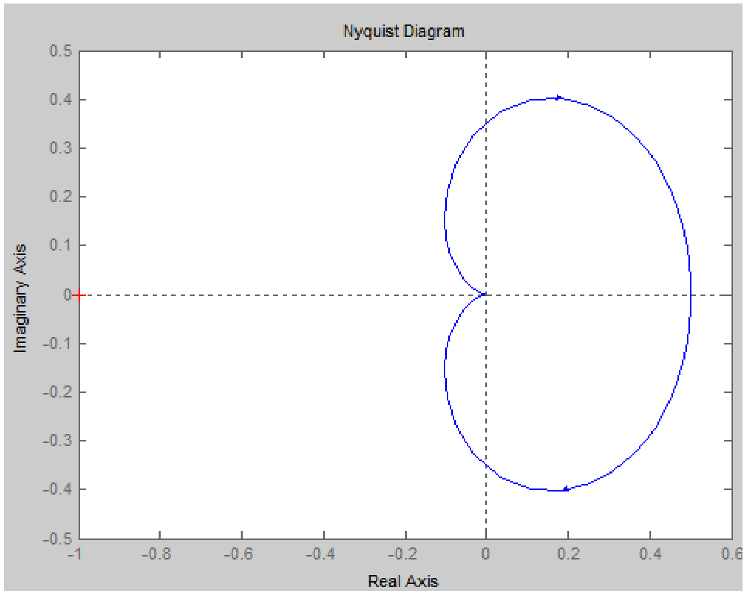
```
num=[١ ٠];
den=[١ ٠ ١];
[num,den]=feedback(num,den,numg٢,deng٢);
i=tf(num,den)
i =
        s
  -------------
  s^٢ + ٢ s + ١
numg٣=[١ ٠];
deng٣=[١ ٢ ١];
[num١,den١]=series(numg١,deng١,numg٣,deng٣);
p=tf(num١,den١)
p =
         s
  ---------------
  s^٣ + ٢ s^٢ + s
>> numg٤=[١ ٠];
>> deng٤=[١ ٢ ١ ٠];
>> [num٢,den٢]=cloop(numg٤,deng٤,-١);
>> o=tf(num٢,den٢)
o =
          s
  -----------------
  s^٣ + ٢ s^٢ + ٢ s
```

```
>> num=[١ ٠];
>> den=[١ ٢ ٢ ٠];
>> [a,b,c,d]=tf٢ss(num,den)
a =
  -٢   -٢    ٠
   ١    ٠    ٠
   ٠    ١    ٠
b =
   ١
   ٠
   ٠
c =
   ٠    ١    ٠
d =
  ٠
>> [z,p,k]=tf٢zp(num,den)
z =
   ٠
p =
  ٠,٠٠٠٠ + ٠,٠٠٠٠i
 -١,٠٠٠٠ + ١,٠٠٠٠i
 -١,٠٠٠٠ - ١,٠٠٠٠i
k =
   ١
```

