Design Project 2

Course Number: ECE 5330/EVE 5430/AET 5330

Title: Modeling and Control of Power Electronics and Electric Vehicle Powertrains

Instructor: Le Yi Wang, Professor

Student Name: Zhenyu Qu

Student ID: 004298218

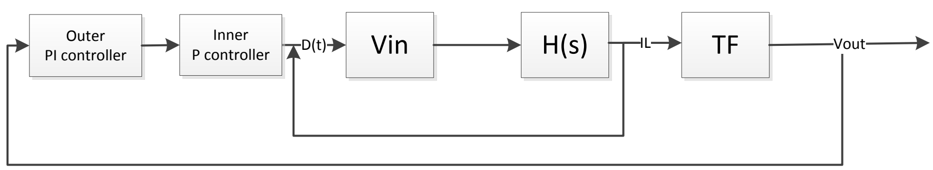
I have not received, nor have I given, any help and assistance on this project.

Signature: Date:

1.Modeling

Transfer function from d to inductor current：

View the whole system as a plant with d as input and I_L as out put



Total impedance

$$Z(s) = LS + R_t + R_L + \frac{R_L(R_C + \frac{1}{Cs})}{R_L + R_C + \frac{1}{CS}}$$

$$P(S) = \frac{1}{Z(s)} = \frac{(R_L + R_C)Cs + 1}{(R_t + Ls + R_L)[(R_L + R_C)Cs + 1] + R_L(R_C Cs + 1)} = \frac{0.02424s + 1}{2.6664 \times 10^{-6} s^2 + 1.3124 \times 10^{-3} + 4.01}$$

Transfer function is:

$$H(S) = \frac{I_L}{d} = V_{in} \times P(s) = \frac{1.0908s + 45}{2.6664 \times 10^{-6} s^2 + 1.3124 \times 10^{-3} + 4.11}$$

Relationship between inductor current I and Load voltage V_L is:

$$V_L = I_i \bullet \frac{(R_C + \frac{1}{Cs}) \bullet R_L}{(R_C + \frac{1}{Cs}) + R_L}$$

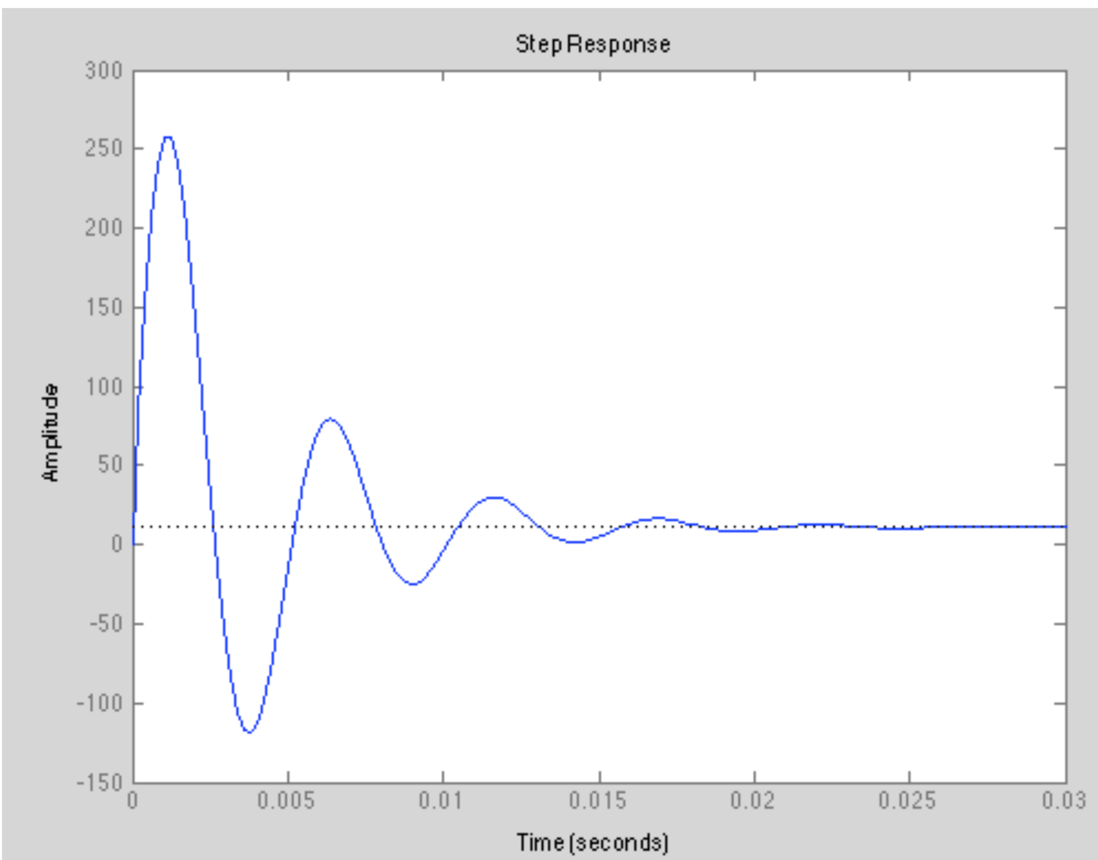
Transfer function From Inductor current to Load Voltage is :

$$\frac{V_L}{I_i}=\frac{(R_C+\frac{1}{Cs})\bullet R_L}{(R_C+\frac{1}{Cs})+R_L}=\frac{9.6\times10^{-4}s+4}{2.424\times10^{-2}s+1}$$

