**به نام خداوند جان و خرد**

**مجموعه گزارش کارهای آز – کنترل خطی**

**تهیه کنندگان :**

**میلاد جلالی (گروه 13:30 تا 16)**

**عباس اعمی نیکبخت(گروه 16 تا 18:20)**

**استاد راهنما :**

**دکتر یوسفی**

**دانشکده ی فنی انقلاب اسلامی تهران**



**خرداد ماه 91**

**تجزیه و تحلیل سیستم های مرتبه اول و دوم**

**سیستم مرتبه ی اول :** یک سیستم مرتبه ی اول را با تابع تبدیل G(s) = را در نظر می گیریم که در آن K بهره ی سیستم است . اگر ورودی سیـستم پله ی واحد باشد، خواهیم داشت :

C(s) = G(s).R(s)

R(s) =

C(s) = × = ⎼

C(t) = ⎼ = (1 ⎼ ) t>0

را ثابت زمانی سیستم می گویند که هر چه ثابت زمانی سیستم کوچکتر باشد پاسخ آن سریعتر است و سرعت سیستم بیشتر خواهد بود. برای به دست آوردن پاسخ حالت ماندگار از قضیه ی مقدار نهایی استفاده می کنیم.

*= = × × =*

*حال با استفاده از نرم افزار متلب سیستم را تعریف کرده و با تعریف K و تحت عنوان ورودی پاسخ سیستم را به ورودی پله ی واحد به ازای مقادیر مختلف خواهیم دید. ابتدا مراحل اولیه را برای توصیف و معرفی سیستم به نرم افزار در محیط m-File انجام می دهیم :*

clear all;clc;

K=input('please quantity K value of first-order system = ');

a0=input('please define your pole position (a0) = ');

numG=[K];

denG=[1 a0];

Gs=tf(numG,denG)

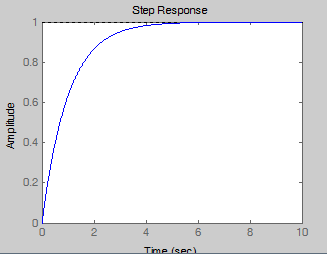
disp(Gs);

t=0:0.01:10;

step(Gs,t)

*با اجرای برنامه متلب دو مقدار k و a0 را از ما درخواست می کند که پس از وارد کردن این دو ، علاوه بر نمایش تابع تبدیل و مشخصات تابع تبدیل، پاسخ سیستم به ورودی پله ی واحد را نیز نمایش می دهد :*

*please quantity K value of first-order system = 1*

*please define your pole position (a0) = 1*

*Transfer function:*

*1*

*-----*

*s + 1*

*tf*

*Properties:*

*num: {[0 1]}*

*den: {[1 1]}*

*Variable: 's'*

*ioDelay: 0*

*InputDelay: 0*

*OutputDelay: 0*

*Ts: 0*

*TimeUnit: ''*

*InputName: {''}*

*InputUnit: {''}*

*InputGroup: [1x1 struct]*

*OutputName: {''}*

*OutputUnit: {''}*

*OutputGroup: [1x1 struct]*

*Name: ''*

*Notes: {}*

*UserData: []*

*Methods, Superclasses*

**سیستم مرتبه ی دوم :** تابع تبدیل یک سیستم مرتبه ی دوم به صورت G(s) = است. پاسخ این سیـــستم به ورودی پله ی واحد در حالت ماندگار برابر است با :

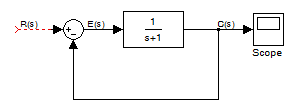
*= = × × =*

پاسخ حالت گذرای یک سیستم مرتبه ی دوم به ورودی پله ی واحد به یکی از سه صورت زیر است :

1. پاسخ میرایی شدید (overdamped Response) .
2. پاسخ میرایی بحرانی (critical Response) .
3. پاسخ میرایی ضعیف (underdamped Response) .

یک سیستم مرتبه ی دوم نمونه با تابع تبدیل حلقه بسته ی G(s) = = را در نظر می گیریم که در آن فرکانس نامیرای طبیعی و نسبت میرایی است.

تابع تبدیل حلقه باز از رابطه ی = به دست می آید. بلوک دیاگرام سیستم حلقه بسته با فیدبک واحد نیز در شکل زیر آمده است :



معادله مشخصه ی سیستم برابر است با :

= = 0

ــ چنانچه 0 <  < 1 باشد، سیستم میرای ضعیف خواهد بود و دو ریشه ی مختلط خواهد داشت :

*= ⎼*

*= را فرکانس طبیعی میرا شونده می گویند.*

*ــ در صورتی که >1 باشد، سیستم میرای شدید خواهد بود و دو ریشه ی حقیقی منفی خواهد داشت :*

*= ⎼*

*ــ* *اگر که شود، سیستم میرای بخرانی خواهد بود و یک ریشه ی مضاعف و حقیقی خواهد داشت :*

*= ⎼*

ــ در صورتی که باشد، سیستم نوسانی بوده و دو ریشه ی موهومی دارد :

*=*

*ــ اگر* -1 < < 0 سیستم ناپایدار بوده و دو قطب موهومی با جزء حقیقی مثبت دارد .

