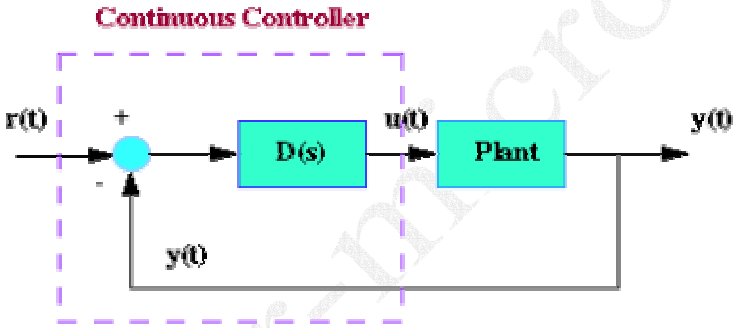
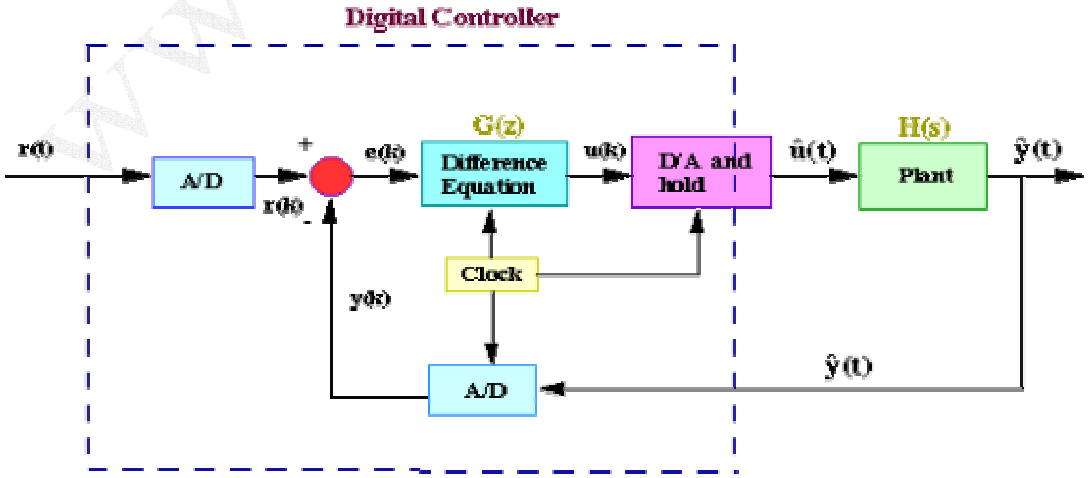
کنترل دیجیتال سرعت موتور با کنترلر PID

مقدمه

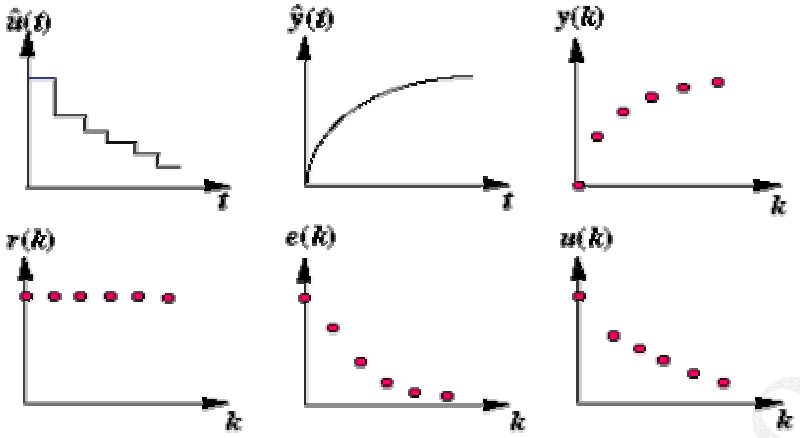
شکل زیر شمای یک سیستم حلقه بسته را نشان میدهد.تقریبا تمام کنترلرهای پیوسته باسیستمهای الکترونیکی قابل پیاده سازی هستند .



کنترلر های دیجیتال همانطور که در شکل نشان داده شده است میتوانند جایگزین سیستم های پیوسته شوند ووظایف کنترلی مشابهی رااجرا کنند . تنها تفاوت این سیستمها نوع سیگنال ویا نمونه برداری آنهاست .



انواع مختلف سیگنالهای مورد استفاده در شماتیک دیجیتال فوق در منحنی های زیر نشان داده شده است .

