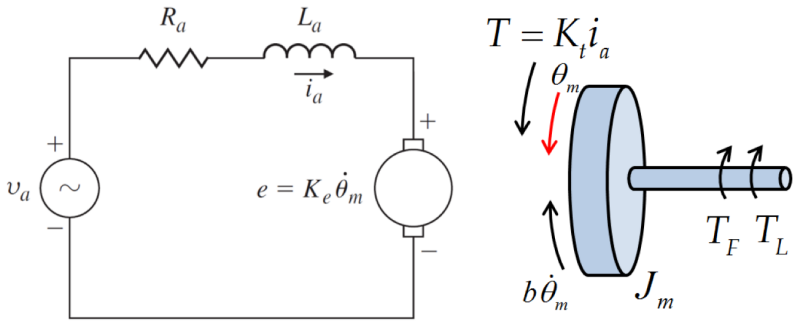
자동제어

Term Project 2

로봇학부
2018741035
한준호

The objective is to control the angular velocity $\omega_m = \dot{\theta}_m$ of the dc motor. The electric circuit and the mechanical parts are shown below. (For detailed description of the model, see the example in the textbook.) Some friction torque T_F and load torque T_L are present.



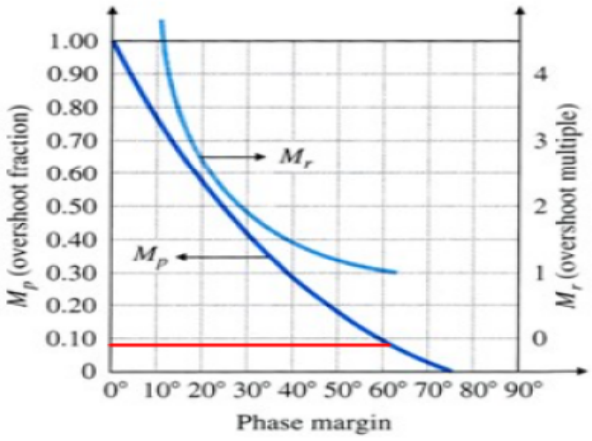
Physical parameters are summarized as follows.
$J_m = 0.05\ \text{kg} \cdot m^2, b = 0.006\ \text{N} \cdot m \cdot sec,\ K_e = 0.05\ \text{V} \cdot sec,\ K_t = 0.05\ \text{N} \cdot \text{m/A}, R_a = 10\ \Omega, L_a = 0.01\ \text{H}$. Note that the values of this problem can be nonrealistic.

4. We would like to design a compensator (lead, lag) using the frequency response techniques.

1) Change the performance index of the closed-loop system you've chosen in Problem 2 into those in the frequency domain (for example, steady state error, phase margin, etc.)

지난 Problem 2에서 설정한 Performance index는 다음과 같다.

Performance Index	
t_r	1.28 sec
w_n	1.406
M_p	9.830%
ζ	0.594
t_s	5.5079 sec
σ	0.83516



$$\zeta \cong \frac{\mathrm{PM}}{100}$$

ζ = 0.594이므로 PM은 59.4이다. 또한 M_p = 9.830%을 위 그림에 적용하여 Phaze margin을 구하면 60°정도가 된다. 그리고 steady state error(e_{ss})는 R(s) = $\frac{1}{s}$일 때 system type 0이기 때문에

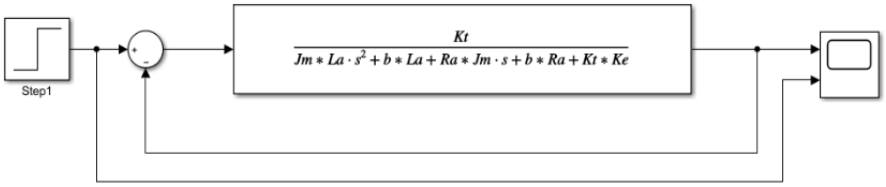
$$e_{ss} = \lim_{s \to 0} s \frac{1}{1 + G(s)} R(s)$$ = 0.5555이 나온다. $e_{ss} \leq 0.1$로 설정한다.

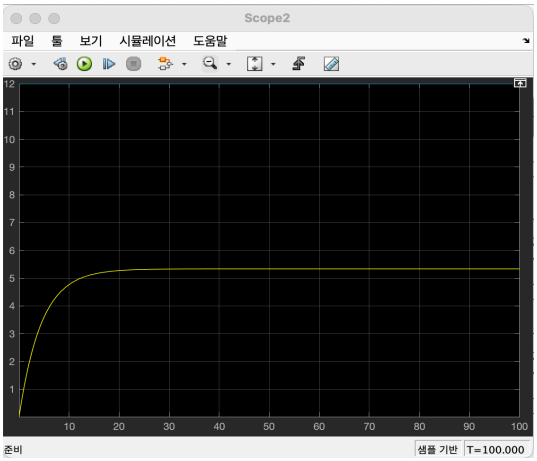
2) Design a compensator which satisfies the performance index you chose in 1). Evaluate your controller by simulation. You should use Bode plots as well as Nyquist plots to design your compensator.

$$G(s) = \frac{K_t}{(J_m s + b)(R_a + sL_a) + K_e K_t}$$

```
Jm = 0.05;
b = 0.006;
Ke = 0.05;
Kt = 0.05;
Ra = 10;
La = 0.01;
```