ــ *اگر که شود، دو قطب حقیقی مثبت داریم و سیستم ناپایدار است. (ریشه ی مضاعف).*

*ــ در صورتی که <1 باشد، دو قطب حقیقی مثبت و مختلف داریم و سیستم ناپایدار است.*

*با استفاده از برنامه ی نوشته شده در زیر که در محیط m-File برنامه ی متلب نوشته شده است، به راحتی با مقدار دهی می توان پاسخ سیستم و رسم مکان هندسی سیستم به ازای مقادیر مختلف از را به دست آورد.*

clear all;clc;

m=menu('Choose your transfer function model','zero-pole model','charecterisitc equation');

switch m

case{1}

k=input('enter the gain k = ');

n=menu('choose your number of zeros','0','1','2');

switch n

case{1}

l=menu('choose your number of poles','1','2','3');

switch l

case{1}

p0=input('enter your pole p0 = ');

Gs=zpk([],p0,k)

case{2}

p0=input('enter your pole p0 = ');

p1=input('enter your pole p1 = ');

Gs=zpk([],[p1,p0],k)

case{3}

p0=input('enter your pole p0 = ');

p1=input('enter your pole p1 = ');

p2=input('enter your pole p2 = ');

Gs=zpk([],[p2,p1,p0],k)

end

case{2}

l=menu('choose your number of poles','1','2','3');

switch l

case{1}

p0=input('enter your pole p0 = ');

z0=input('enter your zero z0 = ');

Gs=zpk([z0],[p0],k)

case{2}

p0=input('enter your pole p0 = ');

p1=input('enter your pole p1 = ');

z0=input('enter your zero z0 = ');

Gs=zpk([z0],[p1,p0],k)

case{3}

p0=input('enter your pole p0 = ');

p1=input('enter your pole p1 = ');

p2=input('enter your pole p2 = ');

z0=input('enter your zero z0 = ');

Gs=zpk([z0],[p2,p1,p0],k)

end

case{3}

l=menu('choose your number of poles','1','2','3');

switch l

case{1}

p0=input('enter your pole p0 = ');

z0=input('enter your zero z0 = ');

z1=input('enter your zero z1 = ');

Gs=zpk([z1,z0],[p0],k)

case{2}

p0=input('enter your pole p0 = ');

p1=input('enter your pole p1 = ');

z0=input('enter your zero z0 = ');

z1=input('enter your zero z1 = ');

Gs=zpk([z1,z0],[p1,p0],k)

case{3}

p0=input('enter your pole p0 = ');

p1=input('enter your pole p1 = ');

p2=input('enter your pole p2 = ');

z0=input('enter your zero z0 = ');

z1=input('enter your zero z1 = ');

Gs=zpk([z1,z0],[p2,p1,p0],k)

end

end

t=0:0.01:10;

o=menu('which one of diagrams would u like to present','Rlocus Diagram','Bode Diagram','Step Response','Nyquist Diagram','Nichols Diagram','step & Rlocus','all of Diagrams');

switch o

case{1}

rlocus(Gs)

case{2}

bode(Gs)

case{3}

step(Gs)

case{4}

nyquist(Gs)

case{5}

nichols(Gs)

case{6}

subplot(2,1,1);

step(Gs,t)

subplot(2,1,2);

rlocus(Gs)

case{7}

subplot(2,2,1);

step(Gs,t)

subplot(2,2,2);

rlocus(Gs)

subplot(2,2,3);

bode(Gs)

subplot(2,2,4);

nyquist(Gs)

end

case{2}

k=input('enter the gain k = ');

b0=input('enter your numeric b0 = ');

b1=input('enter your numeric b1 = ');

a0=input('enter your denumeric a0 = ');

a1=input('enter your denumeric a1 = ');

a2=input('enter your denumeric a2 = ');

numG=[b1 b0];

num=numG\*k;

denG=[a2 a1 a0];

Gs=tf(num,denG)

t=0:0.01:10;

o=menu('which one of diagrams would u like to present','Rlocus Diagram','Bode Diagram','Step Response','Nyquist Diagram','Nichols Diagram','step & Rlocus','all of Diagrams');

switch o

case{1}

rlocus(Gs)

case{2}

bode(Gs)

case{3}

step(Gs)

case{4}

nyquist(Gs)

case{5}

nichols(Gs)

case{6}

subplot(2,1,1);

step(Gs,t)

subplot(2,1,2);

rlocus(Gs)

case{7}

subplot(2,2,1);

step(Gs,t)

subplot(2,2,2);

rlocus(Gs)

subplot(2,2,3);

bode(Gs)

subplot(2,2,4);

nyquist(Gs)

end

*با عنایت به موارد فوق و تابع تبدیل سیستم مرتبه ی دوم و برنامه ی نوشته شده در فوق،پاسخ سیستم را به ازای مقادیر مختلف مشاهده می نماییم : با فرض*  *پاسخ پله ی سیستم و مکان ریشه های آن به شرح زیر خواهد بود :*