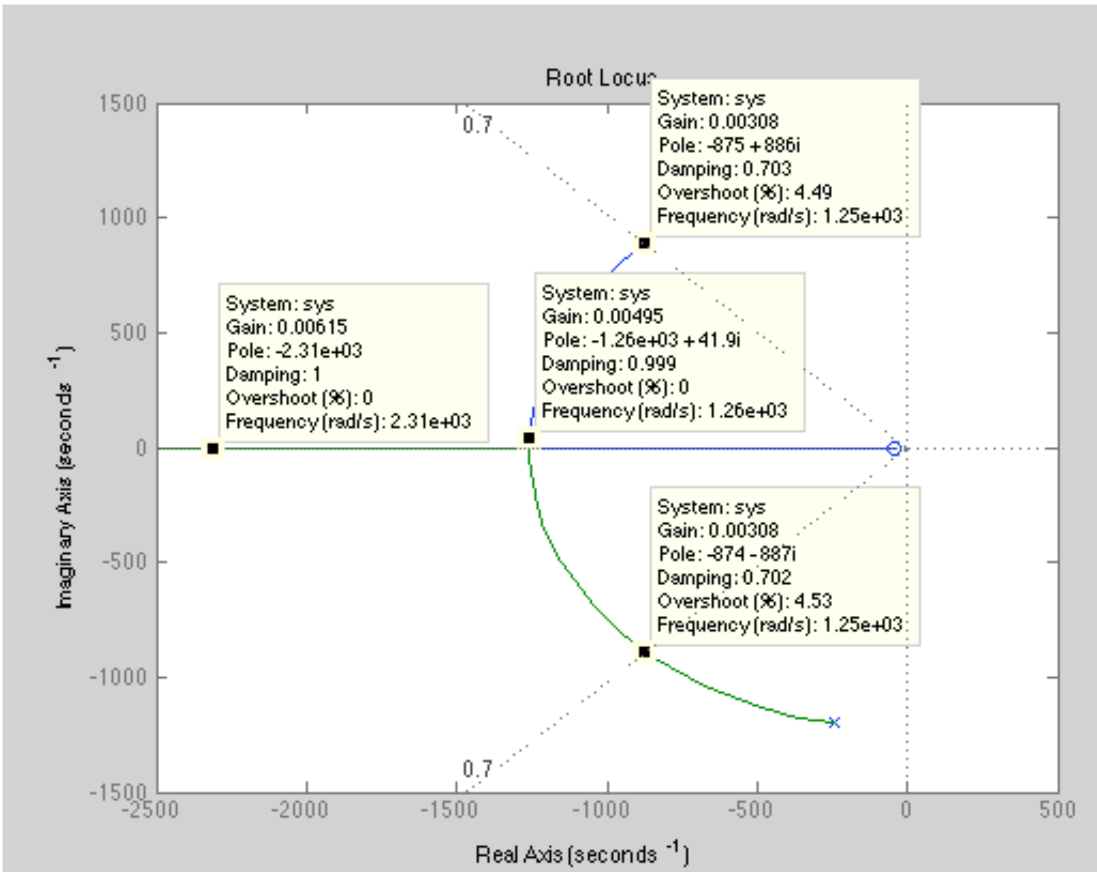
2.Time Domain Control Design

2（1）：Inner-loop Design

```
Step Response:
ng=[1.0908 45]
dg=[2.6664*10^(-6) 1.3124*10^(-3) 4.01]
sys=tf(ng,dg);
step(sys)
```



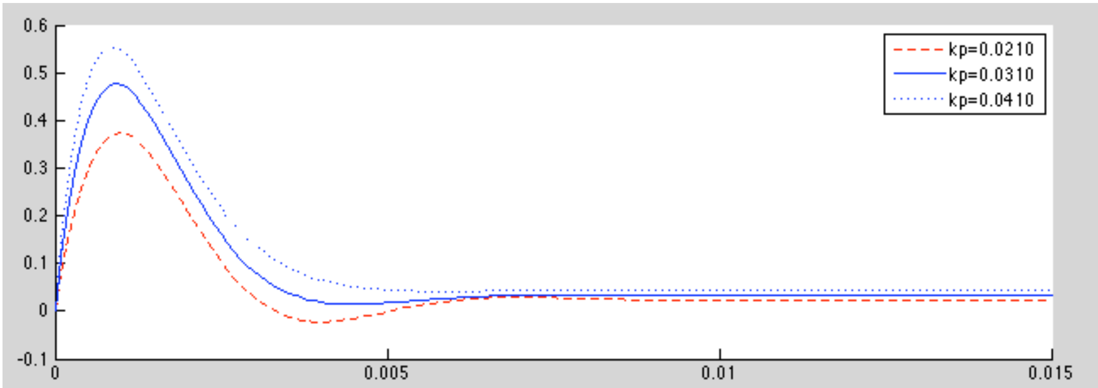
```
rlcous(sys);
sgrid(0.7,[])
```



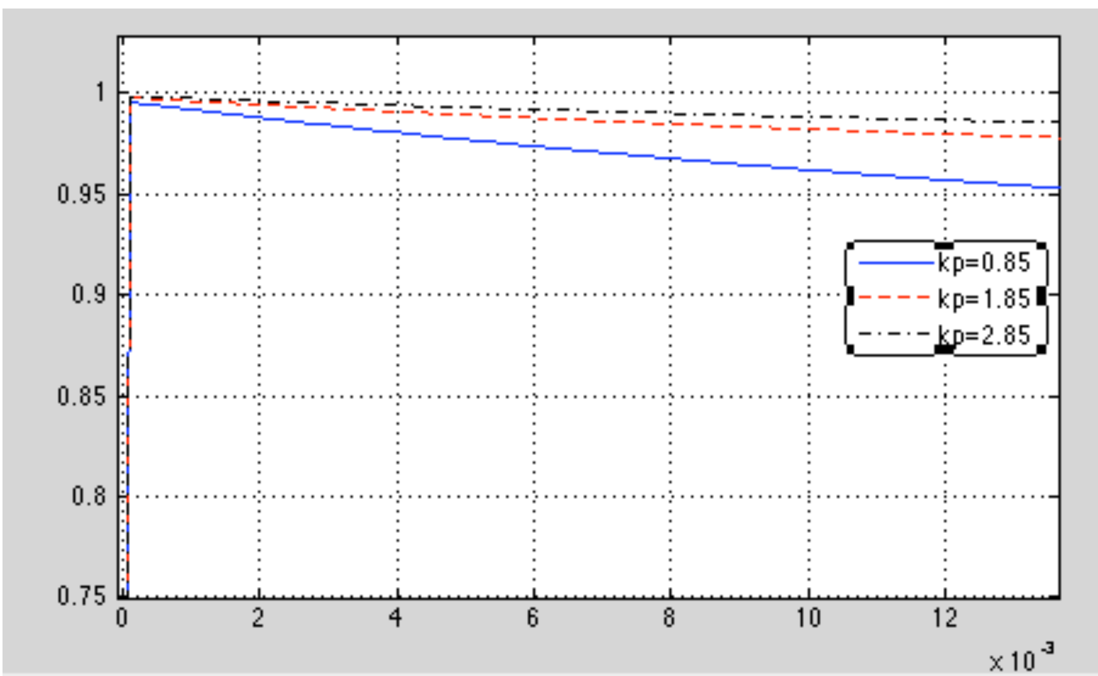
From root locus we could see when Gain is bigger than 0.00308, we can get damping ratio bigger than 0.7. Now use this gain as the P controller value Kp and verify that by doing step response.

```
t=0:0.0001:0.015;
for kp=0.00210:0.001:0.00410;
hold on
clsys=feedback(kp*sys,1);
y=step(clsys,t);
yss=y(length(t));
```

```
plot(t,y)
end
```



Then we can see the system does not follow the root locus, the overshoot is almost 100% while rootlocus shows overshoot is only4.5%, It is because Matlab is using approach to real time systems, it make mistakes during calculating. When kp=0.00308 overshoot is bigger than 100% and ess=0.9 therefore to shrink overshoot and steady state error, we need have a bigger kp.



When kp is bigger, over shoot is much more smaller, even disappear, compare with previous we can also find that system response much faster, raising time is smaller than

0.0001s. Also the kp is smaller with bigger P, this follows $e_{ss} = \frac{1}{1+k_p}$, when kp is bigger steady state error is smaller(closer to 1 from plot).

