|  |
| --- |
| **실습[2-2] 2차시스템의 계단응답**  [MATLAB](https://blog.naver.com/PostList.naver?blogId=kckoh2309&categoryNo=59&from=postList) / [IT강좌](https://blog.naver.com/PostList.naver?blogId=kckoh2309&categoryNo=28&parentCategoryNo=28&from=postList)   2020. 9. 13. 15:43 |

모터제어를 위치제어 하는 경우 모터의 플랜트는 다음과 같이 2차모델로 주어진다.



 -------------------------------------------------------------------- (1)

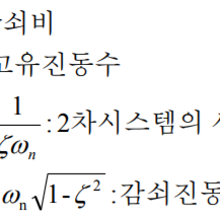
 그리고 이 시스템에 대해 Kp의 P제어로 이루어지는 위치제어시스템의 전체 전달함수는 다음과 같다.



 ----------------------------------------------------- (2)

이 과정은 다음 링크를 참조한다.

<https://blog.naver.com/kckoh2309/222092569490>

[[](https://blog.naver.com/kckoh2309/222092569490)](https://blog.naver.com/kckoh2309/222092569490" \t "_blank)**[제어공학 [3-2] 2차시스템의 시간응답](https://blog.naver.com/kckoh2309/222092569490" \t "_blank)**

[그리고 모터제어를 위치제어 하는 경우 모터의 1차 플랜트 모델)에 적분기가 하나 추가되어 위치제어 모델...](https://blog.naver.com/kckoh2309/222092569490" \t "_blank)

[blog.naver.com](https://blog.naver.com/kckoh2309/222092569490" \t "_blank)



따라서 식(2)의 2차 모델을 다음과 같이 전체 전달함수를 표현할 때,



   -------------------------------------------------------(3)

모터의 전압-속도 모델 파라미터(DC게인, 시상수)와 제어기 게인(Kp)로 부터, 전체 시스템의 감쇠비와 고유진동수를 구하는 매트랩프로그램을 작성해 보자.(여기서 Kp=60, K=1, Tconst=0.024로 한다)

[model\_2ndOder\_motor.m]

Kp=input("Proportional Gain(Kp)=");

K=1;

Tconst=0.024;

omegan=sqrt(Kp\*K/Tconst)

cheta=1/(2\*Tconst\*omegan)

num=[omegan^2];

den=[1 2\*cheta\*omegan omegan^2];

printsys(num,den)

실행결과는 다음과 같다.

Proportional Gain(Kp)=60

omegan =

    50

cheta =

    0.4167

num/den =

            2500

   ----------------------

   s^2 + 41.6667 s + 2500

다음은 위 매트랩 코드를 파이썬 코드로 구현한 예이다.('22.5.4)

from math import \*

Kp=int(input("Proportional Gain(Kp)="))

K=1;Tconst=0.024

omegan=sqrt(Kp\*K/Tconst)

cheta=1/(2\*Tconst\*omegan)

print("omegan=%.2f\n cheta=%.4f\n"%(omegan,cheta))

a=2\*cheta\*omegan;b=omegan\*\*2

print("G(s)=%.2f/(s^2+%.2fs+%.2f)\n"%(b,a,b))

실행결과:

omegan=50.00

 cheta=0.4167

G(s)=2500.00/(s^2+41.67s+2500.00)

식(3)으로 주어지는 2차 시스템에 대해  다음과 같이 입력함수가 계단입력으로 주어지면,



-------------------------------------------------------------------------------------(4)



는 다음과 같이 구할 수 있다.



 ------------------------------------------------------(5)

식(5)를 역라플라스 변환하면,



----------------------------(6)

 여기서,



 ------------------------------------------------------------------------------(7)

이다.

식(4)~(6)의 유도과정은 다음글을 참조한다.

<https://blog.naver.com/kckoh2309/222092569490>

이제  식(6)를 적용하여, 2차 시스템의 시간응답을 구하는 매트랩 프로그램을 작성해 보자.(여기서 Kp=60, K=1, Tconst=0.024로 한다)

[Second\_orderResp\_Motor.m]

Kp=input("Proportional Gain(Kp)=");

K=1;

Tconst=0.024;

omegan=sqrt(Kp\*K/Tconst);

cheta=1/(2\*Tconst\*omegan);

omegad=sqrt(1-cheta^2)\*omegan;

t=[0:0.001:0.3];

for i=1:1:301,

    r(i)=1;