*>1*

*enter the gain k = 1*

*enter your numeric b0 = 1*

*enter your numeric b1 = 0*

*enter your denumeric a0 = 1*

*enter your denumeric a1 = 4*

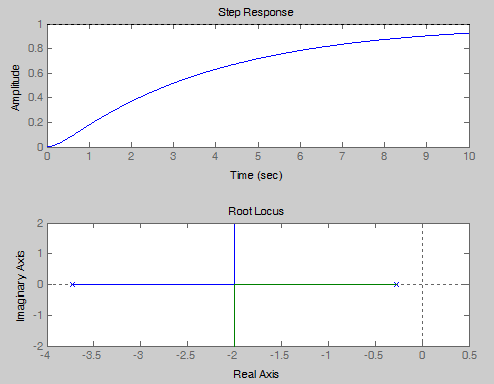
*enter your denumeric a2 = 1*

*Transfer function:*

*1*

*-------------*

*s^2 + 4 s + 1*

**

*=1*

enter the gain k = 1

enter your numeric b0 = 1

enter your numeric b1 = 0

enter your denumeric a0 = 1

enter your denumeric a1 = 2

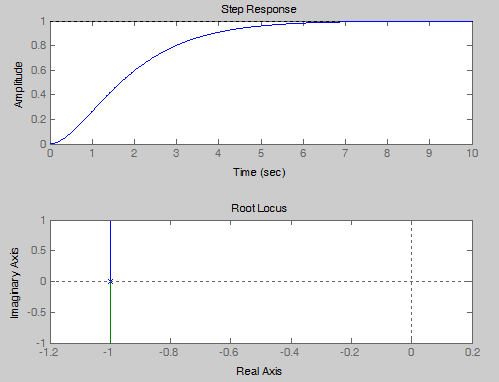
enter your denumeric a2 = 1

Transfer function:

1

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

s^2 + 2 s + 1



*<1 > 0*

enter the gain k = 1

enter your numeric b0 = 1

enter your numeric b1 = 0

enter your denumeric a0 = 1

enter your denumeric a1 = 1

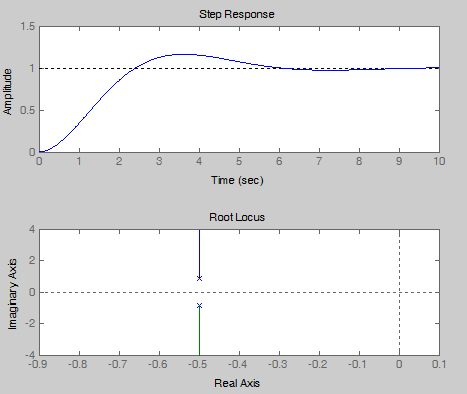
enter your denumeric a2 = 1

Transfer function:

1

-----------

s^2 + s + 1



enter the gain k = 1

enter your numeric b0 = 1

enter your numeric b1 = 0

enter your denumeric a0 = 1

enter your denumeric a1 = 0

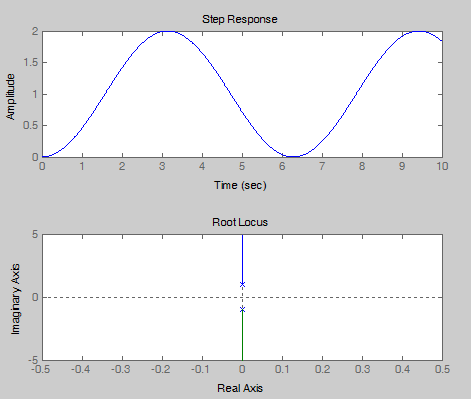
enter your denumeric a2 = 1

Transfer function:

1

-------

s^2 + 1



*< 0 > -1*

enter the gain k = 1

enter your numeric b0 = 1

enter your numeric b1 = 0

enter your denumeric a0 = 1

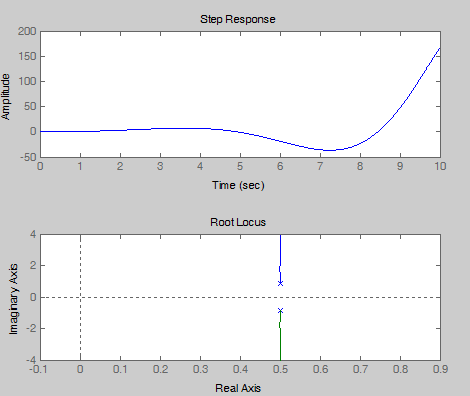
enter your denumeric a1 = -1enter your denumeric a2 = 1