2（2）：Outer-loop Design

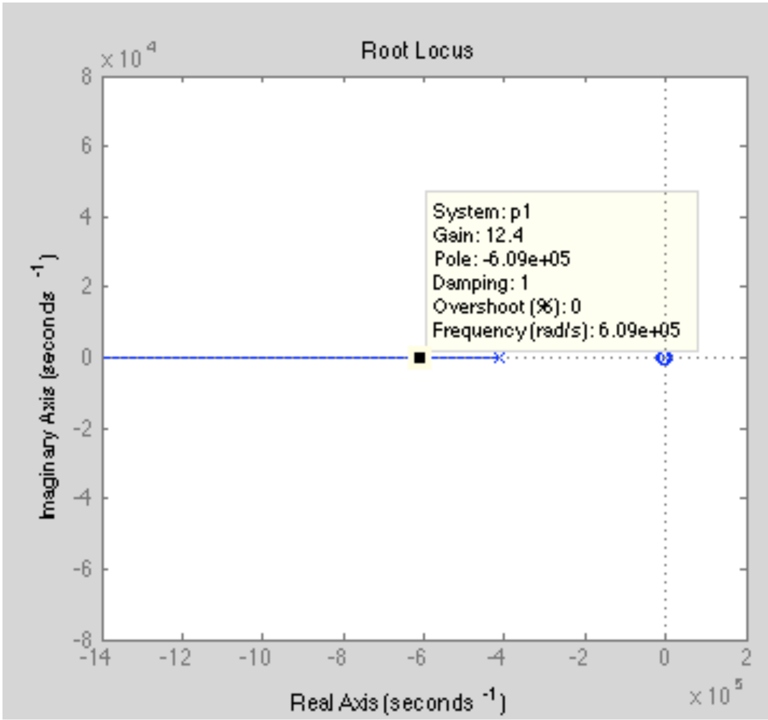
First multiply close-loop system with transfer function from I_L to V_O:

```
ng=[1.0908 45]
dg=[2.6664*10^(-6) 1.3124*10^(-3) 4.01]
sys=tf(ng,dg);
kp=2.85;
clsys=feedback(kp*sys,1);
n=[9.6*10^(-4) 4]
g=[2.424*10^(-2) 1]
vl=tf(n,g)
p1=clsys*vl
```

Then we have outer loop plant

$$P1 = \frac{0.001047s^2 + 4.046s + 180}{6.463 \times 10^{-8} s^3 + 0.02653s^2 + 2.285s + 49.11}$$

Again doing root locus to the new plant.



According to this bode plot the new system is so stable that it never thr damping ratio will always bigger than 1.

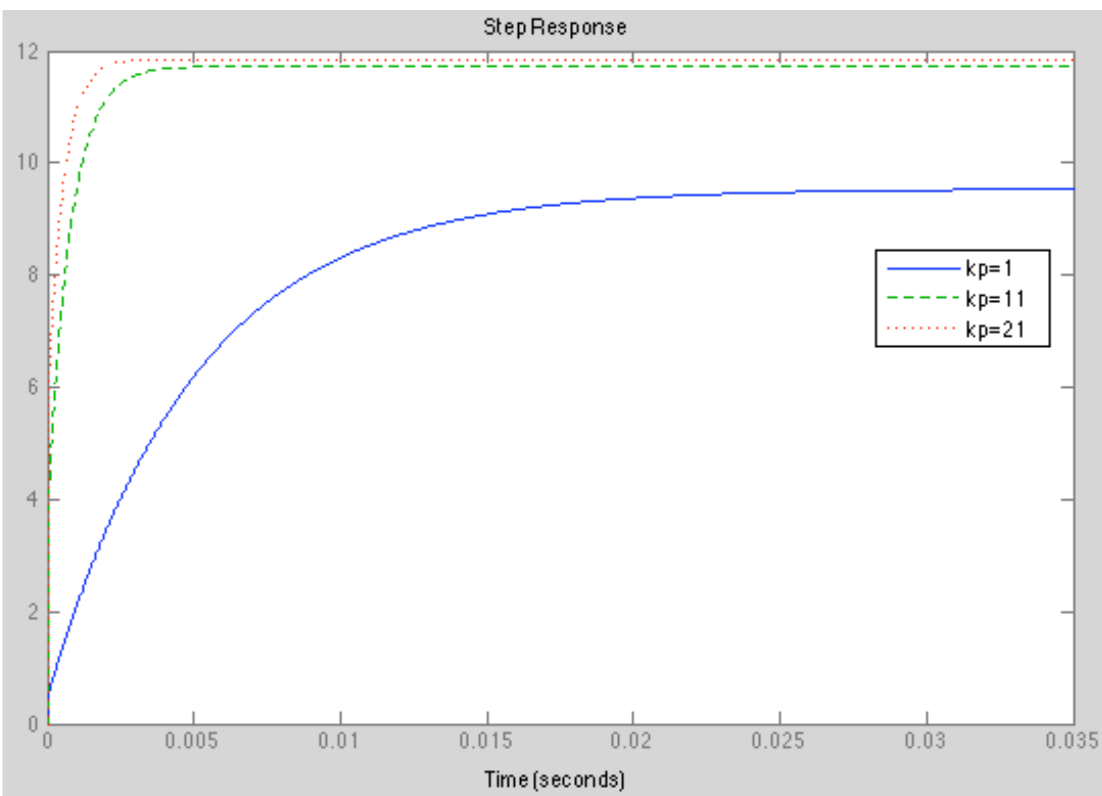
Then we apply PI controller to this new plant.

$$G_c(s) = K_P + \frac{K_I}{s} = K_P\left(1 + \frac{K_I}{K_P s}\right) = K_P\left(\frac{s+\bar{K}_I}{s}\right)$$

```
kp=2.85;% kp for inner loop
    hold on
    clsys=feedback(kp*sys,1);
    y=step(clsys,t);
    yss=y(length(t));
    figure(3)
    plot(t,y)
    grid on
n=[9.6*10^(-4) 4];
g=[2.424*10^(-2) 1];
vl=tf(n,g);
```

```
p1=clsys*vl;    %from inductor current to load voltage
figure(4)
step(p1)
figure(5)
rlocus(p1);
for p=1:10:30
    hold on
    nc=[p 0];%kp=p, Ki=0
    gc=[1 0];
    pi=tf(nc,gc);   % pi controller
    plant2=feedback(pi*p1,1) % close loop tf
    figure(8);
    step(12*plant2);
end
```