이를 Compensator 없이 Closed-loop로 설계하면



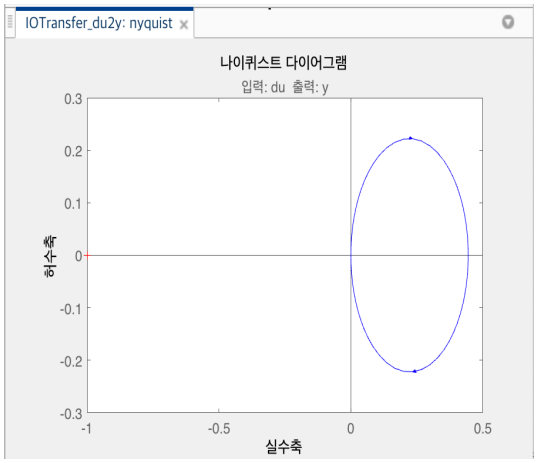


다음과 같이 입력을 12V로 주었을 때 입력을 따라가지 못하고 steady state error를 갖는 것을 확인할 수 있다. 이를 해결하기 위해 Compensator를 이용해야한다.

```
Num = Kt;
Den = [La*Jm Jm*Ra+La*b Ra*b+Ke*Kt];
sys = tf(Num,Den);
sisotool(sys);
```

〈 Matlab 스크립트 〉

'sisotool' 을 통해 Nyquist plot과 Bode plot을 확인한다.



Nyquist plot을 확인 해본 결과, $G(s)=-\frac{1}{K}<0$ 이므로 K 〉0 일 때 stable한 것을 알 수 있다.

- Lag compensator 설계

Steady state error를 줄이기 위해 Lag compensator를 추가한다.

$$D(s)=\alpha\frac{Ts+1}{\alpha Ts+1},\alpha>1$$

Steady state error를 구하는 공식을 이용하여 조건을 만족하는 Lag compensator를 설계한다.

$$e_{ss}=\lim_{s\to 0}s\frac{1}{1+KD(s)G(s)}R(s)\quad R(s)=\frac{1}{s}$$

$$e_{ss}=\lim_{s\to 0}\frac{1}{1+KD(s)\dfrac{K_t}{[(J_ms+b)(R_\alpha+sL_\alpha)+K_eK_t]}}$$

$$e_{ss}=\frac{1}{1+KD(0)\dfrac{K_t}{bR_\alpha+K_eK_t}}\le 0.1$$

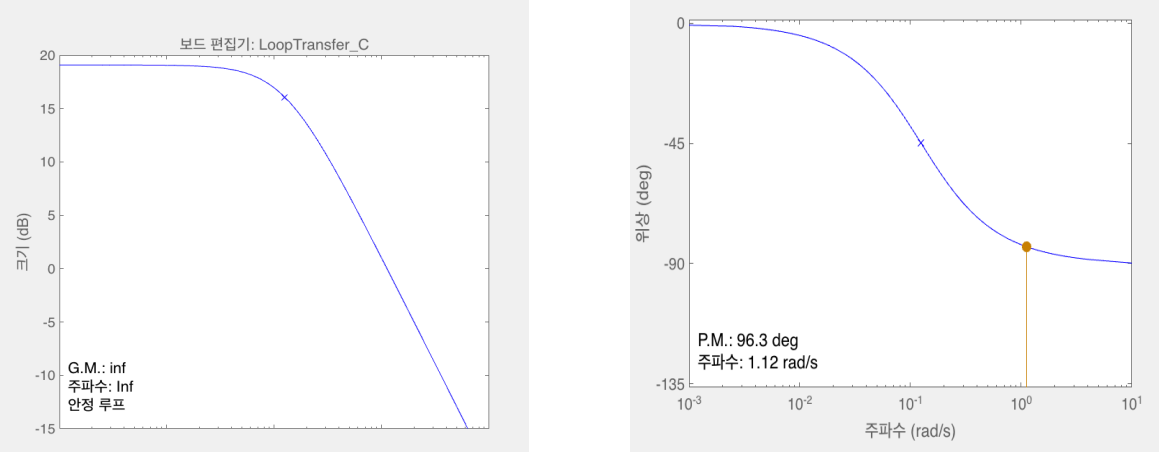
$$e_{ss}=\frac{1}{1+KD(0)*0.8}\le 0.1,D(0)=\alpha$$

$$e_{ss}=\frac{1}{1+K\alpha*0.8}\le 0.1,K\alpha=K_p$$

$$e_{ss}=\frac{1}{1+K_p*0.8}\le 0.1$$

$$K_p\ge 11.25$$

따라서, Kp = 11.25으로 설정하여 ‘sisotool’ 을 통해 Bode plot을 그린다.



〈 Bode plot 〉

결과를 통해 Phase Margin = 96.3°인 것을 확인할 수 있는데 Phase Margin = (-83.7°) - (-180°) = 96.3° 이므로 Magnitude = 1인 점에서의 Phase를 구하면 약 -83.7° 정도이다. 목표했던 M_p = 9.830%에 대한 Phaze margin인 60°이고 96.3° 〉 60° 이므로 조건을 만족하였다.