Transfer function:

1

-----------

s^2 - s + 1



*-1*

enter the gain k = 1

enter your numeric b0 = 1

enter your numeric b1 = 0

enter your denumeric a0 = 1

enter your denumeric a1 = -2

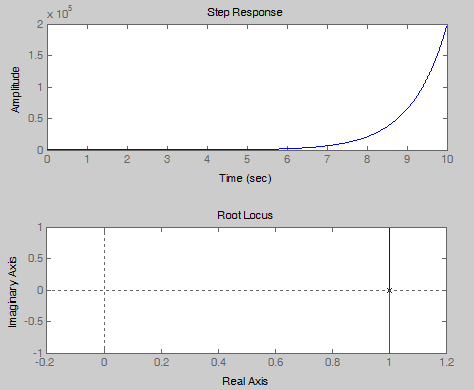
enter your denumeric a2 = 1

Transfer function:

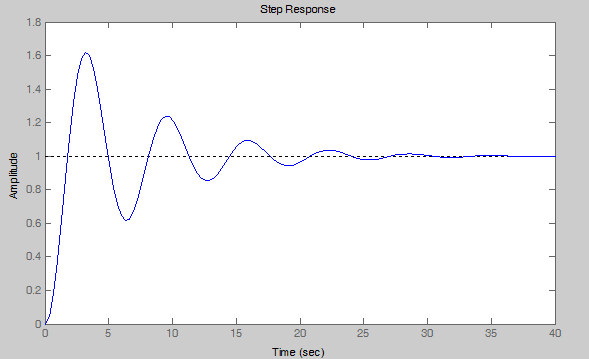
1

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

s^2 - 2 s + 1



پاسخ سیستم را به ازای ( ثابت) و تغییرات نسبت میرایی دیدیم. همواره سیستمی مد نظر ماست که میرای ضعیف باشد تا بتوان آن را به طور مناسب کنترل کرد. از این رو به بررسی دقیق تر پاسخ سیستم درجه دوم با میرایی ضعیف می پردازیم.



**زمان ماکزیمم جهش**

**زمان خیز یا صعود**

**زمان نشست**

**حداکثر فراجهش**

مشخصات مهم سیستم به ازای پاسخ پله همان طور که در شکل فوق نشان داده شده است به شرح زیر است :

1. ماکزیمم فرا جهش که با Mp نمایش داده می شود.
2. زمان ماکزیمم فرا جهش که با tp نمایش داده می شود.
3. زمان نشست یا استقرار که با ts نمایش داده می شود.
4. زمان خیز یا صعود که با tr نمایش داده می شود.

**ماکزیمم فراجهش :** اگر حداکثر مقدار خروجی سیستم را پس از اعمال ورودی پله با Cmax نمایش دهیم و Css نیز مقدار حالت ماندگار آن باشد. آن گاه :

حداکثر فراجهش = Cmax – Css

درصد حداکثر فراجهش = 100%

ماکزیمم فراجهش از رابطه ی Mp = به دست می آید. این مقدار Mp به نسبت میرایی بستگی دارد و با کم شدن آن زیاد می شود.

**زمان ماکزیمم فرا جهش tp :** زمان ماکزیمم فراجهش tp عبارتست از زمان لازم برای رسیدن به اولین ماکزیمم. پاسخ این زمان به فرکانس میرایی طبیعی  بستگی دارد و با کم شدن آن زیاد می شود.

Tp = =

**زمان نشست ts :** زمان استقرار یا نشست عبارتست از زمان لازم برای آن که پاسخ به محدوده ی معینی (معمولاً ) از مقادار نهایی برسد و از آن پس از این محدوده تجاوز نکند. زمان استقرار به بزرگترین ثابت زمانی سیستم مربوط است و با زیاد شدن آن زیاد می شود.

ثابت زمانی سیستم برابر T = = است. می توان نشان داد که ts از روابط زیر به دست می آید :

Ts = 4T = با معیار 2%

Ts = 3T = با معیار 5%

**زمان صعود یا خیز tr :** زمان صعود به زمانی گفته می شود که لازم است تا پاسخ از 10% به 90% یا از 0% به 100% مقدار نهایی خودش برسد. برای سیستم های میرایی ضعیف معمولاً از 0 تا 100 درصد و برای سیــــستم های میرای شدید اکثراً از 10 % تا 90 % استفاده می شود.

Tr =

با توجه به رابطه در می یابیم که اگر ثابت باشد، برای داشتن tr کوچک باید را بزرگ انتخاب کرد.