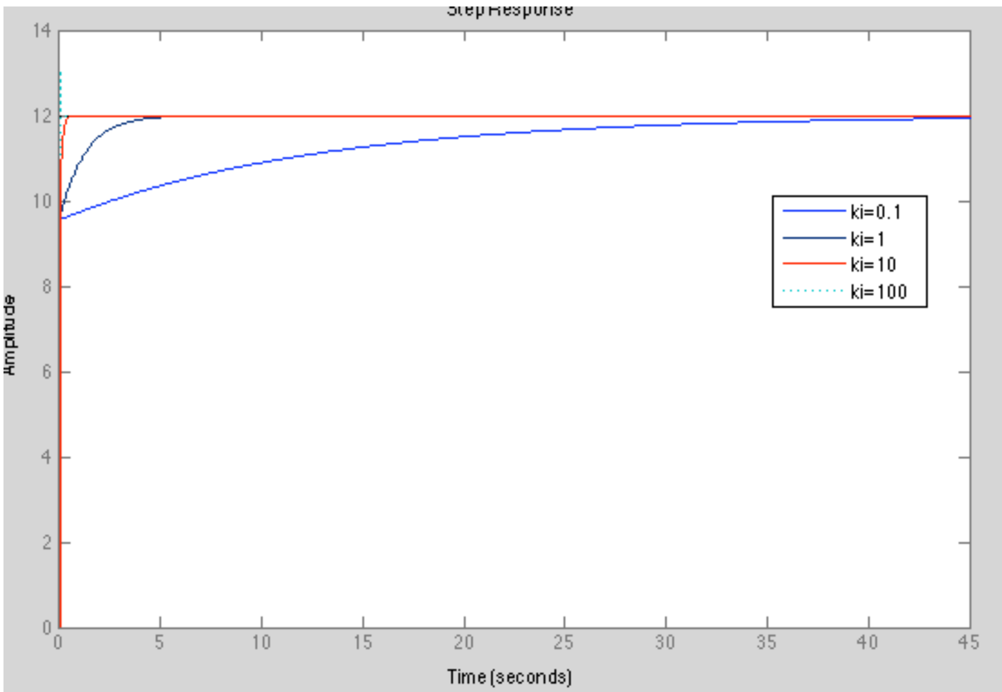
First let’s consider the situation that we only have a P controller here.



From this plot we can see when kp is bigger system response much faster and also steady-state error is smaller, but there always have steady-state error.

Conclusion: kp↑-> t_r ↓-> ess↓

Now apply a PI controller with ki=[0.1 1 10 100]



there we can see ki will eliminate steady-state error but when ki is too big it cause oscillation which is not good.

Ki-> t_r->t_p when we have Ki->ess=0

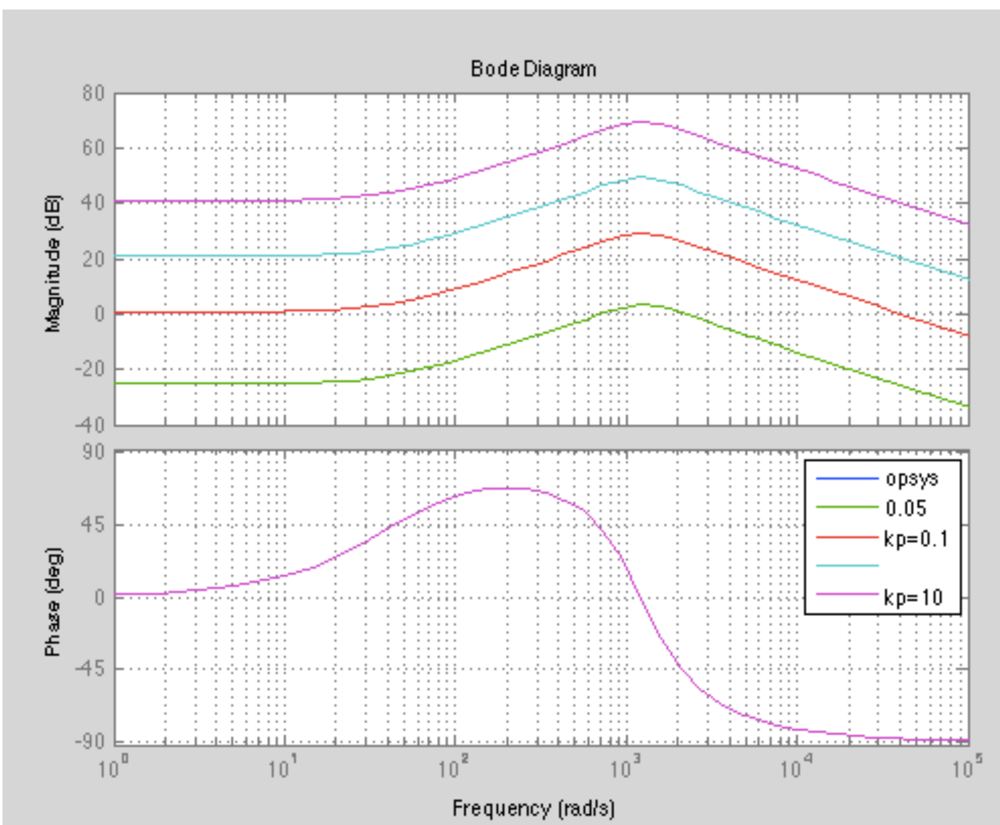
3. Frequency Domain Control Design

3(1): Inner-loop Design

For inner loop, build the plant multiple with a P controller and do the open loop bode plot, then do step response for close loop.

```
ng=[1.0908 45]
```

```
dg=[2.6664*10^(-6) 3.7364*10^(-3) 4.11]
sys=tf(ng,dg);
t=0:0.0001:0.03;
for kp=[0.005 0.1 1 10];% kp for inner loop
    hold on
    clsys=feedback(kp*sys,1) % close loop system transfer function
    opsys=feedback(kp*sys,0) % open loop system transfer function
    figure(3)
    bode(opsys);
    grid on
    figure(1)
    step(clsys)
end
```