Frequency ≈ 0일 때, Magnitude는 약 20이 되는 것을 이용하여, 적절한 K값을 설정하여 α를 구한다.

$$20 \times log(K) = 20$$

$$K = 10$$

$$\alpha = \frac{K_p}{K} = \frac{11.25}{10} = 1.125$$

또, Magnitude가 1인 점에서의 주파수(w_c)가 약 1.12 정도이다. 이 때 Coner frequency(w)는 Crossover frequencyd(w_c)의 0.1배로 설정한다.

$$w = 0.1w_c = 0.112$$

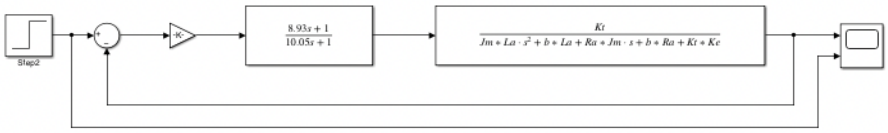
$$T = \frac{1}{w} = 8.93$$

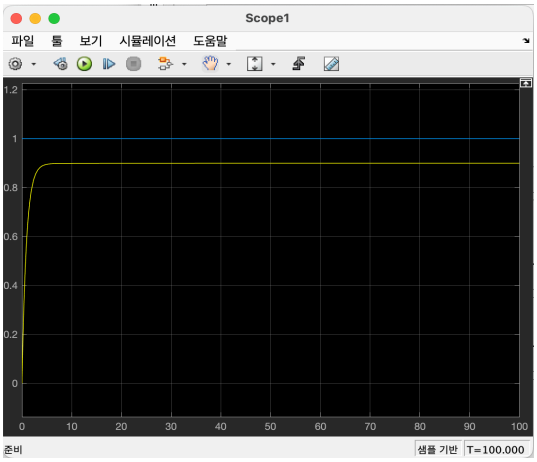
이 값들을 이용하여 Lag compensator를 설계하면

$$D(s) = 1.125\frac{8.93s + 1}{10.05s + 1}$$

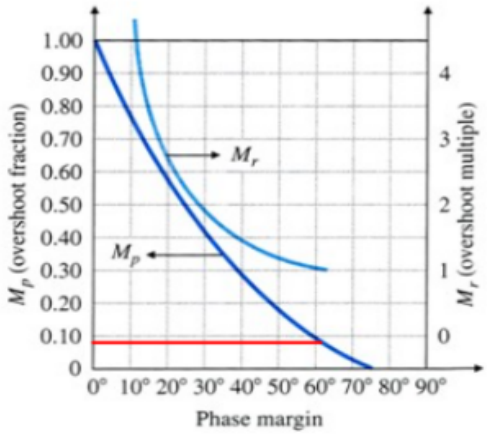
$$KD(s) = K \times 1.125\frac{8.93s + 1}{10.05s + 1} = 11.25\frac{8.93s + 1}{10.05s + 1}$$

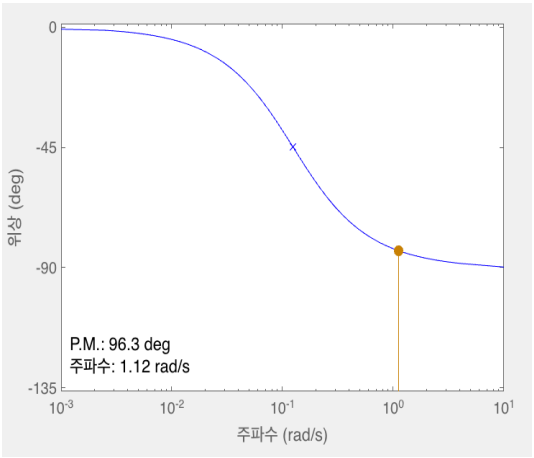
이 Lag compensator를 사용한 Block diagram은 다음과 같다.





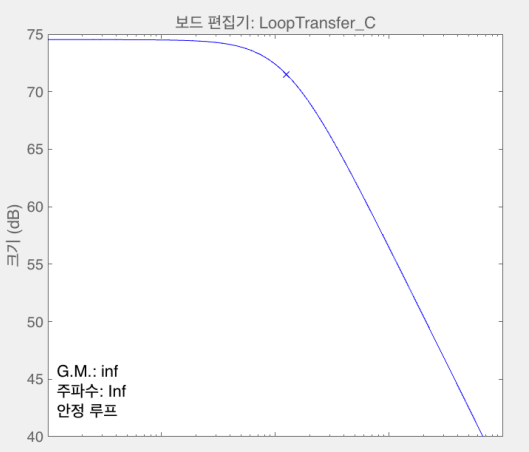
결과를 보면 Input을 1로 주었을 때, 처음에 설정한 0.1의 steady state error를 만족하며 0.9정도에 수렴하는 것을 확인할 수 있다.

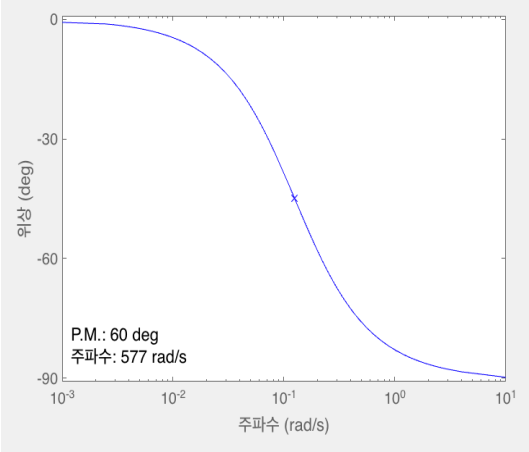




하지만, PM = 96.3인 Overshoot는 왼쪽 그래프에서 확인할 수 없다. 그러므로 이를 사용하는 것은 부적절하다는 것을 알 수 있다. 이를 해결하기위해 K_p의 값을 올린다.

K_p값이 6600일 때, PM이 60°인 것을 확인했다.