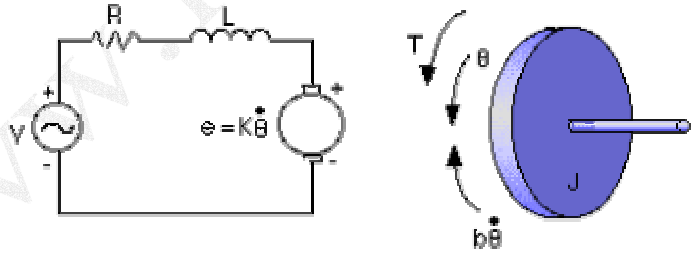


کنترل دیجیتال سرعت موتور با کنترلر PID

تبدیل پیوسته به گسسته

کنترلر PID

یکی از عملگرهای مرسوم در سیستم های کنترلی موتور جریان مستقیم است . که حرکتهای دورانی را برای ماایجاد میکند . مدار معادل الکتریکی ودیاگرام آزاد مدار به صورت زیر است :



مقادیر زیر رابرای پارامترهای فیزیکی در نظر بگیرید :

* moment of inertia of the rotor (J) = 0.01 kg.m^2/s^2
* damping ratio of the mechanical system (b) = 0.1 Nms
* electromotive force constant (K=Ke=Kt) = 0.01 Nm/Amp
* electric resistance (R) = 1 ohm
* electric inductance (L) = 0.5 H

* input (V): Source Voltage
* output (theta): position of shaft
* The rotor and shaft are assumed to be rigid

گشتاور موتورT با جریان آرمیچر I باضریب ثابت Kt متناسب است.Emf نیز با سرعت به صورت زیر رابطه دارد :

$$T = K_t i$$

$$e = K_t \dot{\theta}$$

براساس شکل نشان داده شده در بالا روابط زیر را برپایه قوانین نیوتن وکیرشهف می توان نوشت :

$$J\ddot{\theta} + b\dot{\theta} = Ki$$

$$L\frac{di}{dt} + Ri = V - K\dot{\theta}$$

تابع تبدیل

بااستفاده از تبدیل لاپلاس داریم :

$$s(Js + b)\Theta(s) = KI(s)$$

$$(Ls + R)I(s) = V - Ks\Theta(s)$$

باحذف I(s) ازدو رابطه تابع تبدیل حلقه باززیر را داریم :

$$\frac{\theta}{V} = \frac{K}{(Js + b)(Ls + R) + K^2}$$

پس از مدلسازی موتور DC تابع تبدیل حلقه باز زیر برای مشتق سرعت بدست خواهد آمد .

$$\frac{\dot{\theta}}{V} = \frac{K}{(Js+b)(Ls+R)+K^2}$$

که :

*electrical resistance (R) = 1 ohm
*electrical inductance (L) = 0.5 H
*electromotive force constant (Ke=Kt) = 0.01 Nm/Amp
*moment of inertia of the rotor (J) = 0.01 kg*m^2/s^2
*damping ratio of the mechanical system (b) = 0.1 Nms
*input (V): Source Voltage
*output (theta dot): Rotating speed
*The rotor and shaft are assumed to be rigid

در اینجا به کنترل دیجیتال سرعت موتور میپردازیم . میتوان یک مدل دیجیتال از موتور DC از تبدیل مدل آنالوگ بدست آورد . سپس در ادامه یک کنترلر PID برای آن طراحی خواهد شد .

نیازهای طراحی برای یک ورودی پله 1 rad/sec به صورت زیر می باشد:

- Settling time: Less than 2 seconds
- Overshoot: Less than 5%
- Steady-state error: Less than 1%

تبدیل پیوسته به گسسته

اولین قدم درطراحی سیستم کنترلی گسسته تبدیل تابع تبدیل پیوسته به گسسته می باشد.فرمان c2dm در نرم افزار Matlab این عمل را انجام می دهد. فرمان c2dm پارامترهای زیر نیاز دارد:

1-درجه چندجمله ایها(num), (den)

2-زمان نمونه برداری(Ts)

3-نوع مدار نگهدار

```
[numDz,denDz] = c2dm (num,den,Ts,'zoh')
[F,G,H,J] = c2dm (A,B,C,D,Ts,'zoh')
```

دراین مدار ما از مدار نگهدار ('zoh') استفاده میکنیم .

از نیازهای طراحی زمان نمونه برداری که 1/10 ثابت زمانی در نظر گرفته میشود.