**اثر اضافه کردن صفر و قطب به توابع تبدیل**

در سیستم های مرتبه ی دوم به بالا، پیدا کردن آن دسته از قطب های سیستم که بیشترین اثر را بر پاسخ دارند اهمیت زیادی دارد. به این قطب ها، قطب های غالب سیستم می گویند. از آن جا که اکثر سیستم های کنترل از مرتبه ی بالا هستند، لذا پیدا نمودن و تدوین دستور العملی که این سیستم ها را توسط مدل های مرتبه پایین تر مانند مدل مرتبه دوم تقریب بزند بسیار سودمند است. در کلیه ی کاربردهای عملی نیز می توان صفحه ی S را به دو قسمت شامل قطب های غالب و قطب های کم اثر تقسیم نمود.

قاعده ی کلی آن است که قطب هایی که در صفحه ی s در سمت چپ محور موهومی قرار دارند و به آن نزدیک هستند پاسخ گذرای کندی خواهند داشت. حال آنکه قطب هایی که در سمت چپ محور موهومی s قرار دارند و نسبت به قطب های غالب بسیار دورتر از محور موهومی هستند پاسخ گذرای بسیار سریعی دارند و با سرعت بیشتری صفر خواهند شد. مطلب مهم تشخیص فاصله ی R بین این دو محدوده می باشد. از نظر عملی اگر اندازه ی قسمت حقیقی قطبی حداقل 5 تا 10 برابر قسمت حقیقی قطب غالب یا قطب های مختلف مزدوج غالب باشد، آن قطب کم اثر است.

**اثر افزودن قطب**

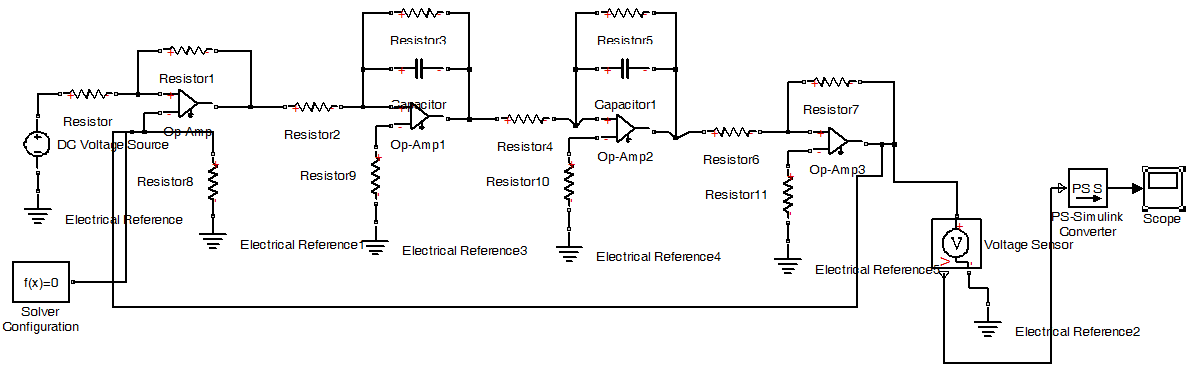
افزودن یک قطب به تابع تبدیل حلقه باز باعث می شود مکان هندسی ریشه ها به سمت راست کشیده شود و درجهت کم شدن پایداري نسبی سیستم و کند کردن نشست پاسخ عمل می کند.

**اثر افزودن صفر**

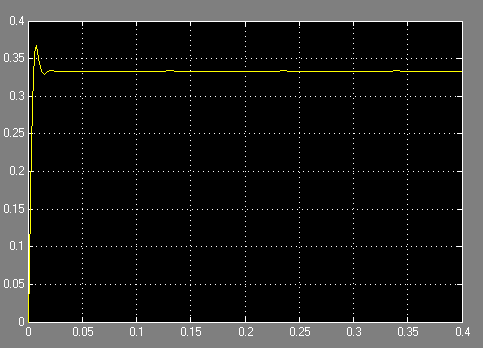
افزودن یک صفر به تابع تبدیل حلقه باز مکان هندسی ریشه ها را به سمت چپ می کــــشد، سیستم را پایدارتر و سرعت را بیشتر می کند. از لحاظ فیزیکی افزودن صفر به تابع تبدیل مسیر پیشخورد، معادل کاربرد کنترل مشتقی است .این کار یک حالت پیش بینی به سیستم می دهد و به پاسخ گذرا سرعت می بخشد. افزودن یک صفر به سیستم آن را از لحاظ تمام مقادیر بهره پایدار کرده است.

منظور از کنترل حلقه بسته استفاده از عمل فید بک براي کاهش خطاي سیستم است. یکی از مزایاي سیستم هاي کنترل فید بک دار این است که فید بک پاسخ سیستم را نسبت به اغتشاش خارجی و تغییر پارامتر هاي داخلی سیســتم تقریباً بی اثر می کند .بنابراین می توان با استفاده از اجزاء ارزان و نه چندان دقیق دستگاه را به خوبی کنترل کرد.