위 과정을 반복하면,

Frequency ≈ 0일 때, Magnitude는 약 75이 되는 것을 이용하여 적절한 K값을 설정하여 α를 구한다.

$$20 \times log(K) = 75$$

$$K = 5623$$

$$\alpha = \frac{K_p}{K} = \frac{6600}{5623} = 1.174$$

또, Magnitude가 1인 점에서의 주파수(w_c)가 약 577 정도이다. 이 때 Coner frequency(w)는 Crossover frequencyd(w_c)의 0.3배로 설정한다.

$$w = 0.3w_c = 173.1$$

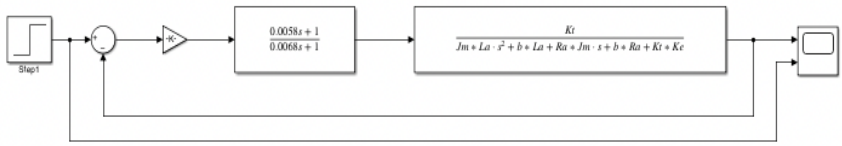
$$T = \frac{1}{w} = 0.0058$$

이 값들을 이용하여 Lag compensator를 설계하면

$$D(s) = 1.174\frac{0.0058s + 1}{0.0068s + 1}$$

$$KD(s) = K \times 1.174\frac{0.0058s + 1}{0.0068s + 1} = 6600\frac{0.0058s + 1}{0.0068s + 1}$$

이 Lag compensator를 사용한 Block diagram은 다음과 같다.

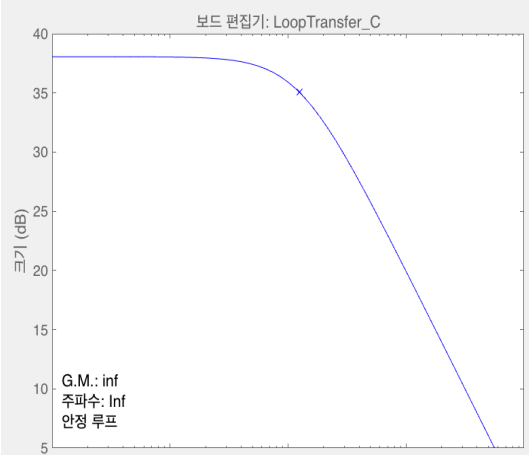


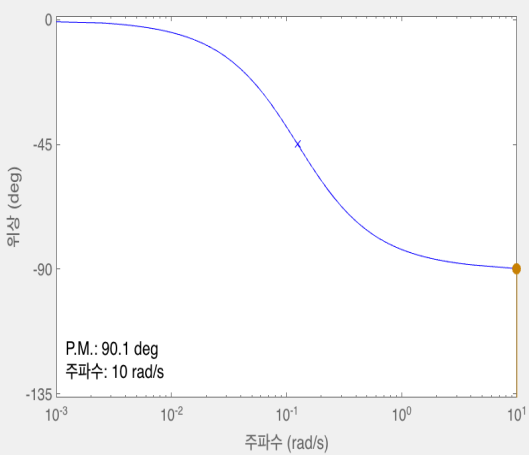
K_p가 6600일 때 1로 잘 수렴하는 것을 볼 수 있다.

Lag Compensator 설계를 하였을 때 K_p의 값을 조정함으로써 pole의 위치를 이동하지 않고 사양에 맞는 System을 설계할 수 있음을 알 수 있다.

3) Repeat 4) of problem 2.

위에서 설계한 System은 simulation 결과 수렴 시간이 너무 빨라서 모터에 적용할 수 없다. 그래서 적당한 Kp값인 100으로 아래 문제를 수행하였다.





Frequency ≈ 0일 때, Magnitude는 약 40이 되는 것을 이용하여 적절한 K값을 설정하여 α를 구한다.

$$20 \times log(K) = 37.5$$

$$K = 80$$

$$\alpha = \frac{K_p}{K} = \frac{100}{80} = 1.25$$

또, Magnitude가 1인 점에서의 주파수(w_c)가 약 10 정도이다. 이 때 Coner frequency(w)는 Crossover frequencyd(w_c)의 0.3배로 설정한다.

$$w = 0.3w_c = 3$$

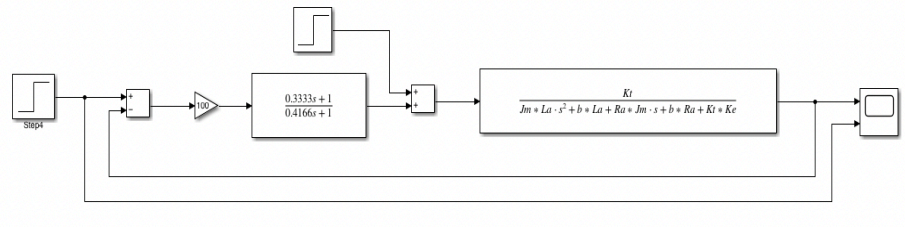
$$T = \frac{1}{w} = 0.3333$$

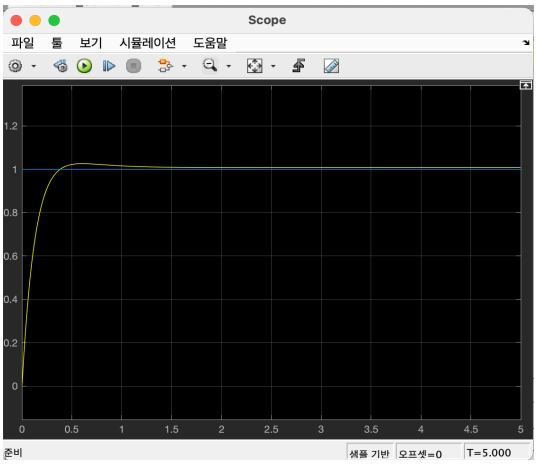
이 값들을 이용하여 Lag compensator를 설계하면

$$D(s) = 1.25\frac{0.3333s + 1}{0.4166s + 1}$$

$$KD(s) = K \times 1.25\frac{0.3333s+1}{0.4166s+1} = 100\frac{0.3333s+1}{0.4166s+1}$$