حال M فایل مربوطه رامی نویسیم .

```
R=1;
L=0.5;
Kt=0.01;
J=0.01;
b=0.1;

num = Kt;
den = [(J*L) (J*R)+(L*b) (R*b)+(Kt^2)];

Ts = 0.12;
[numz,denz] = c2dm(num,den,Ts,'zoh')
```

با اجرای این برنامه خروجیهای زیر بدست می آید :

```
numz =

0    0.0092    0.0057

denz =

1.0000   -1.0877    0.2369
```

که میتوان به کمک این ماتریسها تابع تبدیل گسسته رابه صورت زیر نوشت.

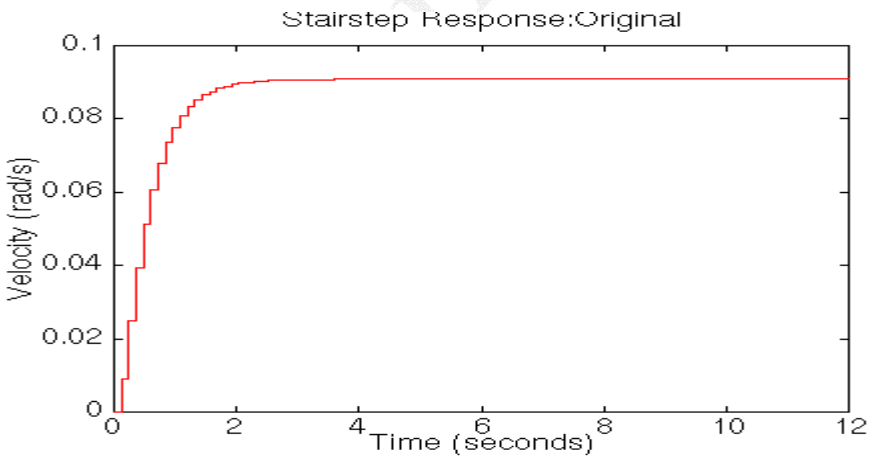
$$\frac{\theta(z)}{V(z)} = \frac{0.0092z + 0.0057}{z^2 - 1.0877z - 0.2369}$$

اگردقت کنید یک صفر اضفی در مقابل وجود دارد که باید آنرا قبل از بستن حلقه حذف کنیم . این عمل با فرمان cloop صورت میگیرد . باضافه کردن این فرمانها به برنامه این عمل صورت میپذیرد .

```
numz = [numz(2) numz(3)];
[numz_cl,denz_cl] = cloop(numz,denz);
```

حال که این کار صورت پذیرفت حال اجازه دهید ببینیم که شکل پاسخ پله حلقه بسته سیستم به چه شکلی در می آید .
فرمان dstep مقادیر سیگنالهای خروجی راتولید میکند وفرمان stairs سیگنالها را به هم مرتبط میکند . بااضافه کردن کدهای زیر به برنامه مطلب شکل زیر بدست خواهد آمد .

```
[x1] = dstep(numz_cl,denz_cl,101);
t=0:0.12:12;
stairs(t,x1)
xlabel('Time (seconds)')
ylabel('Velocity (rad/s)')
title('Stairstep Response:Original')
```



کنترلر PID

میدانیم که تابع تبدیل زمان پیوسته کنترلر PID به صورت زیر است :

$$K_P + \frac{K_I}{s} + K_D s$$

چندین روش برای mapping صفحه s-plane به صفحه z-plane وجود دارد . دقیق ترین روش استفاده از تبدیل $z = e^{Ts}$ است . ماما این روش نمیتوانیم به تابع تبدیل PID دست یابیم . چون تعداد صفرهای تابع تبدیل زمان گسسته ازصفرها بیشتر خواهد بود که غیرممکن است . به جای آن از روش bilinear transformation استفاده میکنیم که در زیر نشان داده شده است :

$$s = \frac{2}{Ts} \cdot \frac{z-1}{z+1}$$

به این ترتیب میتوان به PID گسسته دست یافت .

$$K_P + K_I \cdot \frac{Ts}{2} \cdot \frac{z+1}{z-1} + K_D \cdot \frac{2}{Ts} \cdot \frac{z-1}{z+1}$$

$$\frac{\left(K_P + \frac{K_I Ts}{2} + \frac{2K_D}{Ts}\right)z^2 + \left(K_I Ts - \frac{4K_D}{Ts}\right)z + \left(-K_P + \frac{K_I Ts}{2} + \frac{2K_D}{Ts}\right)}{z^2 - 1}$$

فرمان c2dm کمک میکند تا جبران ساز PID پیوسته را به گسسته تبدیل کنیم .

