

제어기 설계

팀프로젝트 발표
- 모터 회전각 제어 -

2014146042 주성민

01 초기 계획 모델

매트랩을 사용한 제어기 설계

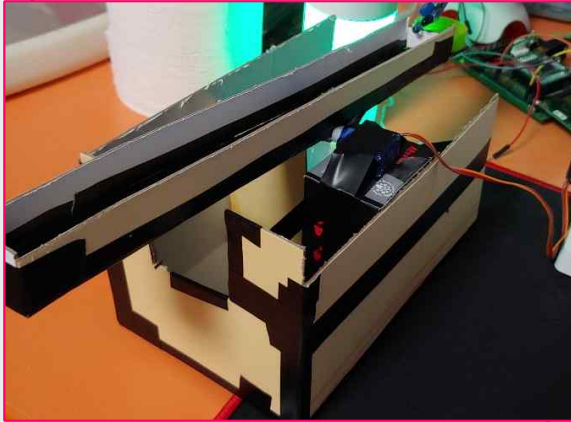
02

03 주요코드 설명

최종 결과 및 평가

04

초기 계획 모델



모델 설명

- 서보모터의 각을 제어하여 공의 위치를 제어

모델에 사용한 코드

```
distance = (short)ADCW;|
err = target - distance;
Kp_err = Kp*err;
if(Kp_err < 0)
{
    if(err > -3)
    {
        OCR2 = 30;
    }
    else if(err > -7)
    {
        OCR2 = 31;
    }
    else if(err > -13)
    {
        OCR2 = 32;
    }
    else
    {
        OCR2 = 33;
    }
}
```

- 간단한 비례제어기로 먼저 설계하여 동작을 확인 후 더 정밀한 제어기를 설계하려고 계획.

- 공과 거리를 측정하여 오차를 구한 후, 오차에 비례상수를 곱하여 서보모터를 제어하는 알고리즘.

- 초기에 5번의 거리측정을 한 후 중앙 값을 받아오는 Median Filter를 사용하였으나 모델의 반응속도가 느려서 정밀한 제어가 불가능.

- ADC의 최소 분주비(2분주)를 사용하여 사용하였지만 정밀 제어 실패.

매트랩을 사용한 제어기 설계

사용 모델

- DC모터의 회전(각) 제어 모델.

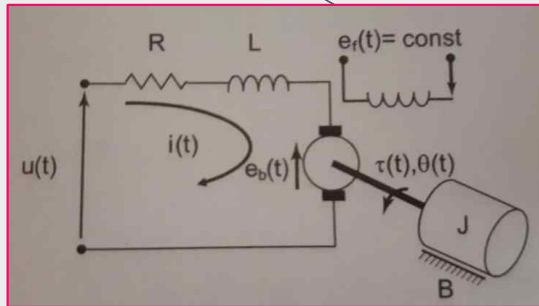
모델 사용 이유

- 앞서 시도했던 모델과 비슷한 모델을 선택.

설계 계획

- 개인적으로 부족했던 P, PI, PD, PID 제어기를 설계해본 후 비교.
- 제어가능성을 판단 후 상태 피드백 제어기 설계.

- 매트랩에 사용한 모델 설명



$$\dot{x}(t) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\frac{B}{J} & \frac{K_t}{J} \\ 0 & -\frac{K_e}{L} & -\frac{R}{L} \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L} \end{bmatrix} u(t)$$

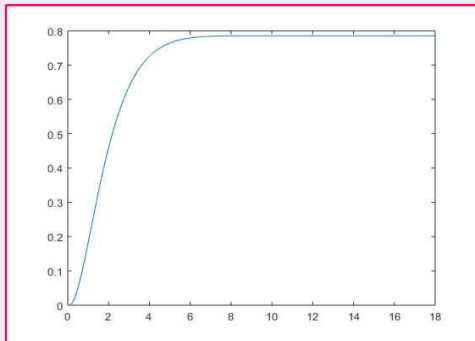
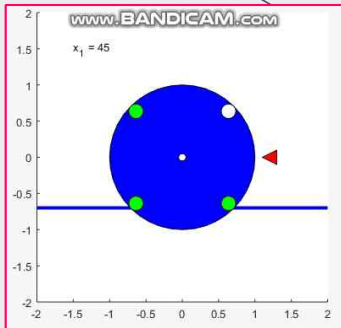
$$y(t) = [1 \ 0 \ 0] x(t)$$