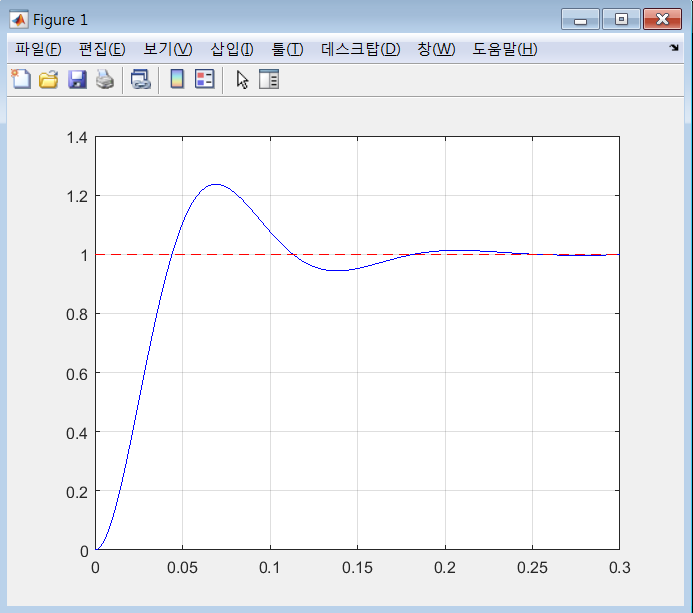
    x(i)=1-exp(-cheta\*omegan\*t(i))\*(cos(omegad\*t(i))+cheta\*sin(omegad\*t(i))/sqrt(1-cheta^2));

end

plot(t,x,'b',t,r,'--r')

grid on

실행결과는 다음과 같다.



이를 파이썬 코드로 구현한 예이다.('22.5.4)

- 여기서 파이썬 코드 작성시 주의할 점

 > 벡터연산을 위해 math 대신 numpy를 사용(step\_2nd\_system()함수에서 t와 x는 array변수임을 주의)

 > 따라서, exp(), cos(), sin(), sqrt()함수들이 numpy에서 제공되는 함수(벡터-스칼라 변환 연산 가능)

 > arange(), array()함수등은 리스트 변수를 numpy 어레이변수로 변환

 > matplotlib에서 grid가 안그려지는 이유(보완예정)

from numpy import \*

import matplotlib.pyplot as plt

def step\_2nd\_system(t):

    x=1.0-exp(-cheta\*omegan\*t)\*(cos(omegad\*t)+cheta\*sin(omegad\*t)/sqrt(1-cheta\*\*2))

    return x

plt.style.use('seaborn-whitegrid')

Kp=int(input("Proportional Gain(Kp)="))

K=1;Tconst=0.024

omegan=sqrt(Kp\*K/Tconst)

cheta=1/(2\*Tconst\*omegan)

omegad=sqrt(1-cheta\*\*2)\*omegan

print("omegan=%.2f\n cheta=%.4f\n"%(omegan,cheta))

a=2\*cheta\*omegan;b=omegan\*\*2

print("G(s)=%.2f/(s^2+%.2fs+%.2f)\n"%(b,a,b))

t=arange(0,0.3,0.001)

r=array([1.0 for i in range(300)])

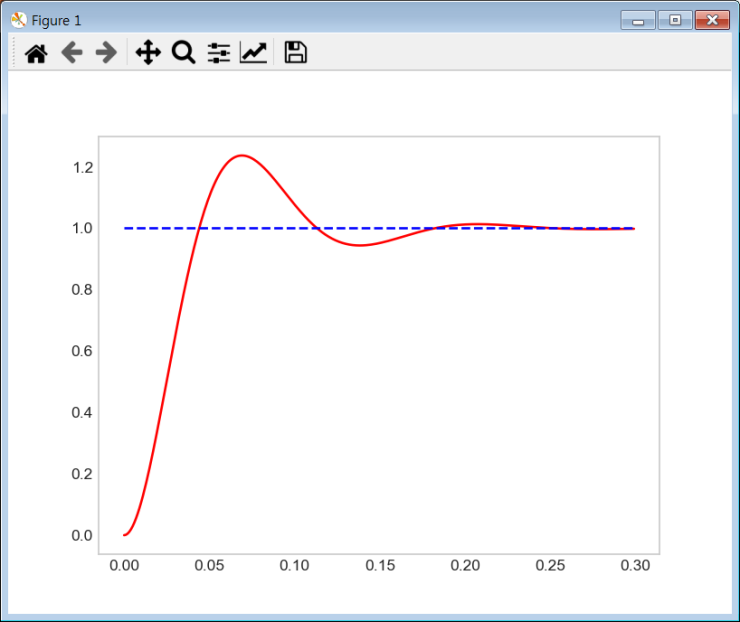
x=step\_2nd\_system(t)

plt.grid()

plt.plot(t,x,'r-',t,r,'b--')

plt.show()

실행결과



이번에는 같은 문제를 matlab이 제공하는 step.m함수를 이용하여 풀어보자. 이를 위한 m코드는 다음과 같다.