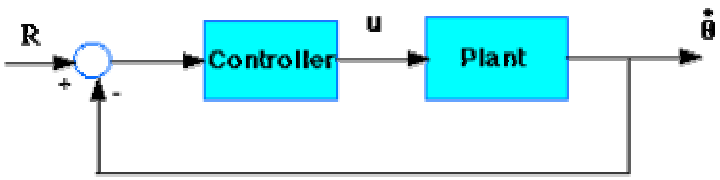
مطابق صورت مساله اصلی معادلات به صورت زیر است .

$$s(Js + b)\Theta(s) = KI(s)$$

$$(Ls + R)I(s) = V - Ks\Theta(s)$$

$$\frac{\theta}{V} = \frac{K}{(Js + b)(Ls + R) + K^2}$$

وشماتیک سیستم به صورت زیر است:



برای یک ورودی پله 1 rad/sec نقاط حساس سیستم به صورت زیر است:

- Settling time less than 2 seconds
- Overshoot less than 5%
- Steady-stage error less than 1%

حال اجازه دهید کنترلر PID راطراحی کنیم . فرامین زیر را برای مدلسازی سیستم در مطلب مینویسیم .

```
J=0.01;
b=0.1;
K=0.01;
R=1;
L=0.5;
num=K;
den=[(J*L) ((J*R)+(L*b)) ((b*R)+K^2)];
```

تابع تبدیل کنترلر PID رانیز داریم .

$$K_p + \frac{K_I}{s} + K_D s = \frac{K_D s^2 + K_P s + K_I}{s}$$

کنترل تناسبی

ابتدا بایک کنترل تناسبی با گین 100 سعی میکنیم کدهای زیر را به برنامه اضافه میکنیم .

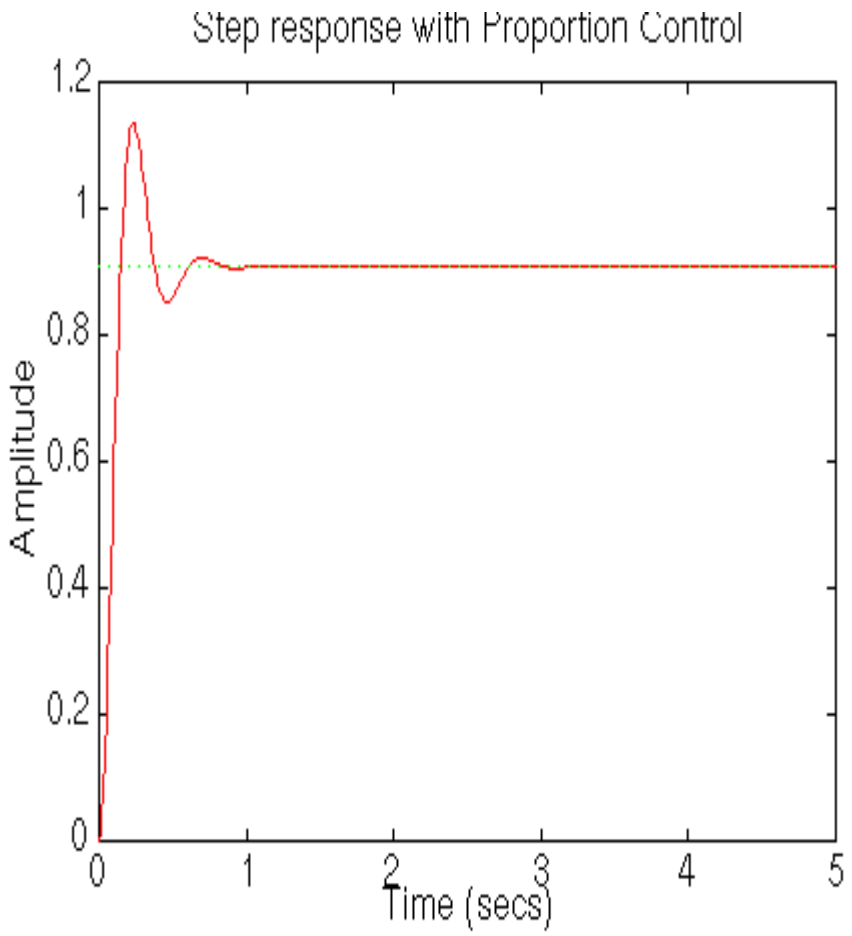
```
Kp=100;
numa=Kp*num;
dena=den;
```

برای تعیین تابع تبدیل حلقه بسته از فرمان cloop استفاده میکنیم.

```
[numac,denac]=cloop(numa,dena);
```

حالا اجازه دهید شکل پاسخ پله رابا اضافه کردن کدهای زیر بررسی کنیم.

```
t=0:0.01:5;
step(numac,denac,t)
title('Step response with Proportion Control')
```