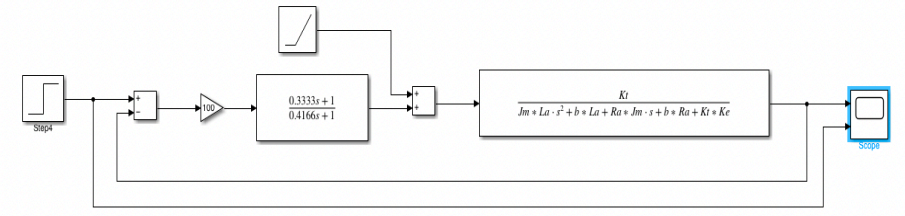
1. Disturbance 입력이 Step의 형태 일 때

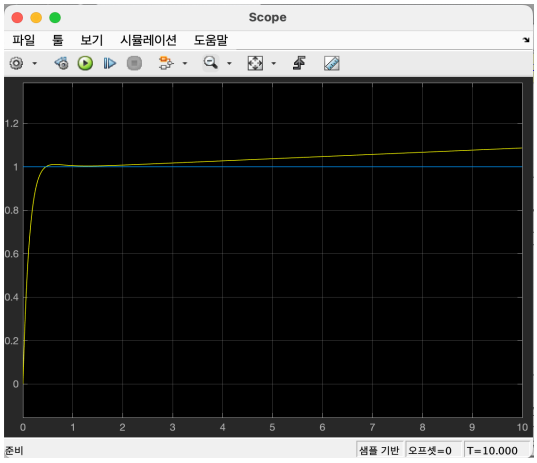




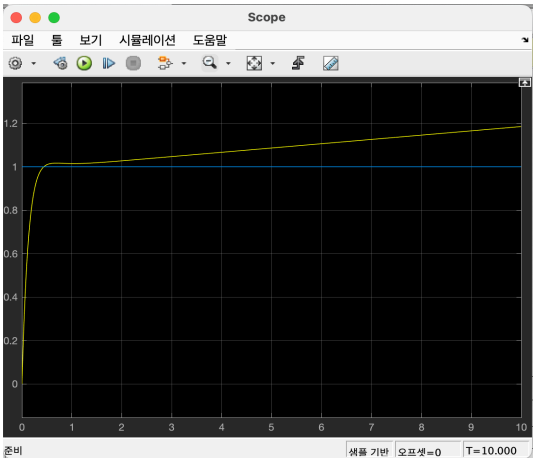
위 그래프는 0s부터 disturbance인 step input값을 2로 넣어주었고, steady state error가 존재하는 것을 확인할 수 있다. 따라서 system type은 0차이다.

1. Disturbance 입력이 Ramp의 형태 일 때





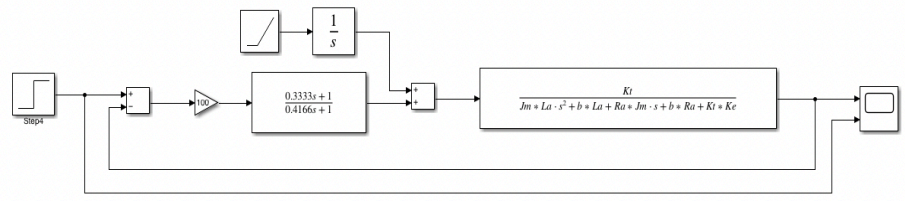
〈 Ramp input slope: 1 〉

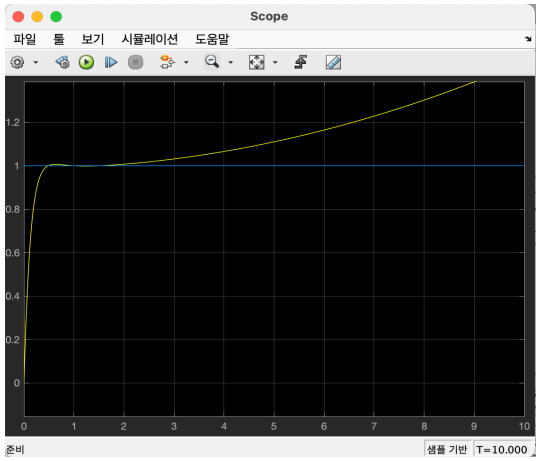


〈 Ramp input slope: 2 〉

왼쪽 그래프은 disturbance로 slope 값이 1인 Ramp input을 넣어주었고, 오른쪽 그래프는 disturbance로 slope 값이 2인 Ramp input을 넣어준 것이다.
결과를 보면 Reference를 따라가지 못하고 Ramp외란에 비례하여 증가하여 발산하는 것을 볼 수있다. 이는 type 0차이기 때문이다.