[step\_resp\_2nd\_order.m]

Kp=input("Proportional Gain(Kp)=");

K=1;

Tconst=0.024;

omegan=sqrt(Kp\*K/Tconst);

cheta=1/(2\*Tconst\*omegan);

omegad=sqrt(1-cheta^2)\*omegan;

t=[0:0.001:0.3];

for i=1:1:301,

    r(i)=1;

    %x(i)=1-exp(-cheta\*omegan\*t(i))\*(cos(omegad\*t(i))+cheta\*sin(omegad\*t(i))/sqrt(1-cheta^2));

end

num=[omegan^2];

den=[1 2\*cheta\*omegan omegan^2];

x=step(num,den,t);

plot(t,x,'b',t,r,'--r')

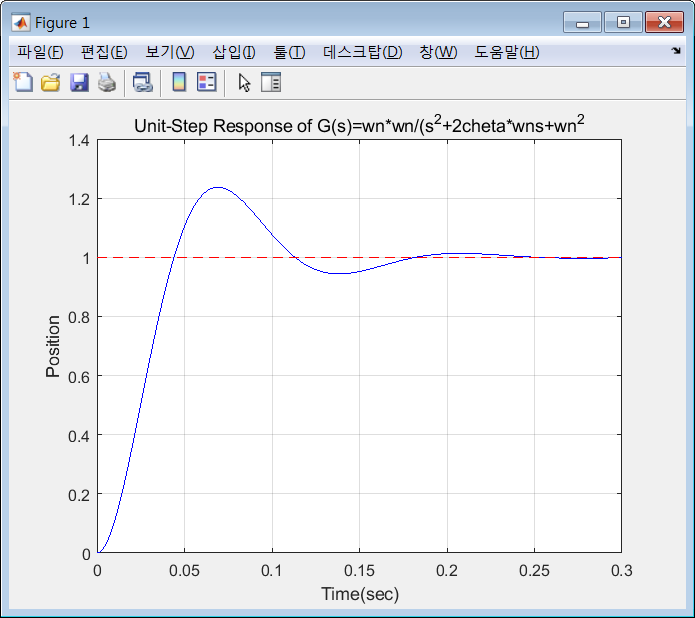
grid on

title('Unit-Step Response of G(s)=wn\*wn/(s^2+2cheta\*wns+wn^2')

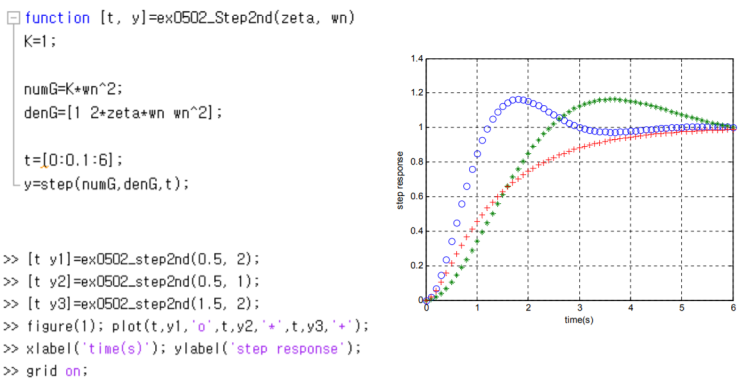
xlabel('Time(sec)')

ylabel('Position')

실행결과는 다음과 같다.



다음과 같이 함수를 만들어서 3가지 감쇄비에 대한 응답곡선을 그려보자



[참고글] 이전에 작성했던 문서입니다.



[model\_2ndorder.m]

kp=input('Stiffness =');

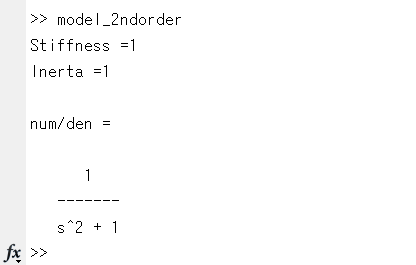
J=input('Inerta =');

num = [0   0   Kp];

den = [J   0   Kp];

printsys(num, den)

그리고 실행화면은 다음과 같다.