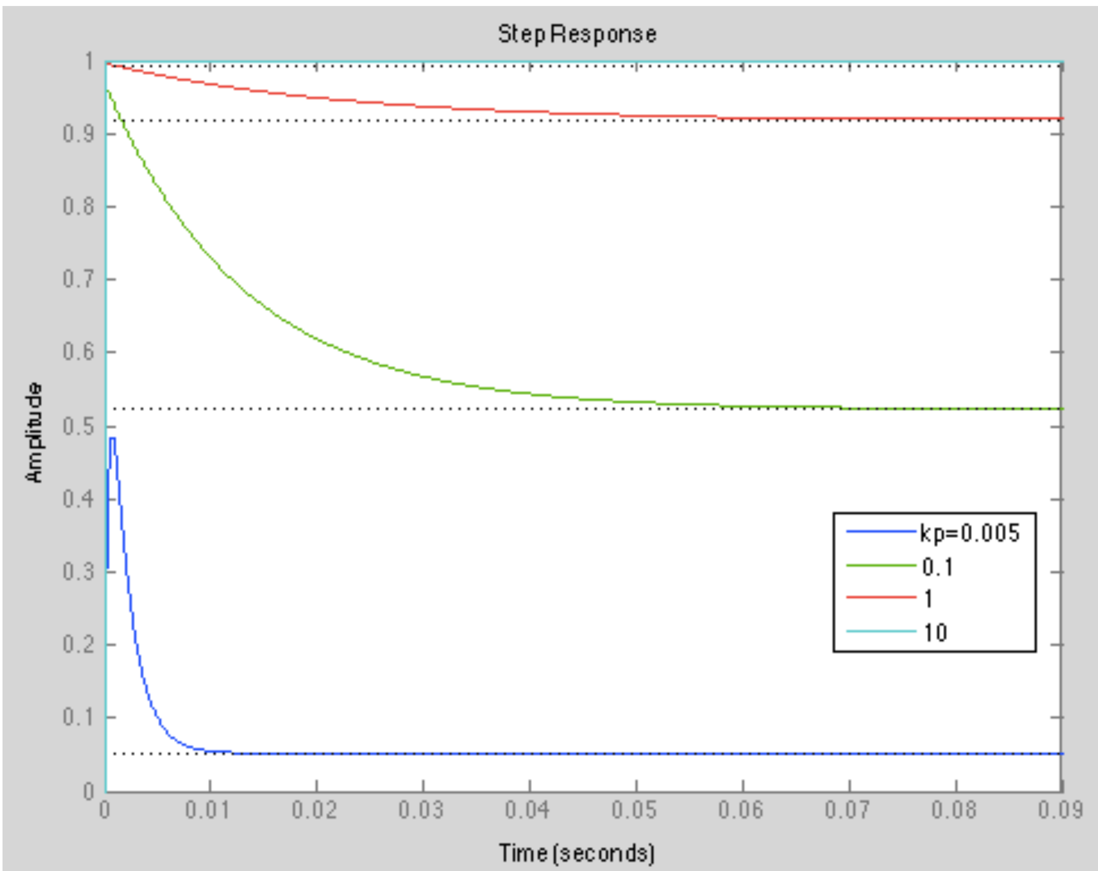


when kp=[0.005 0.1 1 10]， from the figure above we can see kp controller will shift magnitude curve up and down but it take no affect on phase curve.

More specifically when kp is bigger than 1 magnitude curve will shift up, when 0<kp<1 magnitude curve shit down.

For this system phase curve never cross -180° so system have unlimited gain margin. For kp=0.05 phase margin = 225°. Other curve does not cross gain=0.

Then we do step response to the system with different kp to see which kp is suitable for the goal of inner loop has a damping ratio larger than 0.7.



We already talk about trade-off among raising-time, peak time , steady state error.

When Kp>1 damping ratio > 0.7

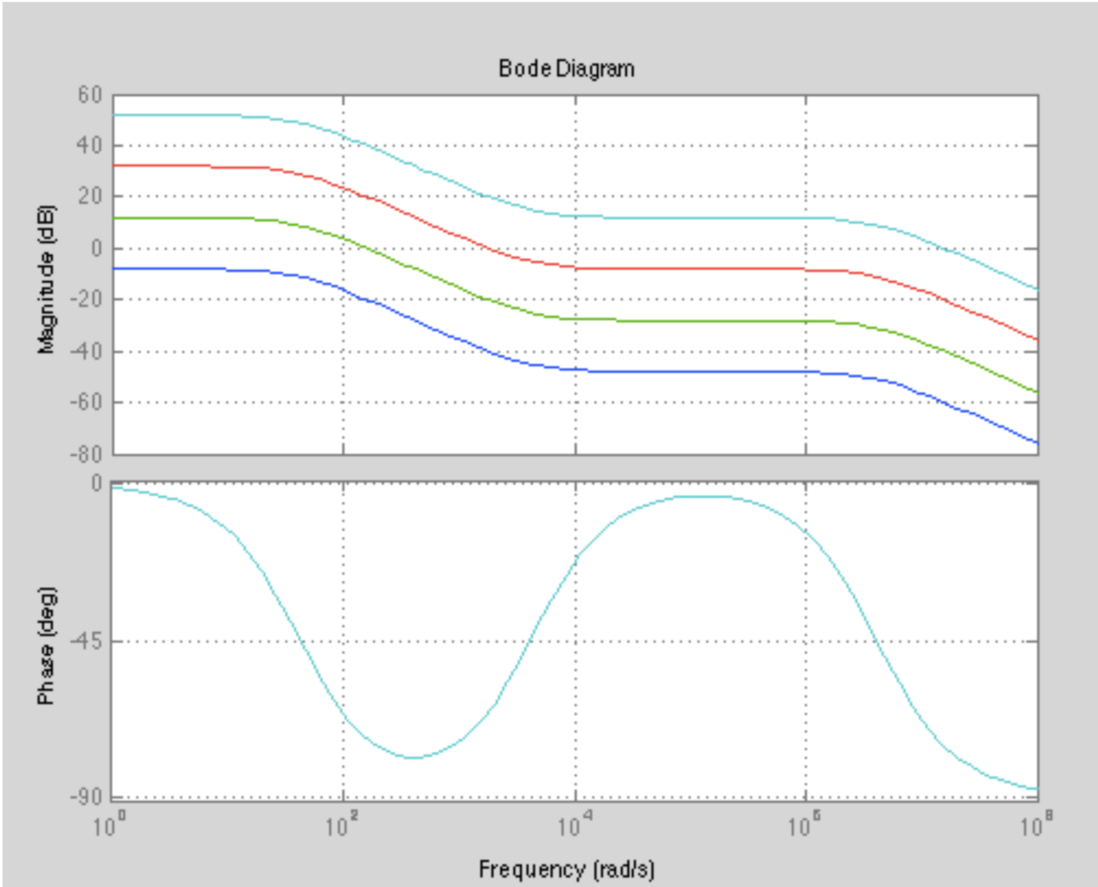
3(2) Outer-loop PI design

Bode open-loop system and run step response.