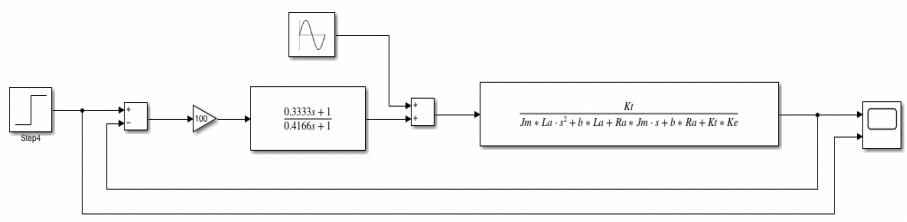
3. Disturbance 입력이 Parabola의 형태일 때

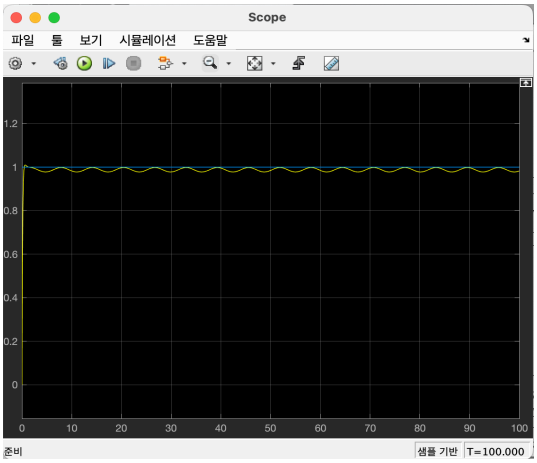




위의 결과를 보면 발산하는 것을 확일 할 수 있다. 이는 system type이 0이기 때문에 parabola 형태의 disturbance에서 발산하는 것이다.

4. Disturbance 입력이 Sinusoidal의 형태일 때





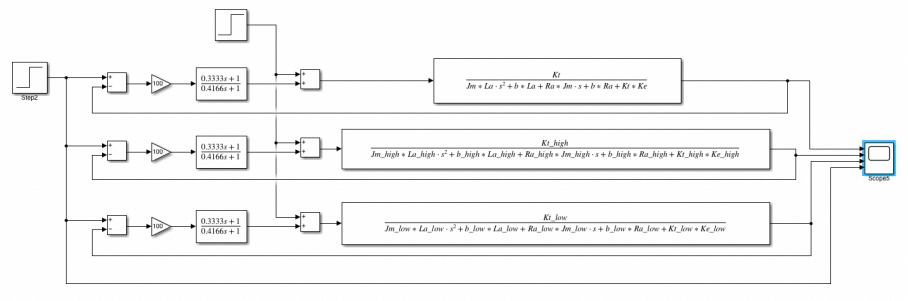
위의 그래프를 보면 sinusoidal 형태의 disturbance에 대해서 수렴하지 못하고 진동하는 것을 확인할 수 있다.

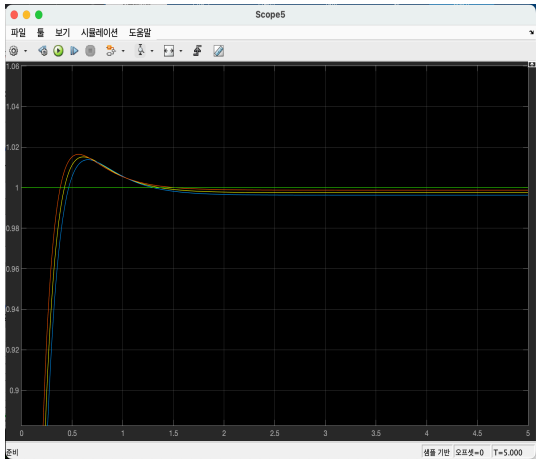
4) Repeat 5) of problem 2.

변수이름	기존값	10% 감소	10% 증가
J_m	0.05	0.045	0.055
B	0.006	0.0054	0.0066
K_e	0.05	0.045	0.055
K_t	0.05	0.045	0.055
R_a	10	9	11
L_a	0.01	0.009	0.011

Matlab 정의	`Jm = 0.05;`
	`b = 0.006;`
	`Ke = 0.05;`
	`Kt = 0.05;`
	`Ra = 10;`
	`La = 0.01;`
		`Jm_low = 0.045;`
		`b_low = 0.0054;`
		`Ke_low = 0.045;`
		`Kt_low = 0.045;`
		`Ra_low = 9.0;`
		`La_low = 0.009;`
			`Jm_high = 0.055;`
			`b_high = 0.0066;`
			`Ke_high = 0.055;`
			`Kt_high = 0.055;`
			`Ra_high = 11.0;`
			`La_high = 0.011;`