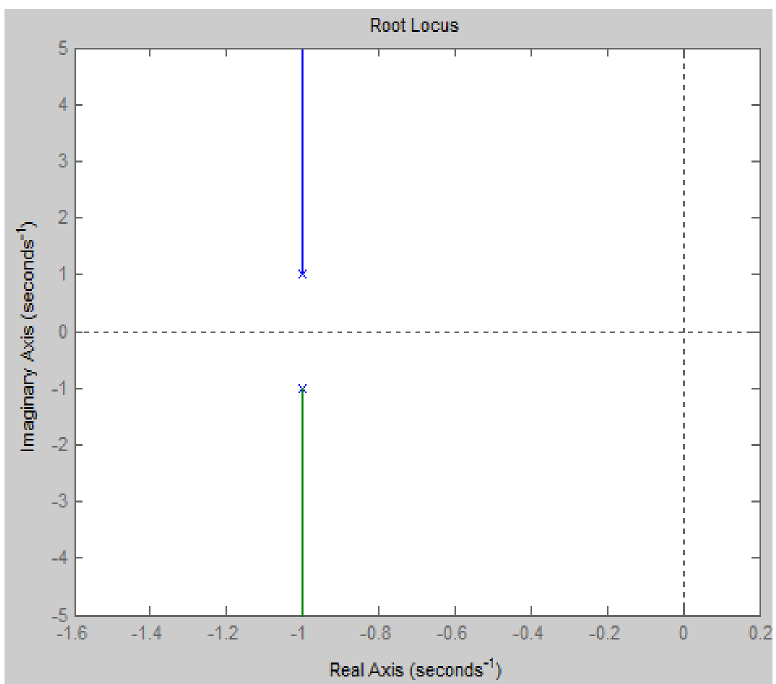
```
>> i=tf(num,den);
>> bode(i)
```

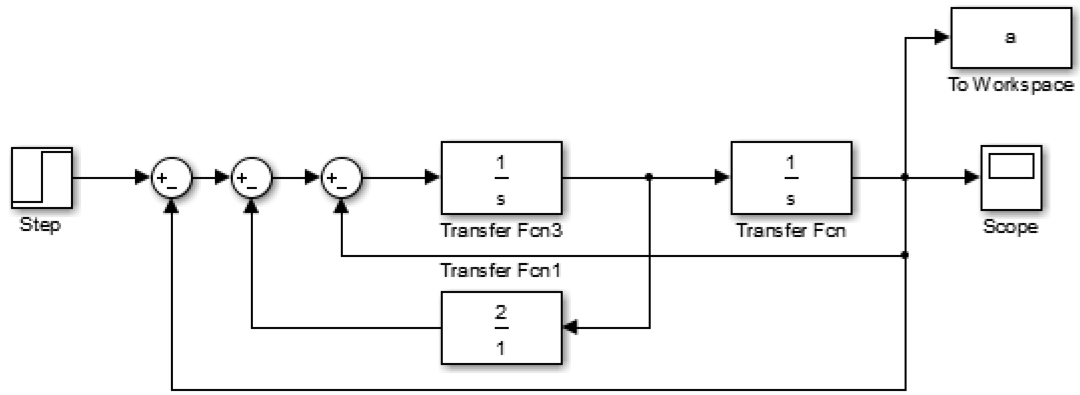


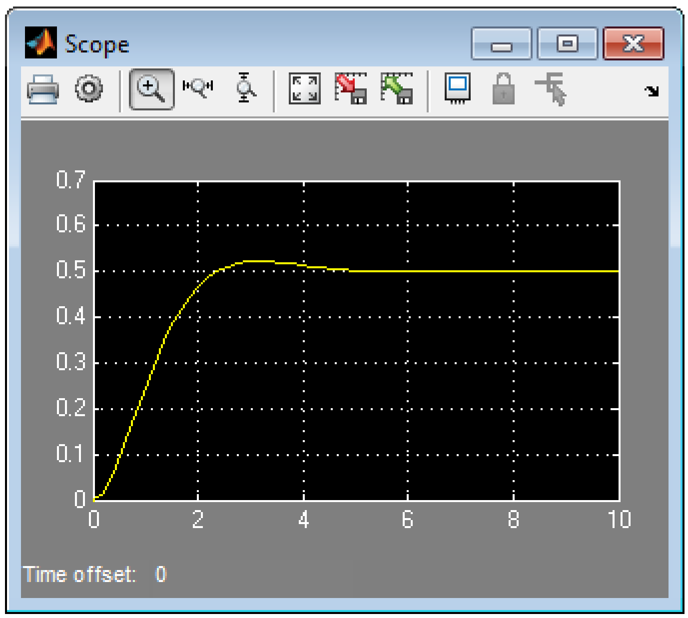
```
>> nyquist(i)
```



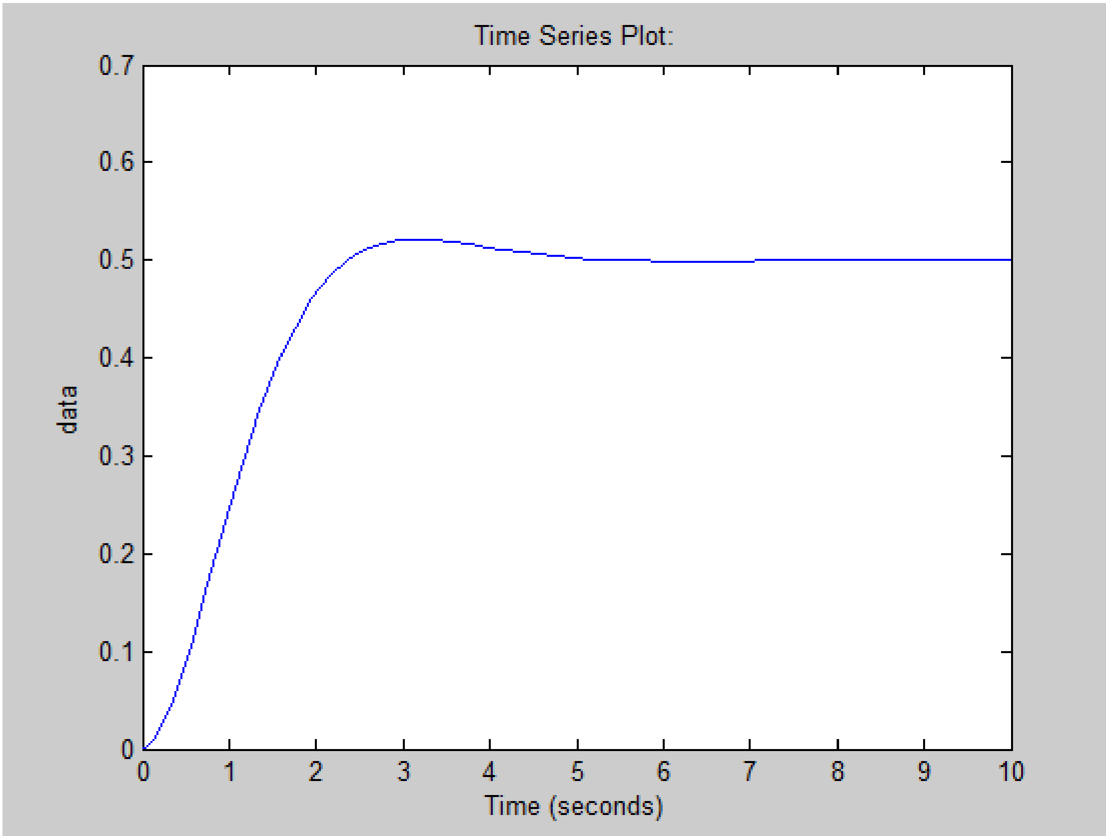

```
>> rlocus(i)
```




a
To Workspace
Step