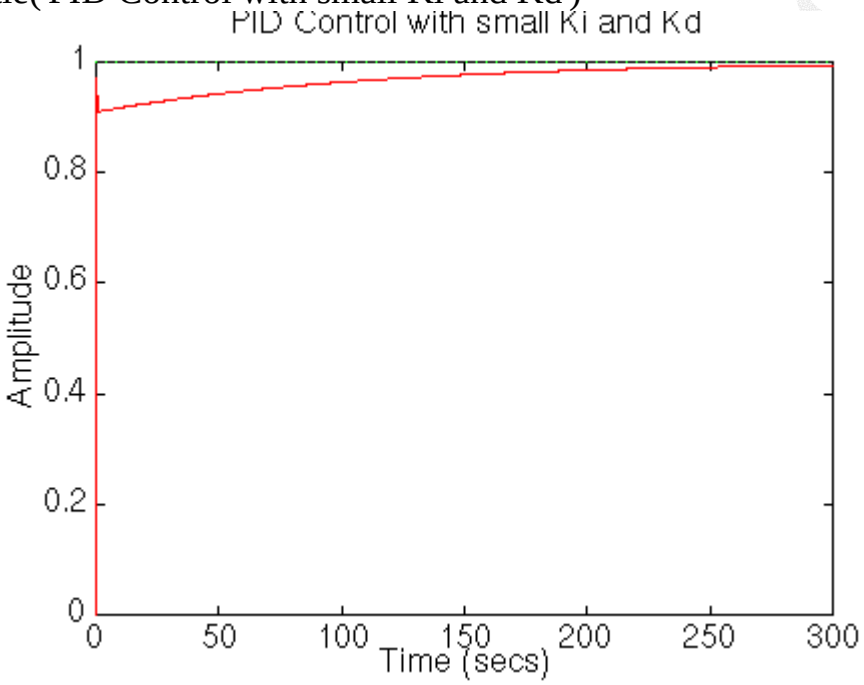


کنترلر PID

درشکل بالا میبینیم که خطای ماندگار وovershoot بسیارزیاد است . برای کاهش خطای ماندگار از یک انتگرالگیر وبرای کاهش overshoot ازیک مشتق گیر استفاده مینماییم . برای آغاز از ضرایب کوچک استفاده مینماییم . کدهای زیر را نوشته وپاسخ را مشاهده میکنیم .

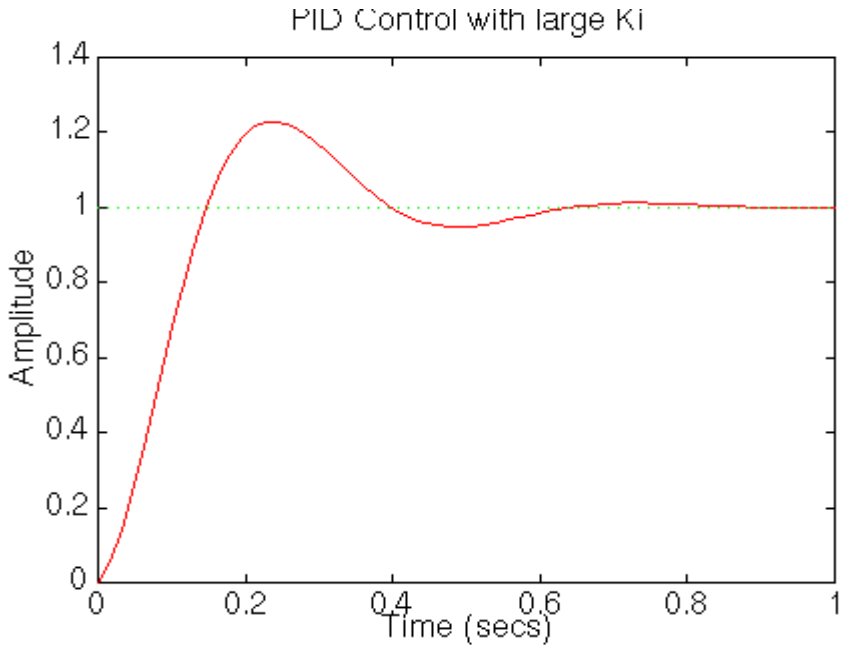
```
J=0.01;
b=0.1;
K=0.01;
R=1;
L=0.5;
```

```
num=K;
den=[(J*L) ((J*R)+(L*b)) ((b*R)+K^2)];

Kp=100;
Ki=1;
Kd=1;
numc=[Kd, Kp, Ki];
denc=[1 0];
numa=conv(num,numc);
dena=conv(den,denc);
[numac,denac]=cloop(numa,dena);
step(numac,denac)
title('PID Control with small Ki and Kd')
```