$$x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix}$$

$$x_1(t) = \theta, \quad x_2(t) = \dot{\theta}, \quad x_3(t) = i(t)$$

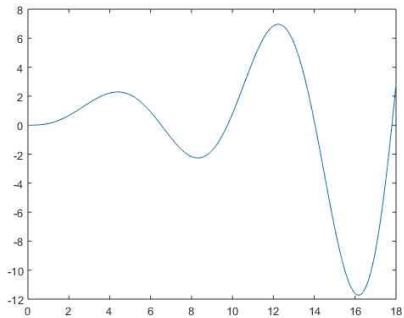
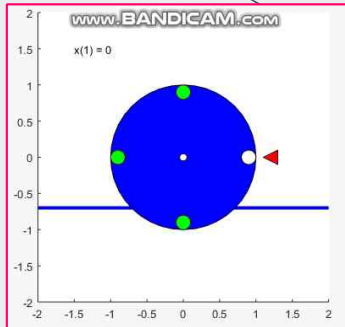
* **B** = 점성마찰계수, **K_t** = 모터 토크 상수, **K_e** 역기전력 상수

최종 결과 및 평가 - P제어기



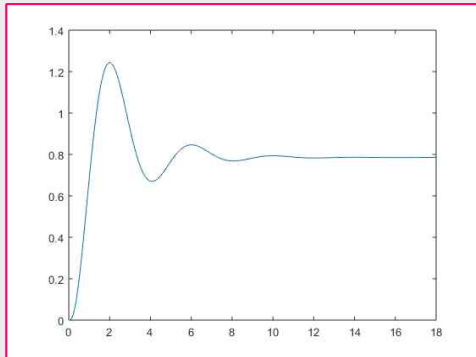
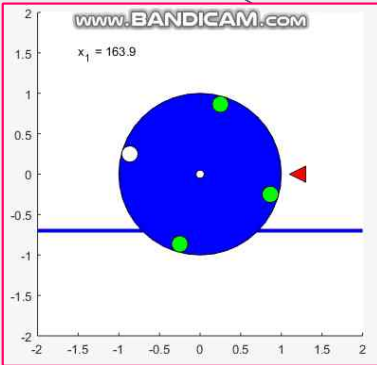
- 적절한 비례상수를 선택하여 제어한 결과.
- 목표값에 도달할수록 힘이 약해진다.
- 너무 작은값을 선택하면 잔류편차가 생기게 된다.

최종 결과 및 평가 - PI제어기($K_p = 0$)



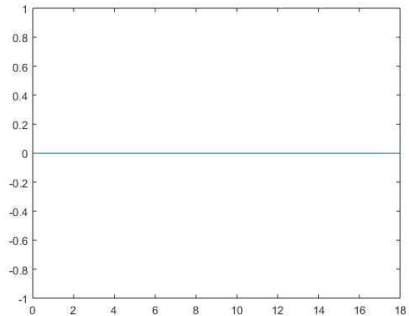
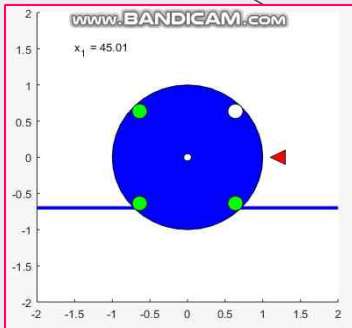
- 비례제어가 없다면 진동, 발산하는 결과가 나옴.
- 누적 적분 성분때문에 변화량이 늘어나는것을 볼 수 있음.

최종 결과 및 평가 - PI제어기



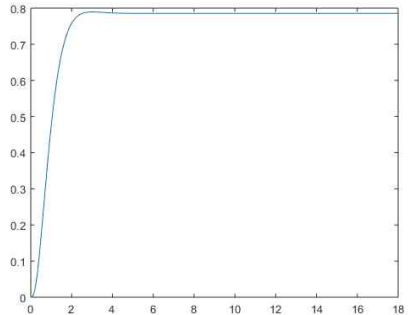
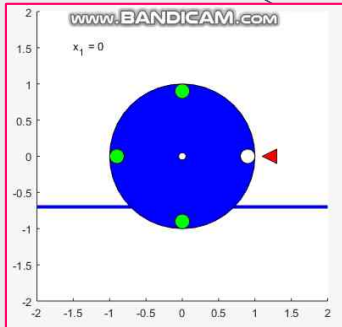
- 비례상수를 적당히 설정하여 PI제어를 한 결과.
- 트래킹 성능이 좋지 않지만, 목표치 부근에서 토크를 생성해주는 장점.
- 잔류편차를 제거해주는 역할을 한다.