Transfer Fcn3
Transfer Fcn
Scope
Transfer Fcn1
1
s
2
1

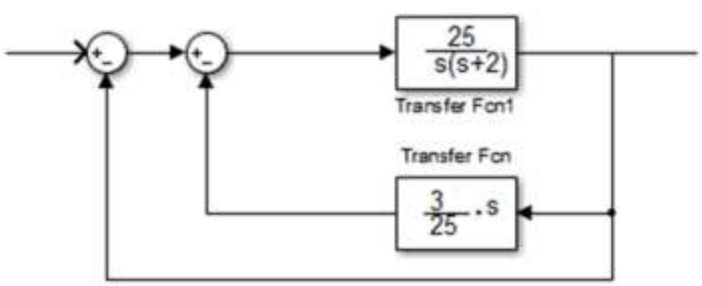

Scope
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
2
4
6
8
10
Time offset: 0

>> plot(a)



Time Series Plot:
data
Time (seconds)

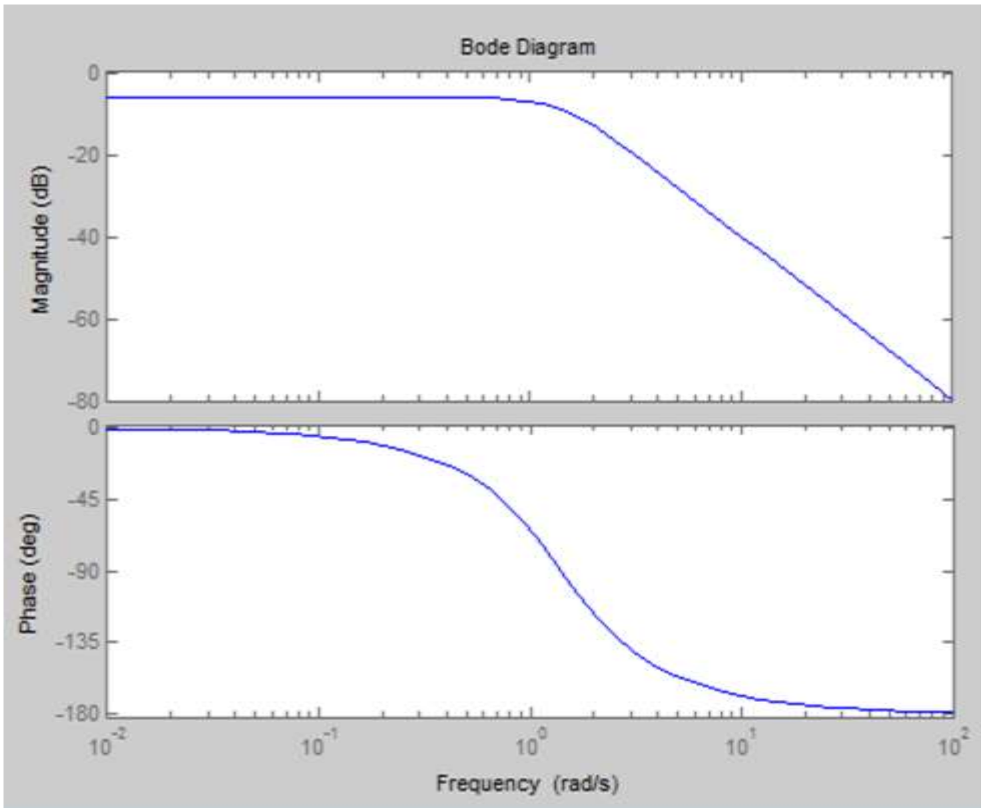
مثال:



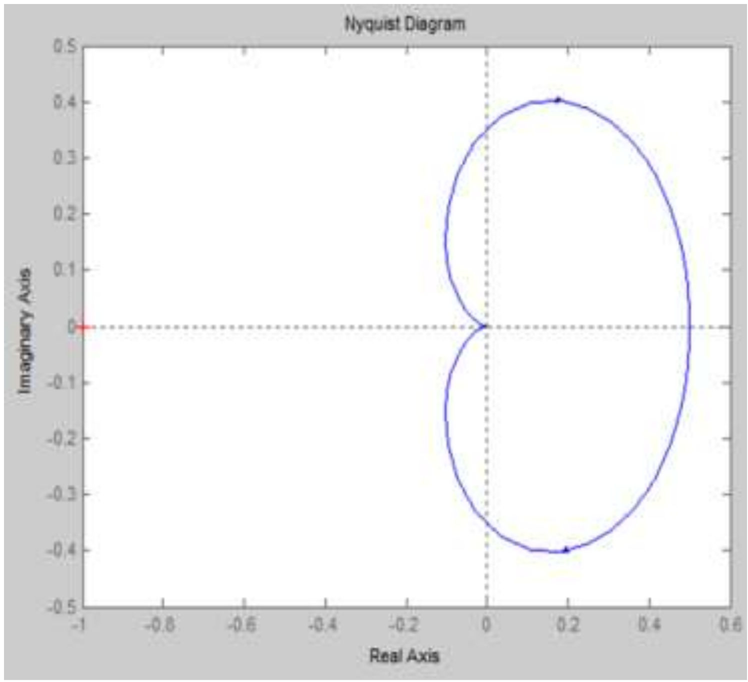
25
s(s+2)
Transfer Fcn1
Transfer Fcn
3
25
.s

```
>> numg١=[٢٥];
>> deng١=[١ ٢ ٠];
>> numg٢=[٣ ٠];
>> deng٢=[٢٥];
>> sys=tf(numg١,deng١);
>> sys١=tf(numg٢,deng٢);
>> [num,den]=feedback(sys,sys١);
>> a=tf(num,den)
a =
       ٦٢٥
  --------------
  ٢٥ s^٢ + ١٢٥ s
>> num=[٦٢٥];
>> den=[٢٥ ١٢٥ ٠];
>> [num,den]=cloop(num,den,-١);
>> s=tf(num,den)
s =
      ٦٢٥
  --------------------
  ٢٥ s^٢ + ١٢٥ s + ٦٢٥
>> num=[٦٢٥];
>> den=[٢٥ ١٢٥ ٦٢٥];
>> [a,b,c,d]=tf٢ss(num,den)
```