**بررسی مدارات مرتبه دوم عملی با Op-Amp**

به منظور بررسی و تجزیه و تحلیل مدارات مرتبه ی دوم با Op-Amp از محیط سیمولینک متلب بهره می گیریم. شکل زیر یک نمونه مدار مرتبه ی دوم را نمایش می دهد.

پاسخ سیستم فوق به ازای خازن های 3.3 میکروفاراد و مقاومت های 1 کیلو اهمی به صورت زیر است :

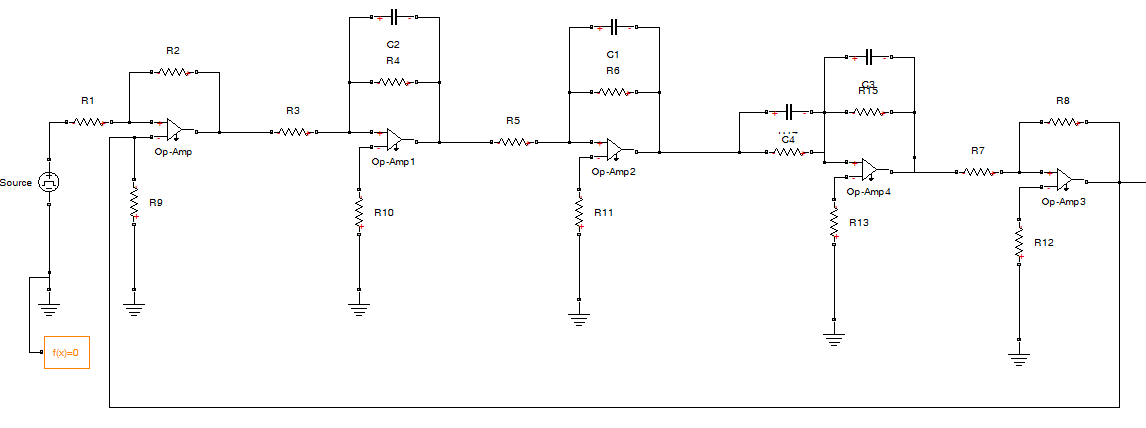


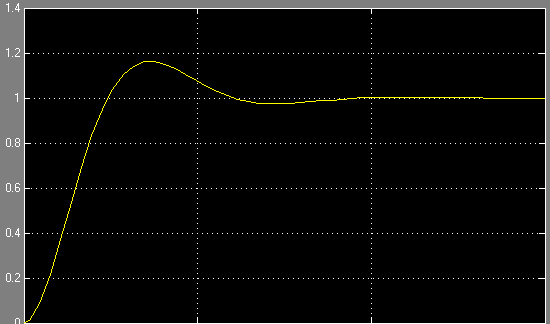
مشاهده می کنیم که خطای حالت ماندگار سیستم بسیار زیاد بوده و سیستم با توجه به نیاز ما عمل نمی کند. لذا می بایست این خطای نه چندان مناسب را به گونه ای صفر کنیم. بدین منظور از کنترل کننده ها و جبران ساز ها استفاده می نماییم.

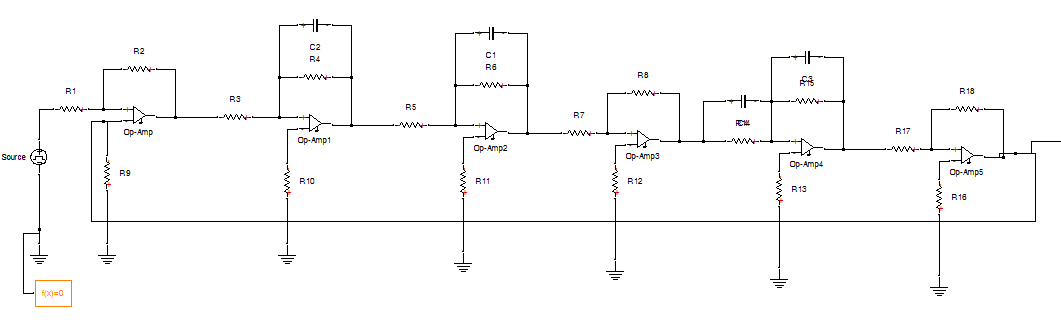
**کنترل کننده ها و جبران سازها**

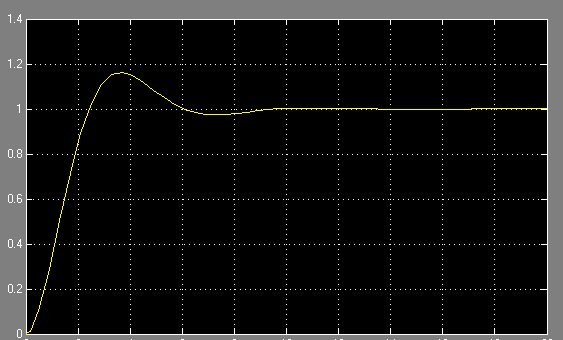
**جبران ساز پیش فاز :** تابع تبدیل این جبران ساز به صورت زیر می باشد :