최종 결과 및 평가 - PD제어기($K_p = 0$)



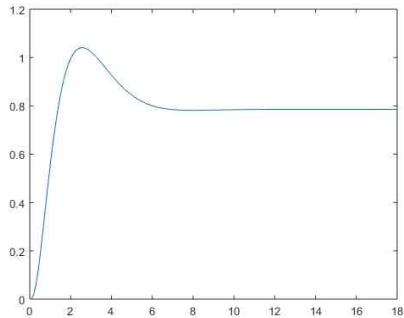
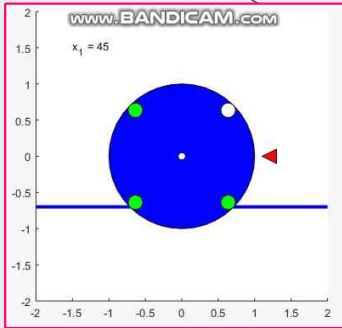
- 비례상수가 0이면 비례제어기의 오차의 변동이 없기때문에 아무 변화가 없음.

최종 결과 및 평가 - PD제어기



- 비례상수를 적당히 설정하여 PD제어를 한 결과.
- 트래킹성능이 상당히 좋아진 결과를 볼 수 있음.

최종 결과 및 평가 - PID제어기



- P,I,D(비례, 적분, 미분) 각각의 장점들을 합친 PID제어의 결과.
- $K_p = 18$, $K_i = 8$, $K_d = 7$ 로 설정.

최종 결과 및 평가 - 상태 피드백 제어기(제어가능성 확인)

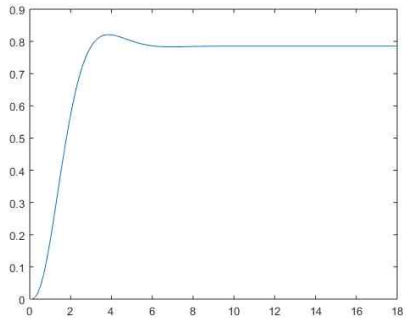
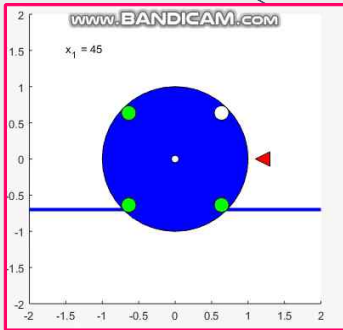
```
A = [0, 1, 0; 0, -B/J, Kt/J; 0, -Kb/L, -R/L];  
B_1 = [0; 0; 1/L];  
C = [1 0 0];  
P = [-2, -1+j, -1-j]; %%예제와 같은 성능을 구현  
K = acker(A,B_1,P);  
N = 1/(C*inv(-A+B_1*K)*B_1);  
ITT = rank([B_1,A*B_1,(A*A)*B_1])
```

ITT =

3

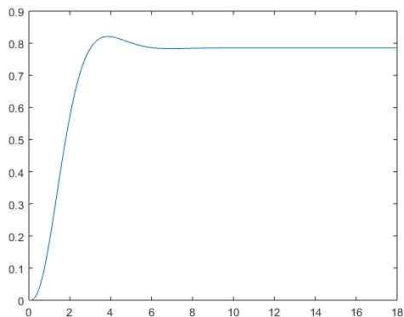
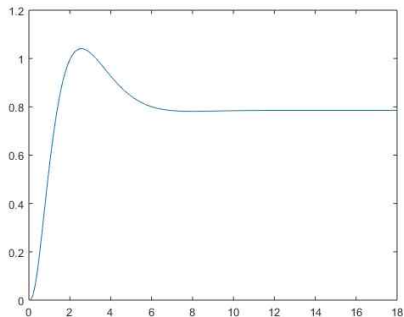
- 제어가능성을 확인하는 과정.
- $N = 3$ 으로 만족.

최종 결과 및 평가 - 상태 피드백 제어기



- 적절한 극점을 선택 후, Ackerman 공식을 사용하여 제어.

최종 결과 및 평가 - 두 제어기 비교



- 좌측은 PID, 우측은 상태피드백 제어기.
- 매번 오차를 계산해야하는 PID제어기와 다르게 상태피드백 제어기는 Ackerman 공식을 이용해 한 번 계산.
- 과도응답 특성은 차이가 있지만, 정상상태 특성은 비슷하다고 볼 수 있음.



감사합니다

2014146042 주성민