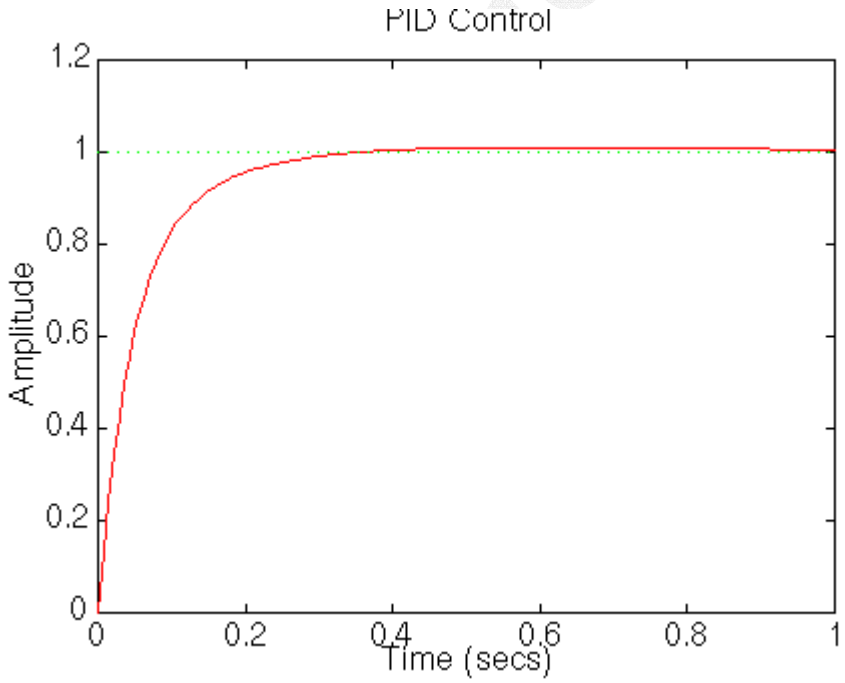


تنظیم ضرایب

حال میبینیم که settling time بسیار افزایش می یابد Ki را افزایش میدهیم تا settling timeکاهش یابد . Ki رابه 200 تغییر میدهیم . این فایل منحنی زیر رابر میگرداند .



حالا میبینیم که پاسخ بسیار سریع شده است ولی overshoot زیاد شده است . حال Kd رابرای کاهش overshoot زیاد میکنیم . Kd را 10میکنیم خروجی به صورت زیر خواهد شد :



خوب حالا ما میدانیم که اگر یک PID کنترلر با ویژگیهای زیر داشته باشیم همه نیازهای طراحی ارضا خواهد گردید .

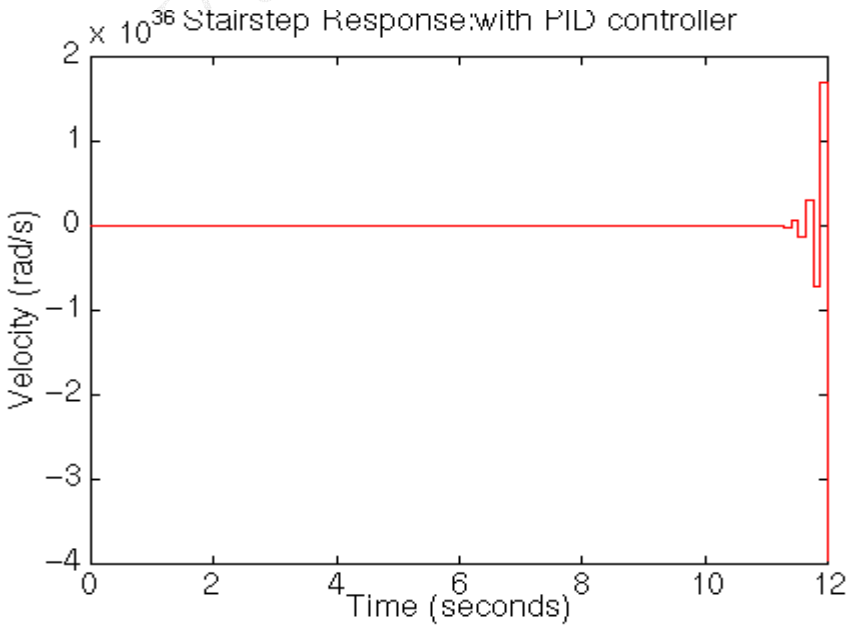
Kp=100,
Ki=200,
Kd=10,

حالا خطوط زیر را به برنامه اضافه میکنیم.

```
% Discrete PID controller with bilinear approximation

Kp = 100;
Ki = 200;
Kd = 10;

[dencz,numcz]=c2dm([1 0],[Kd Kp Ki],Ts,'tustin');
```