=

با توجه به تابع تبدیل می بینیم که صفر در سمت راست قطب قرار دارد، زیرا > 0 است. یک سیستم مرتبه ی دوم به همراه جبران ساز پیش فاز در شکل زیر رسم شده است :

 پاسخ سیستم به ازای ورودی پله ی واحد نیز در زیر رسم شده است . با تغییر مقادیر مقاومت و خازن به شکل موج زیر دست یافته ایم. این پاسخ خروجی نیز به ازای مقادیر زیر حاصل شده است.

**جبران ساز پس فاز :** تابع تبدیل جبران ساز پس فاز شبیه به جبران ساز پیش فاز می باشد با این تفاوت که در جبران ساز پس فاز صفر در سمت چپ قطب قرار دارد. مدار عملی با استفاده از Op-Amp یک جبران ساز پس فاز که روی سیستم مرتبه دوم نمونه بسته شده است در شکل زیر قرار گرفته است که در ادامه پاسخ این مجموعه را به ازای ورودی پله ی واحد مشاهده خواهیم کرد.



**کنترل کننده ی PID :** کنترل کننده ی PID به تنهایی از سه ترم تناسبی، مشتقی و انتگرالی تشکیل شده است. در ترم تناسبی فرمان کنترلی متناسب با میزان خطا و با بهره ی K تقویت می شود. در ترم مشتقی فرمان کنترل متناسب با نرخ تغییرات خطا تغییر می کند. در ترم انتگرالی فرمان کنترل متناسب با جمع تابع خطا از زمان صفر تا به حال به صورت انتگرال این تابع تغییر می کند. با توجه به نیازی که ما از کنترل کننده داریم می توانیم از هر سه ترم کنترل کننده و یا تنها از یک ترم آن استفاده کنیم. تابع تبدیل این کنترل کننده در زیر آمده است :



با صفر کردن ضریب هر ترم از کنترل کننده ی PID می توان به کنترل کننده ی خاص دیگری دست یافت. به عنوان مثال با صفر کردن ضریب کنترلر مشتقی می توان به کنترل کننده ی تناسبی انتگرالی یا PI دست یافت. پس کنترلرهای P، D، I،PI،PD هر کدام حالت خاصی از کنترلر PID هستند که هم اکنون اثرات هر یک را بیان می کنیم.

هدف استفاده از کنترل کننده در یک فرایند به صورت زیر خلاصه می شود:

* پایداري داخلی
* دنباله روي از فرمان ورودي
* تضعیف اغتشاش
* کاهش نویز
* عدم حساسیت به تغییرات فرایند

1. کنترل کننده ی تناسبی P : اگر تنها ترم تناسبی کنترل کننده ی PID را در نظر بگیریم، آن گاه U(t) = Ke(t) حال یک سیستم کنترل فرآیند به فیدبک و حضور نویز و اغتشاشات داخلی سیستم را به صورت زیر در نظر بگیرید.

با افزایش بهره ی تناسبی K مقدار بهره حلقه افزایش می یابد لذا بهره تناسبی در جهت تنظیم خروجی بسیار موثر است اما از طرف دیگر اثر نویز به صورت 100% در خروجی فرایند مشاهده می شود و توانایی تضعیف تاثیر نویز بر روی خروجی وجود ندارد. از طرف دیگر با افزایش بهره کنترلی K پایداری سیستم حلقه بسته کاهش می یابد. به این دلایل معمولاً مقدار بهینه ای را بایستی برای بهره کنترل کننده تناسبی K تعیین نماییم تا به حاشیه ی پایداری دلخواه برسیم.

یک مثال از تاثیر کنترل کننده ی تناسبی K را روی یک فرایند در زیر مشاهده می کنید. که با افزایش بهره کنترلی، خطای حالت ماندگار سیستم کاهش یافته، اما پاسخ ناپایدارتر و نوسانی تر می شود.