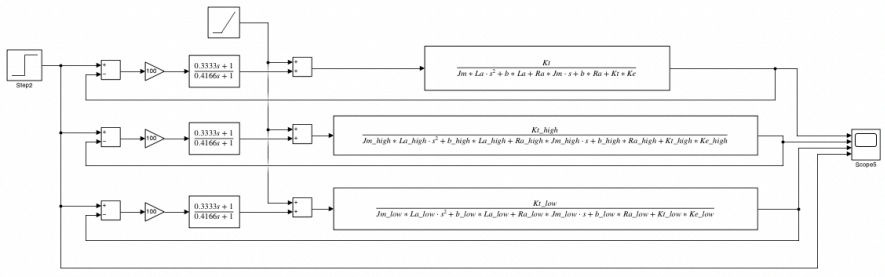
1. Disturbance 입력이 Step 형태일 때

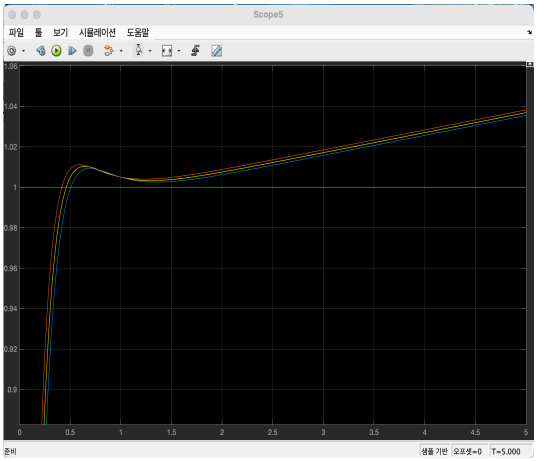




빨간색은 parameter를 10% 감소시킨 그래프, 파란색은 parameter를 10% 증가시킨 그래프, 노란색이 기존 값의 그래프이다. System parameter의 크기가 달라졌음에도 파형의 변화 없이 disturbance로 인해 steady state error가 존재한 채로 수렴하는 것을 확인할 수 있다.

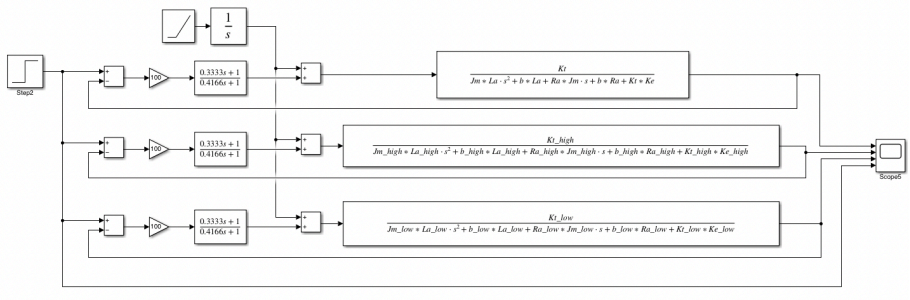
2. Disturbance 입력이 Ramp(slope: 1)의 형태일 때

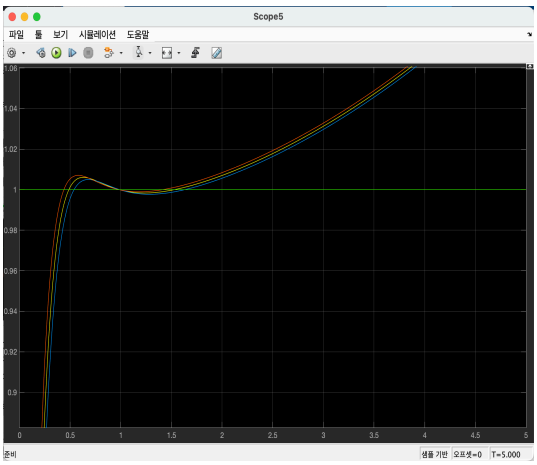




빨간색은 parameter를 10% 감소시킨 그래프, 파란색은 parameter를 10% 증가시킨 그래프, 노란색이 기존 값의 그래프이다. system parameter의 크기가 달라졌음에도 파형의 변화 없이 Angular velocity가 증가하여 발산하는 것을 볼 수 있다. 이는 system type이 0이기 때문이다.

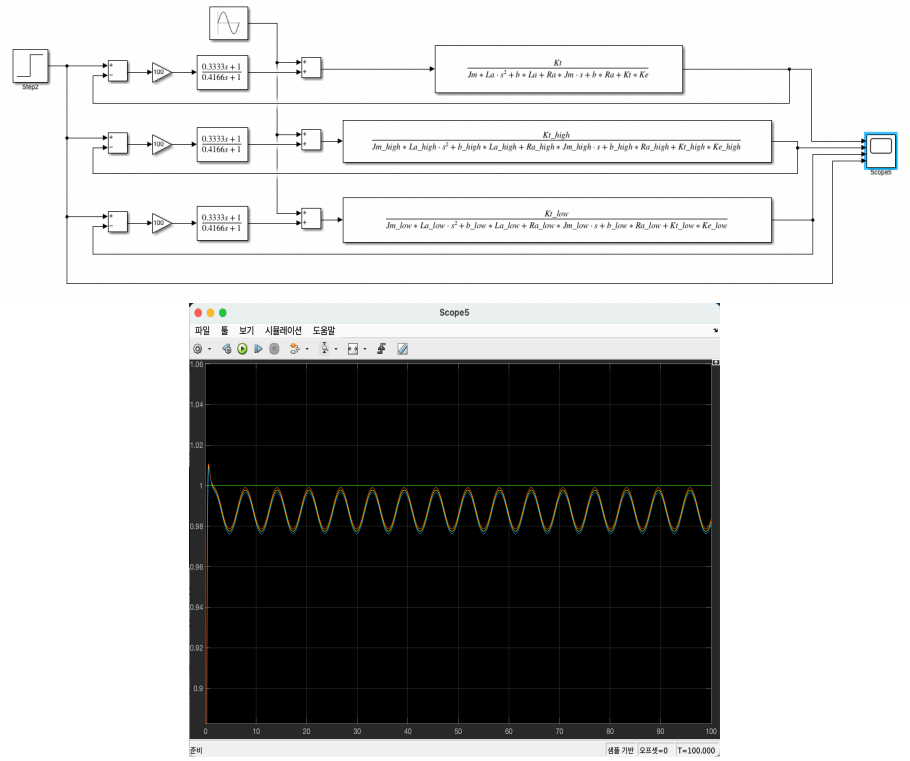
3. Disturbance 입력이 Parabola 형태일 때





빨간색은 parameter를 10% 감소시킨 그래프, 파란색은 parameter를 10% 증가시킨 그래프, 노란색이 기존 값의 그래프이다. 위의 결과를 보면 system parameter의 크기가 달라졌음에도 system type이 0이기 때문에 파형의 변화 없이 모두 발산하는 것을 확인할 수 있다.