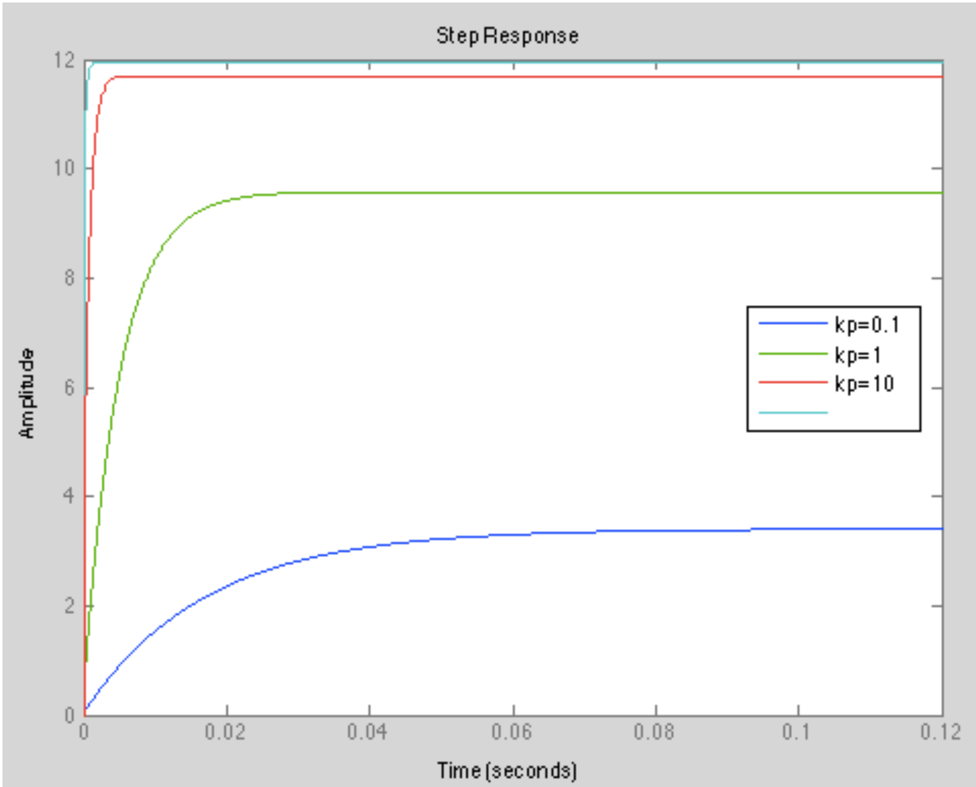
First consider the situation that only P controller implemented in this plant.

```
n=[9.6*10^(-4) 4];
```

```
g=[2.424*10^(-2) 1];

vl=tf(n,g);

p1=clsys*vl; %from inductor current to load voltage

for p=[0.1 1 10 100]

hold on

nc=[p 0];%kp=p, Ki=0

gc=[1 0];

pi=tf(nc,gc);% pi controller

opplant= feedback(pi*p1,0);%open loop tf

figure(6)

bode(opplant);

grid on;

clplant=feedback(pi*p1,1) % close loop tf

figure(8);

step(12*clplant);

end
```

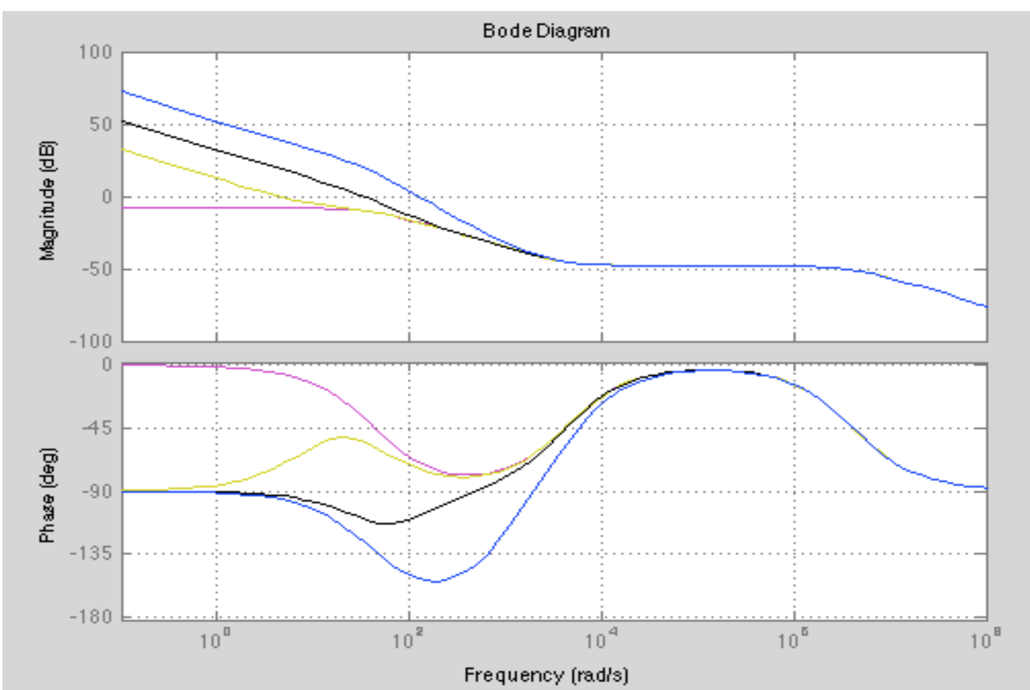


For p=[0.1 1 10 100] magnitude curve been shift up and down, phase curve remain the same.



Step response shows larger kp↑-> t_r ↓-> ess↓

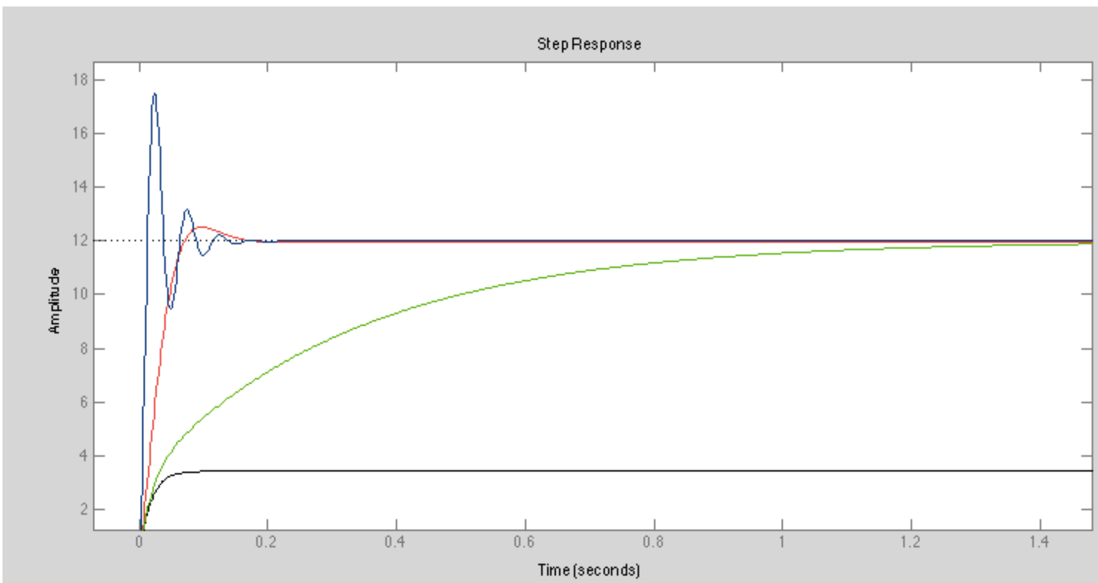
Then we choose fixed P value kp=0.1 and make ki=`[0 0.1 1 10] run bode plot`



Now Pi controller both take affect on magnitude and phase curve.