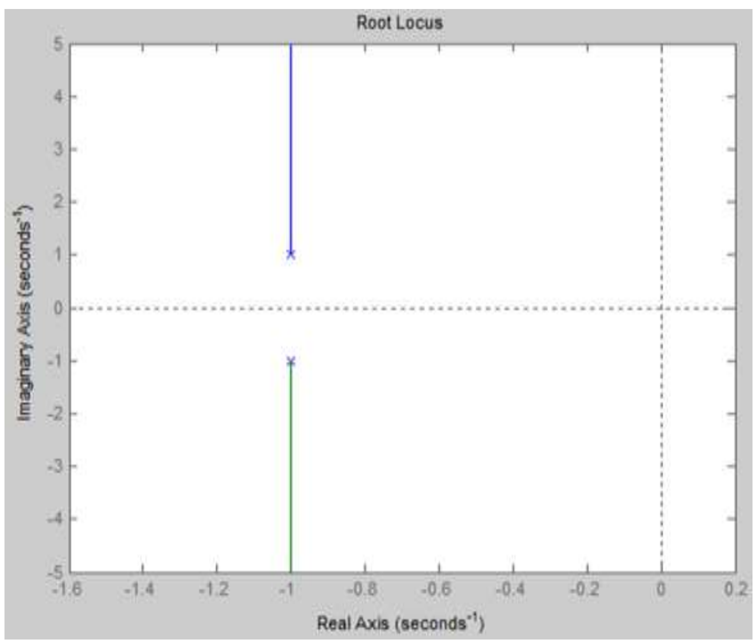
```
a =

   -٥   -٢٥

    ١     ٠

b =

    ١

    ٠

c =

    ٠    ٢٥

d =

    ٠

>> [z,p,k]=tf٢zp(num,den)

z =

   Empty matrix: ٠-by-١

p =

  -٢,٥٠٠٠ + ٤,٣٣٠١i

  -٢,٥٠٠٠ - ٤,٣٣٠١i

k =

    ٢٥

>> bode(s)
```


Bode Diagram
Magnitude (dB)
0
-20
-40
-60
-80
Phase (deg)
0
-45
-90
-135
-180
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}
Frequency (rad/s)

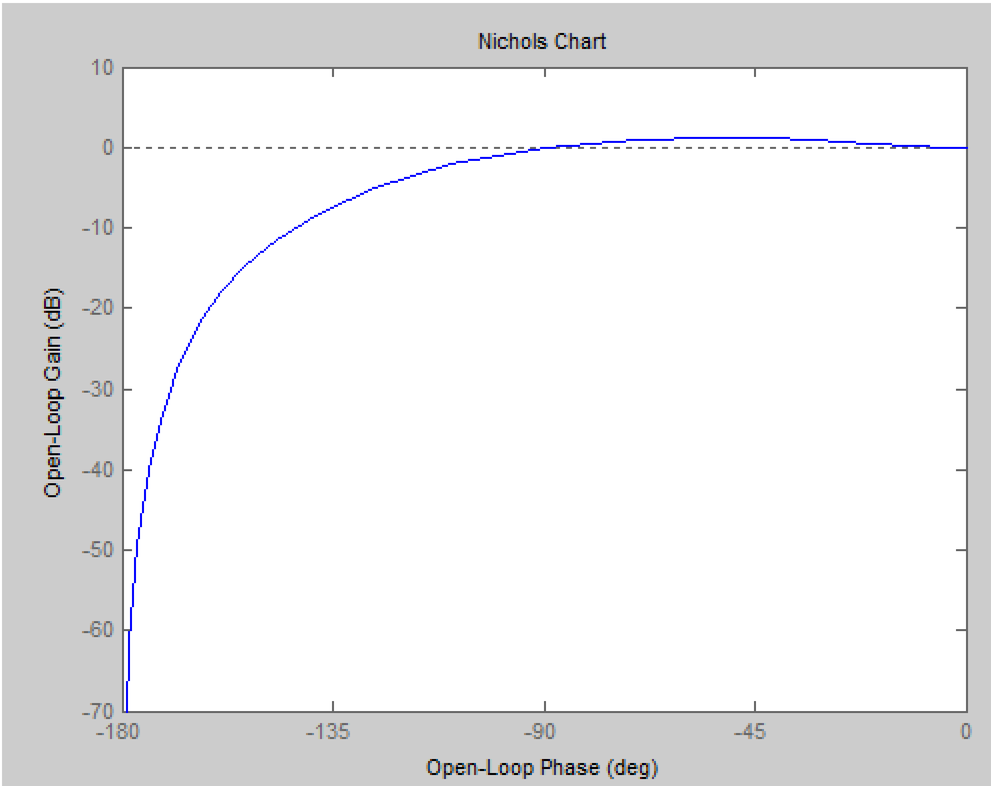
```
>> nyquist(s)
```


Nyquist Diagram
Imaginary Axis
Real Axis
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6

>> rlocus(s)


Root Locus
Imaginary Axis (seconds⁻¹)
Real Axis (seconds⁻¹)
5
4
3
2
1
0
-1
-2
-3
-4
-5
-1.6
-1.4
-1.2
-1
-0.8
-0.6
-0.4
-0.2
0
0.2

```
>> nichols(s)
```



Nichols Chart
Open-Loop Gain (dB)
Open-Loop Phase (deg)
10
0
-10
-20
-30
-40
-50
-60
-70
-180
-135
-90
-45
0