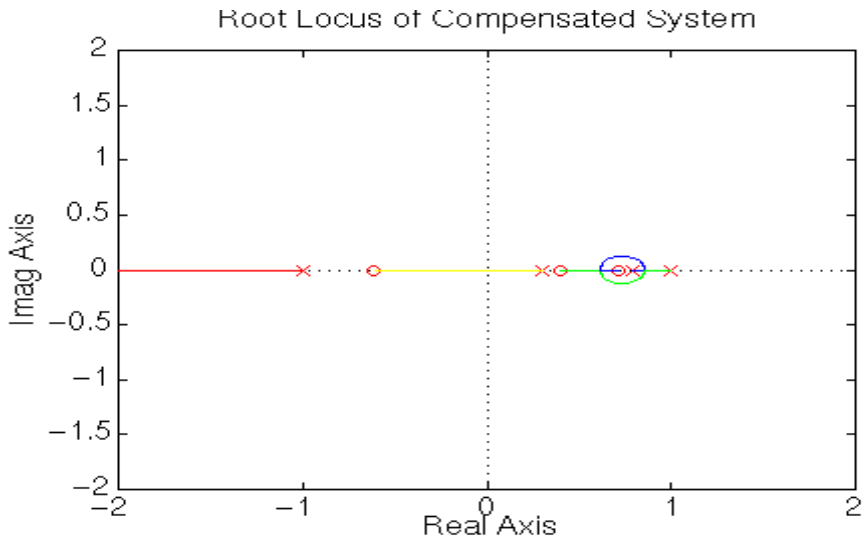
حال اجازه دهید پاسخ حلقه بسته سیستم رابا جبران سازها مشاهده کنیم . با اضافه کردن کدهای زیر پاسخ در خروجی نمایش داده خواهد شد .

```
numaz = conv(numz,numcz);
denaz = conv(denz,dencz);
[numaz_cl,denaz_cl] = cloop(numaz,denaz);

[x2] = dstep(numaz_cl,denaz_cl,101);
t=0:0.12:12;
stairs(t,x2)
xlabel('Time (seconds)')
ylabel('Velocity (rad/s)')
title('Stairstep Response:with PID controller')
```



همانطور که در شکل دیده میشود پاسخ حلقه بسته ناپایدار میشود. پس یک چیزی در جبران ساز اشتباه شده است.پس باید به پاسخroot locus سیستم نگاهی بیاندازیم.با اضافه کردن فرمانهای زیر به برنامه پاسخ root locus بدست می آید.

```
rlocus(numaz,denaz)
title('Root Locus of Compensated System')
```