4. Disturbance 입력이 Sinusoidal 형태일 때



빨간색은 parameter를 10% 감소시킨 그래프, 파란색은 parameter를 10% 증가시킨 그래프, 노란색이 기존 값의 그래프이다. 위의 그래프를 보면 system parameter의 크기가 달라졌음에도 파형의 변화 없이 수렴하지 못하고 진동하는 것을 확인할 수 있다.

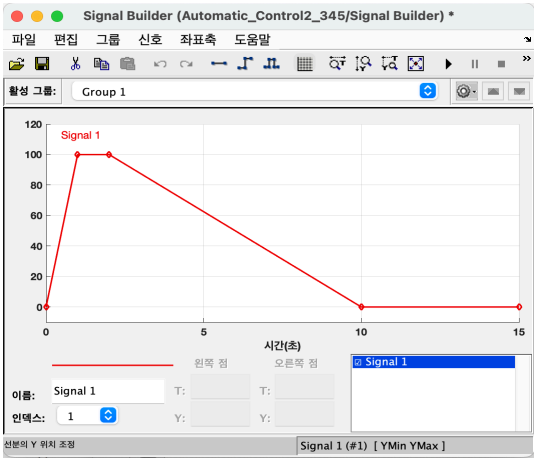
결과적으로 parameter 크기에 관계없이 system이 유지되는 것을 확인할 수 있다. 따라서 system parameter가 바뀌면 rising time, settling time, overshooting의 값에는 약간의 변화가 있지만 큰 영향을 미치지는 않는 것을 알 수 있다.

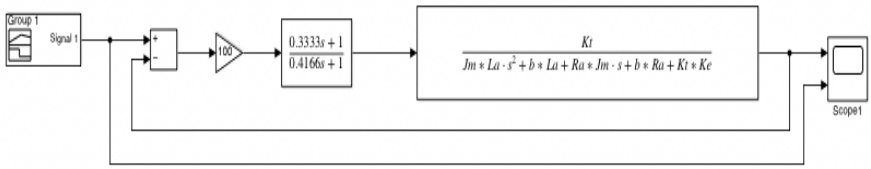
5) Repeat 6) of problem 2.

$$r_0 = 100 \ (\geq 10 * 2\pi = 62.831863\cdots)$$

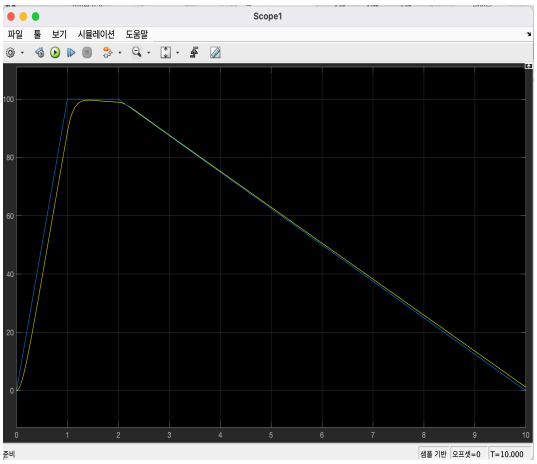
$$t_1 = 1.0,\ t_2 = 2.0\ (t_2 - t_1 = 1)$$

다음과 같은 값들로 Signal Builder를 통해 아래와 같이 singal을 만들었다.





위에서 제작한 signal을 input으로 설정한다. 위의 block diagram을 통해 얻은 그래프는 다음과 같다.



그래프를 보면 Overshoot는 발생하지 않고 일정한 Steady state error를 가진 채로 Reference를 따라가는 것을 확인할 수 있다. Lag Compensator를 적용함으로써 Transient response 특성을 유지하면서 Steady state error를 개선할 수 있다.

5. You need to submit an additional document (not technical) related this project and this will be announced later.

〈 결과 및 고찰 〉

Term Project를 진행하며 제어기를 직접 설계하였다. PID, Lead Compensator, Lag Compensator 세 가지의 제어기를 다루며 다양한 disturbance 형태에 대한 제어기에 대해서 각 제어기의 성능을 확인해볼 수 있었다. 그 과정속에서, P I D의 Gain들을 직접 구해보고, Lead compensator와 비교하면서 차이점을 알 수 있었다. 또한 system parameter를 증가하거나 감소하여 변경 함으로써 외란에 대해서 어떻게 반응을 하는지도 볼 수가 있었다.

Bode plot과 Nyquist plot을 통해 Phase margin을 찾아내고 Lag Compensator를 설계하였다. 여러 가지 값을 대입해보며 Kp값에 따른 제어기의 성능을 확인해 보았다. 조건을 만족하는 내에서 Kp가 클수록 Phase margin이 적었다. 하지만 Reference에 수렴하는 속도가 너무 빠르기 때문에 모터에 적용할 수 없는 값이었다.

〈 참고문헌 〉

- 자동제어 수업 강의자료 참고

- 네이버블로그(양반김의 잡동사니), “[제어공학, 매트랩] 시뮬링크를 이용한 간단한 시스템의 블록도 구현 및 반응과 시간에 대한 변화 관찰1”, https://m.blog.naver.com/may980911/221785075984

- 티스토리(shlinear), “MATLAB SISOTOOL을 이용한 PID 제어기 설계”, https://slowlee.tistory.com/3