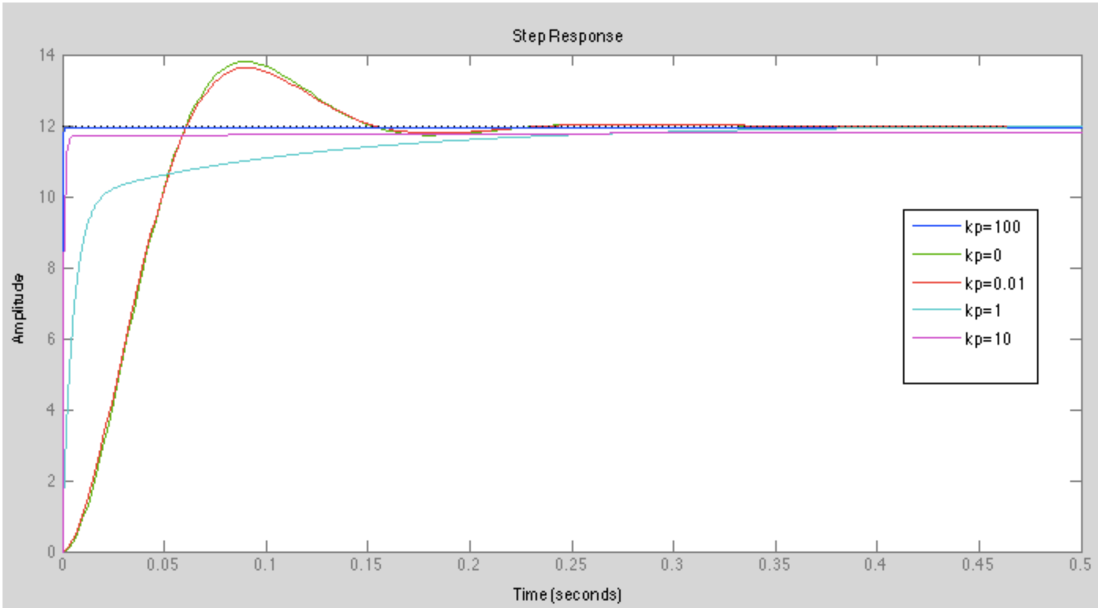
From magnitude curve we can see Ki will change type of system (initial curve slope changes from 0 to 20dB/dec), in this case from Type 0 to Type 1, Also Ki will only take affect for low frequency, both four bode have same high frequency curve.

From phase curve, ki have negative affect of my system, when Ki=0 system have biggest phase margin, when ki is bigger phase curve goes down, when k is bigger than 10 phase curve will touch -180°



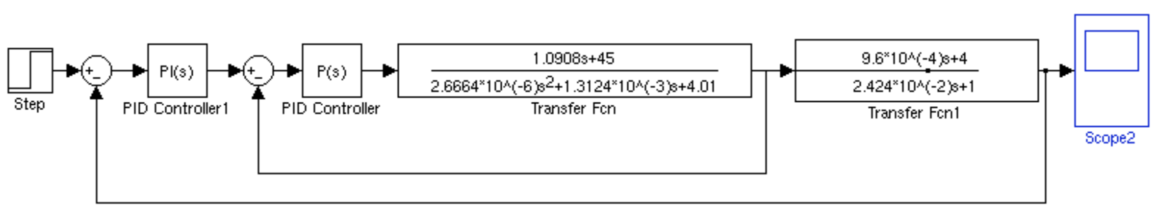
From step response we can see that ki will eliminate steady state error. Since there have an ki in the system, steady state error is always 0. Furthermore, when ki is bigger system response much more faster, but if ki is too big(ki>1 in this case), step response start to have over shoot and oscillation.

Then we choose fixed I value ki=10 and make kp=`[0 0.01 1 10 100] step response`

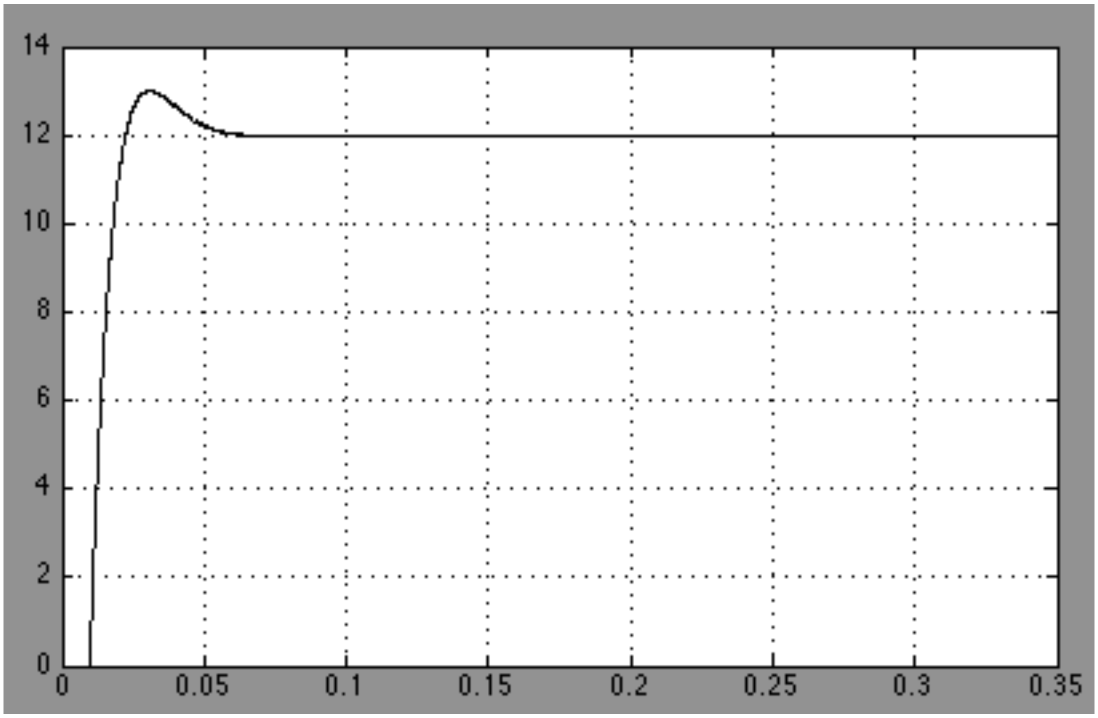


From previous we talk about bigger ki can cause overshoot and oscillation, now Choose a fixed ki=10 and do step response with different kp. From the figure above we can see when kp=0 system have a over shoot , but when kp become bigger not only overshoot disappear, but also system response faster.