이번에는 다음과 같이 2차시스템으로 주어지는 계에 대해, 스텝입력에 대해 시간응답을 구해보자.



이를 구하기 위한 m코드는 다음과 같다.

(step\_response.m)

t=[0:0.01:3];

for i=1:1:301,

    r(i)=1;

end

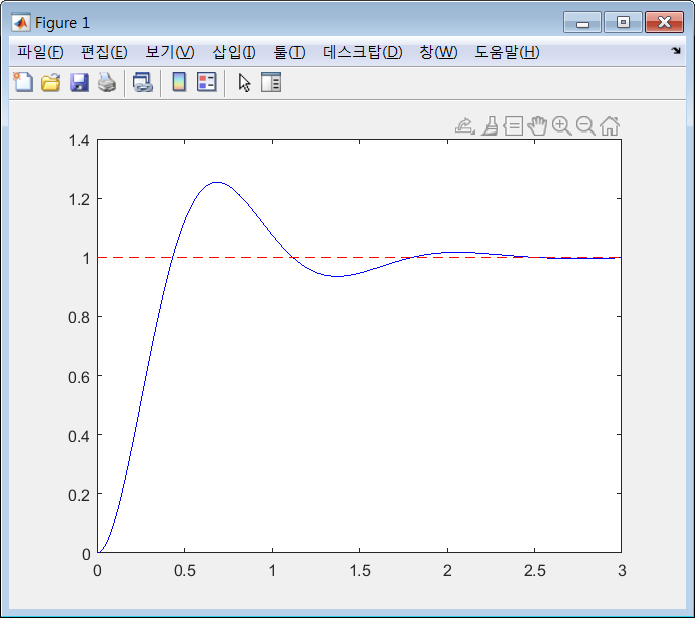
num=[25];

den=[1 4 25];

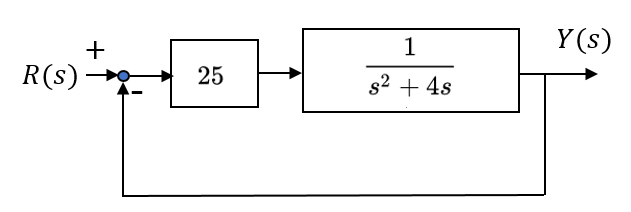
y=step(num,den,t);

plot(t,y,'b',t,r,'--r'))

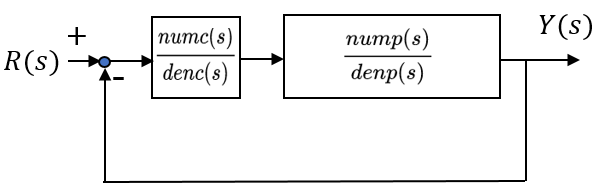
실행결과는 다음과 같다.



이번에는 다음과 같이 페루프시스템을 예로 들어 보자.



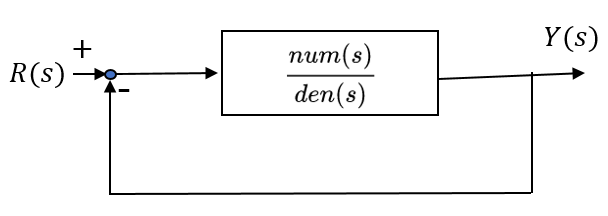
이와 같은 시스템 다음과 같이 모델링 할 수 있다.



그러면 [numc,denc]와 [nump,nump]는 직렬로 연결된다. 따라서 전체 전달함수는 다음과 같은 series()함수를 호출하여 구할 수 있다.

[num,den]=series(nump,denp,numc,denc);

그리고 다시 [num,den]는 그림과 같이 페루프(closed loop)시스템을 구성한다.



따라서 다음과 같은 cloop()함수를 호출하여 전체 시스템을 구성할 수 있다.

[num,den]=cloop(num,den,-1);

다음은 이 같은 함수를 이용하여 매트랩코드를 작성한 예이다.

(model\_feedback.m)

t=[0:0.01:3];

for i=1:1:301,

    r(i)=1;

end

nump=[1];

denp=[1 4 0];

numc=[50];

denc=[1];

[num,den]=series(nump,denp,numc,denc);

[num,den]=cloop(num,den,-1);

printsys(num,den);

y=step(num,den,t);

plot(t,y,'b',t,r,'--r')

그리고 실행한 결과 명령창은 다음과 같다.

 num/den =

         50

   --------------

   s^2 + 4 s + 50

그리고 그림은 다음과 같다.