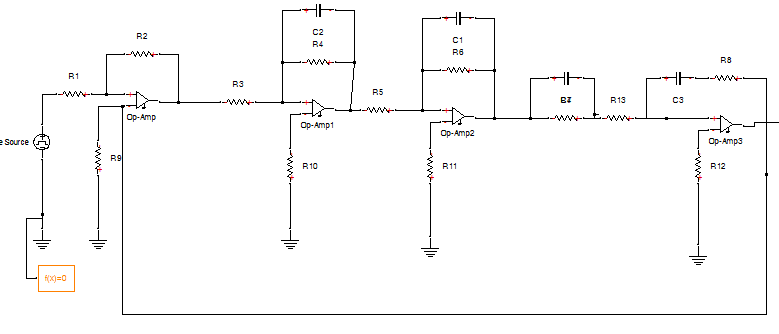
1. کنترل کننده ی انتگرال گیر I : تاثیر اصلی ترم انتگرال گیر در کنترلرهای PID کاهش خطای ماندگار است. به این معناست که پس از گذشت زمان گذرا، پاسخ سیستم به مقدار مطلوب نزدیک گردیده و خطای ماندگار کاهش می یابد. در مثال زیر نشان می دهد که در صورت عدم وجود کنترل کننده ی انتگرالی خطای ماندگار تا حدود 50% است اما با افزایش بهره ی انتگرالی این خطا به تدریج کاهش یافته و یا کاملاً از بین می رود. البته عیب این کنترل کننده این است که با کاهش خطا سرعت پاسخ سیستم را نیز کاهش می دهد. پس افزایش این ضریب موجب کند شدن سیستم می گردد.

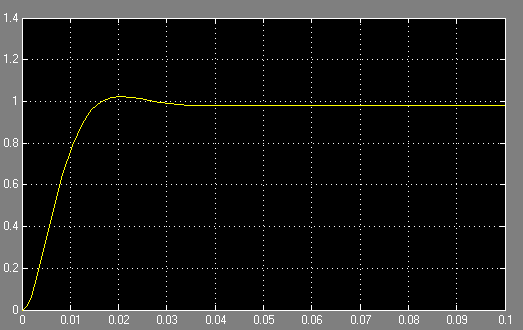


1. کنترل کننده ی مشتق گیر D : اضافه نمودن ترم مشتقی به کنترل کننده PID به دو منظور صورت می پذیرد : افزایش حاشیه ی پایداری در سیستم مدار بسته و افزایش سرعت پاسخ یا پهنای باند سیستم . در شکل زیر این موضوع بسیار آشکار است و مشاهده می شود که با افزایش بهره مشتقی سرعت پاسخ افزایش یافته ولی نوسانات سیستم افزایش یافته و پایداری نیز کاهش یافته است. پس نبایستی ضریب این کنترلر را خیلی افزایش دهیم.



1. کنترل کننده PI : این کنترلر برای حذف خطای حالت ماندگار به کار می رود، هر چند که ممکن است موجب ناپایداری سیستم شود. این کنترل کننده در واقع یک جبران ساز پس فاز است ، یک صفر در S= و یک قطب در S=0 دارد. پس مشخصه ی این کنترلر داشتن بهره ی بی نهایت در فرکانس صفر است که باعث بهبود رفتار حالت ماندگار سیستم می شود. استفاده از این کنترلر مرتبه سیستم را افزایش داده بنابراین پایداری کم می شود. با طراحی مناسب این کنترلر می توان کاری کرد که پاسخ فراجهشی کوچک داشته باشد و حتی بدون فراجهش باشد ولی سرعت پاسخ خیلی کم می شود.
2. کنترل کننده ی PD : این کنترلر برای افزایش زمان صعود و کاهش فراجهش به کار می رود. این کنترلر گونه ی ساده شده ی جبران ساز پیش فاز است و تابع تبدیلی با یک صفر و بدون قطب دارد و نمی توان با عناصر غیر فعال RLC آن را ساخت. کنترل کننده ی PD را می توان با اپ امپ و مقاومت و خازن ساخت ولی چون این کنترلر یک فیلتر بالاگذر است عمل مشتق گیری آن ممکن است در بعضی موارد با مشکلات جدی نویز همراه باشد. این کنترلر همانند کنترلر پیش فاز مشخصات پاسخ گذرا را بهبود می بخشد، پایداری را بهتر می کند و پهنای باند سیستم را زیاد می کند که به معنی زمان صعود کوتاه است.
3. کنترل کننده ی PID : این کنترل کننده تمام مشخصات مناسب کنترل کننده های فوق را دارد. و زمانی استفاده می شود که کاهش خطای حالت ماندگار را همراه با سرعت پاسخ زیاد سیستم بخواهیم. در واقع این کنترلر یک کنترلر پیشفاز-پس فاز است. توجه شود که کنترل PI در ناحیه فرکانس پایین و کنترل PD در ناحیه فرکانس بالا صورت می گیرند.

مدار Op-Amp کنترل کننده ی PID در شکل زیر آمده است که با توجه به مشخصات خوب کنترلر PID شاهدیم که پاسخ این سیستم به ازای ورودی پله ی واحد بسیار مناسب و خوب می باشد.



مقادیر مقاومت ها و خازن های سیستم و کنترلر PID فوق به شرح زیر است :

R1, R3,R4,R5,R6, R8,R9,R10,R11,R12 = 1KΩ

R2 = 100KΩ

R7 = 5KΩ

R13 = 200KΩ

C1,C2,C3,C4 = 3.3uF