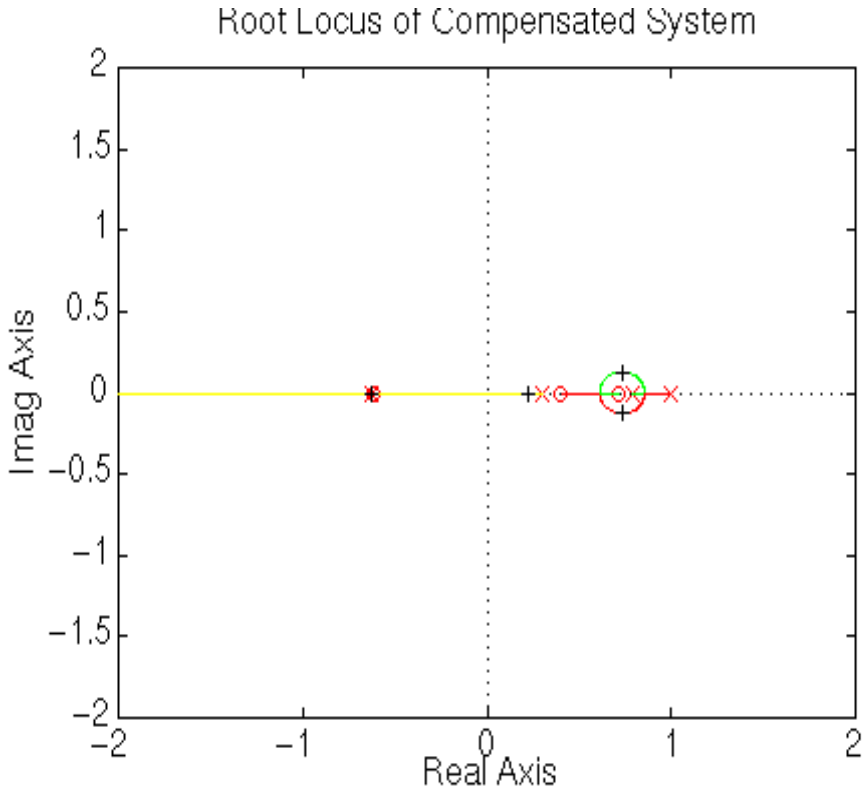


میبینیم که تابع تبدیل PID کنترلر یک قطب در 1- در صفحه z-plane دارد . میدانیم که اگر قطب در خارج از دایره واحد باشد سیستم ناپایدار است پس این سیستم جبرانسازی ناپایدار است. مایک صفر در 0.62- قرار می دهیم که این موجب میگردد سیستم برای حداقل برخی گین ها پایدار شود.علاوه براین ما یک گین مناسب برای ارضای نیازهای طراحی با استفاده ازrlocfind پیدا میکنیم . کدهای زیر را برای اجرای این برنامه به سیستم اضافه میکنیم :

```
dencz = conv([1 -1],[1.6 1])
numaz = conv(numz,numcz);
denaz = conv(denz,dencz);

rlocus(numaz,denaz)
title('Root Locus of Compensated System');
[K,poles] = rlocfind(numaz,denaz)
[numaz_cl,denaz_cl] = cloop(K*numaz,denaz);

[x3] = dstep(numaz_cl,denaz_cl,101);
t=0:0.12:12;
stairs(t,x3)
```

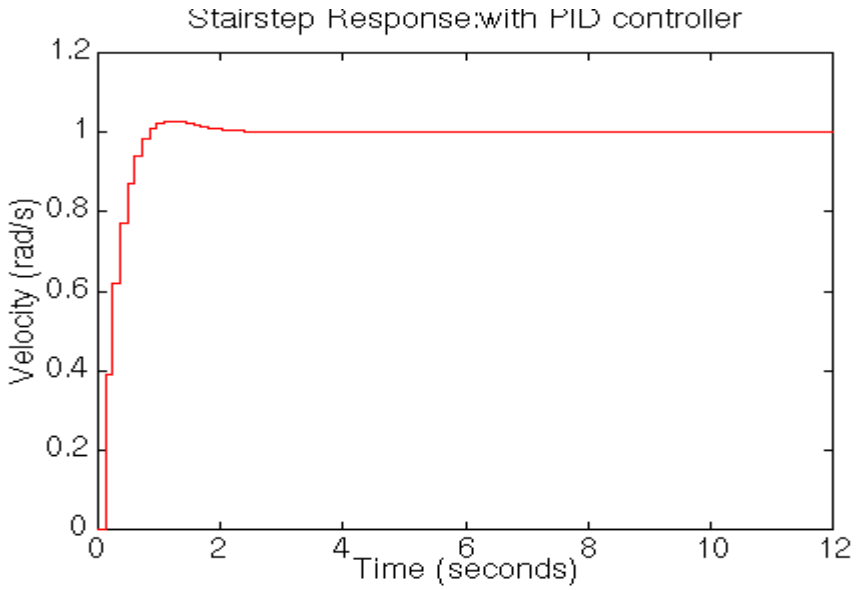
```
xlabel('Time (seconds)')
ylabel('Velocity (rad/s)')
title('Stairstep Response:with PID controller')
```

مایک قطب در 625.- قرار میدهیم که تقریبا اثر صفرهای جبران نشده سیستم را جبران میکند.در پنجره مطلب شما باید فرمان انتخاب نقاط در منحنیroot-locus رابببینید.باید برروی آن مطابق شکل زیر کلیک کنید.



سپس نرم افزار مطلب گین مناسب وقطب جبرانسازی شده مناسب رابه شما برمیگرداند .
وپاسخ حلقه بسته سیستم جبرانسازی شده را مطابق شکل زیر نمایش میدهد .

نتیجه :



منحنی نشان میدهد که settling time کمتر از 2 ثانیه وovershoot حدود 3٪ است وعلاوه براین خطای ماندگار صفر است . همچنین بهره مطابق منحنی 2425. است که منطقی بوده وبنابراین پاسخ نیازهای طراحی را ارضا میکند .

تهیه کننده: امیر حسین اکبری

گروه برقیران

Barghiran.Persianblog.com

با تشکر ار همکاری آقای اسماعیل زاده ها