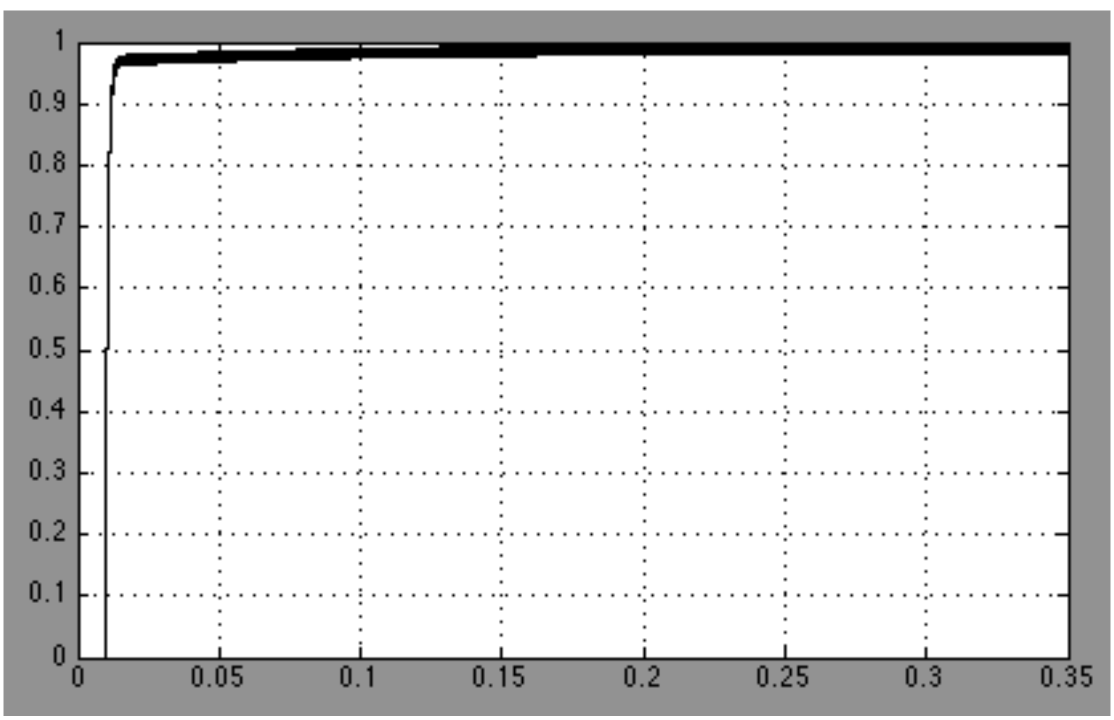
4. Simulate the overall closed-loop system



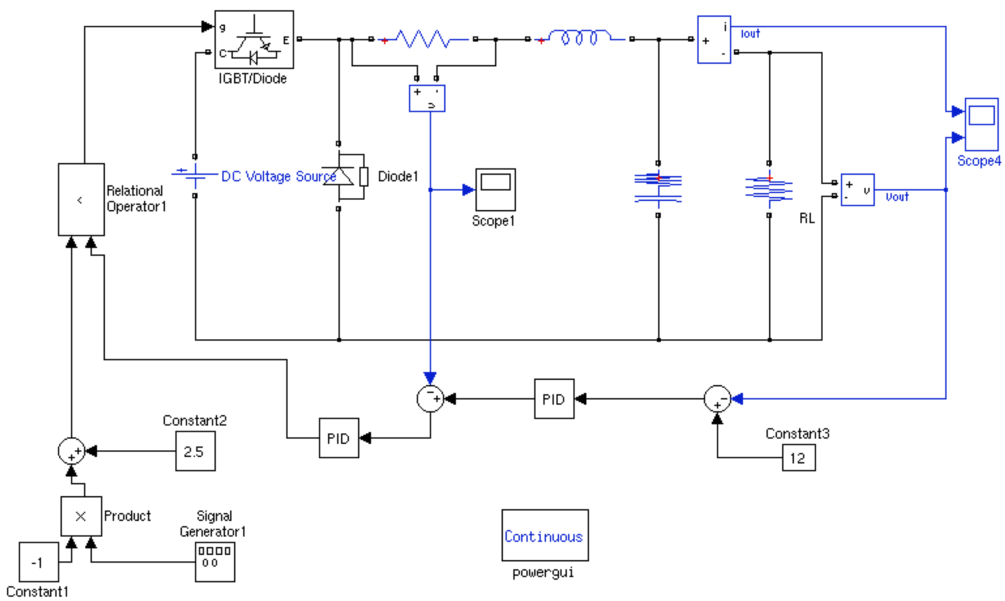
While inner-loop kp=10 outer loop kp=1 ki=100:



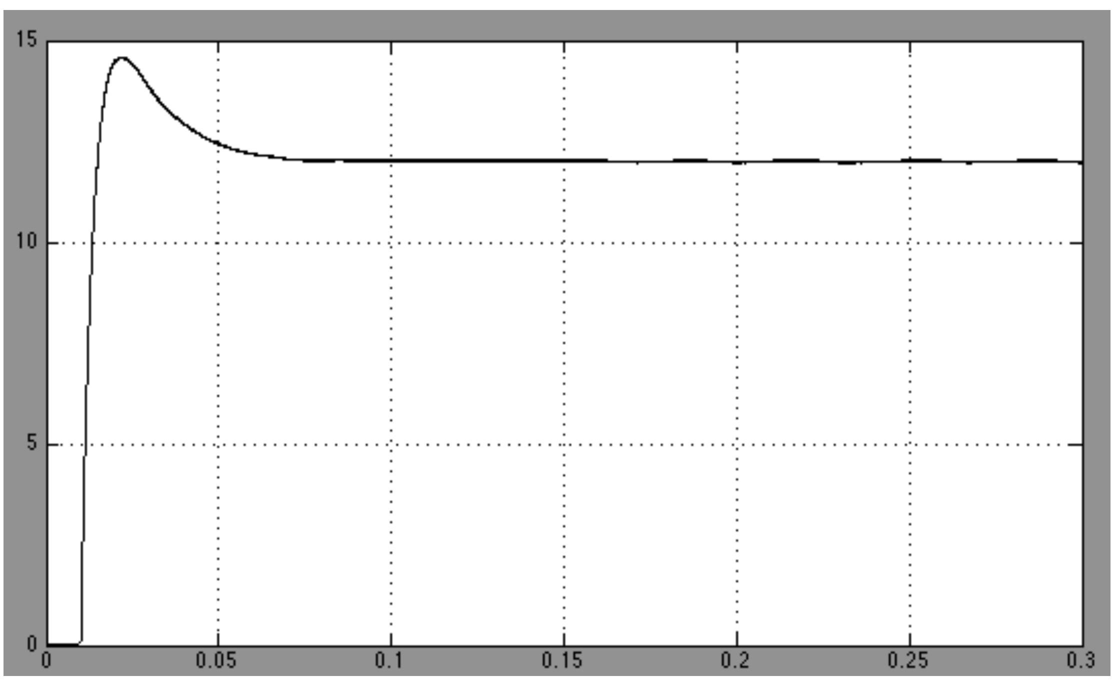
While inner-loop kp=10 outer loop kp=10 ki=100



5. Simulate the overall closed-loop system in circuit form



While inner-loop kp=10 outer loop kp=1 ki=100:



While inner-loop kp=10 outer loop kp=10 ki=100