|  |  |
| --- | --- |
| 년도-학기 | 2021년 2학기 |
| 과목명 | 자동화프로그래밍 |

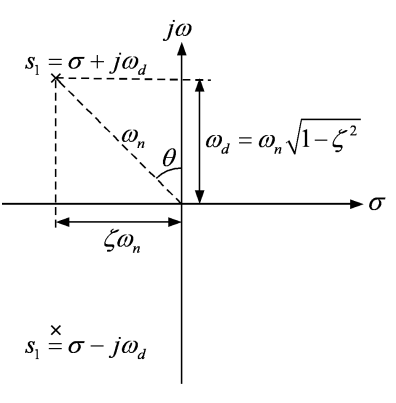
|  |  |
| --- | --- |
| **번호** | **실험 제목** |
| LAB 3 | Digital PID Controller |

|  |  |
| --- | --- |
| 실험 일자 | 2021년 9월 23일 |
| 제출자 이름 | 강\*\* |
| 제출자 학번 | 201803\*\*\*\* |
| 팀원 이름 |  |
|  |  |

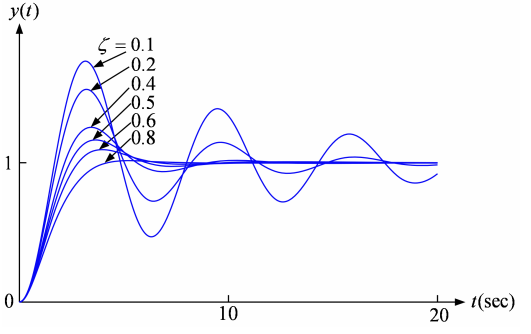
**Chapter 1. 관련 이론(Theoretical Background)**

2차 시스템 전달함수의 표준형은 다음과 같다. 이 때 분모를 0으로 만드는 s 값은 pole이 되며 pole의 위치에 따라 시스템의 응답 특성이 달라진다.

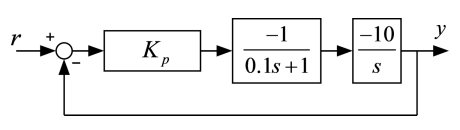
테이블이(가) 표시된 사진

자동 생성된 설명 

ζ는 감쇠비, damping ratio로 그 크기가 클수록 overshoot이 작아진다.

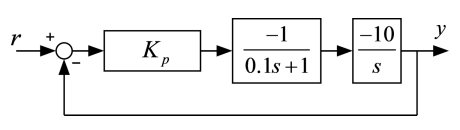


- P 제어기(비례 제어기)



출력과 입력의 차인 에러에 Kp를 곱한 값에 비례하는 제어기이다. Kp 값이 커질수록 급격하게 출력이 변화한다. 정상상태임에도 항상 오차가 존재하여 목표 값에 도달할 수 없는 단점을 가진다.

- I 제어기(적분 제어기)



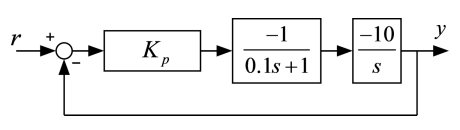
***KI /s***

적분제어기는 에러를 누적하는데 출력 값이 목표 값보다 높으면 +, 낮으면 - 가 되므로 정상상태오차를 줄일 수 있다.

- PI 제어기

위의 특징으로 비례 제어와 적분 제어를 함께 사용하면 비례 제어기의 정상상태오차를 줄일 수 있게 된다.

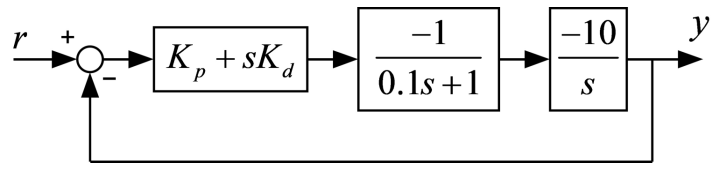
- D 제어기(미분 제어기)



***Kds***

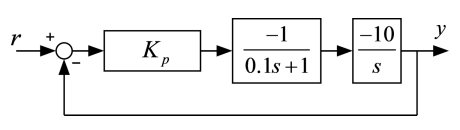
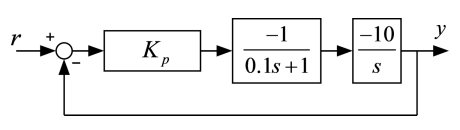
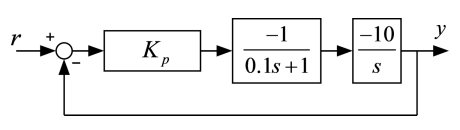
Kds가 추가되어 상태방정식을 구했을 때 0.1s2+(1+10Kd)s+10으로 damping ratio가 증가하게 된다. 따라서 Kd 값이 커질수록 overshoot을 줄이는 정도가 커진다.

- PD 제어기



비례 제어만을 사용할 때의 overshoot을 줄이고 급격한 변동을 제어한다. 이전 결과와의 편차를 파악해 출력 값을 조절하여 제어한다. 이에 따라 안정성이 향상된다.

- PID



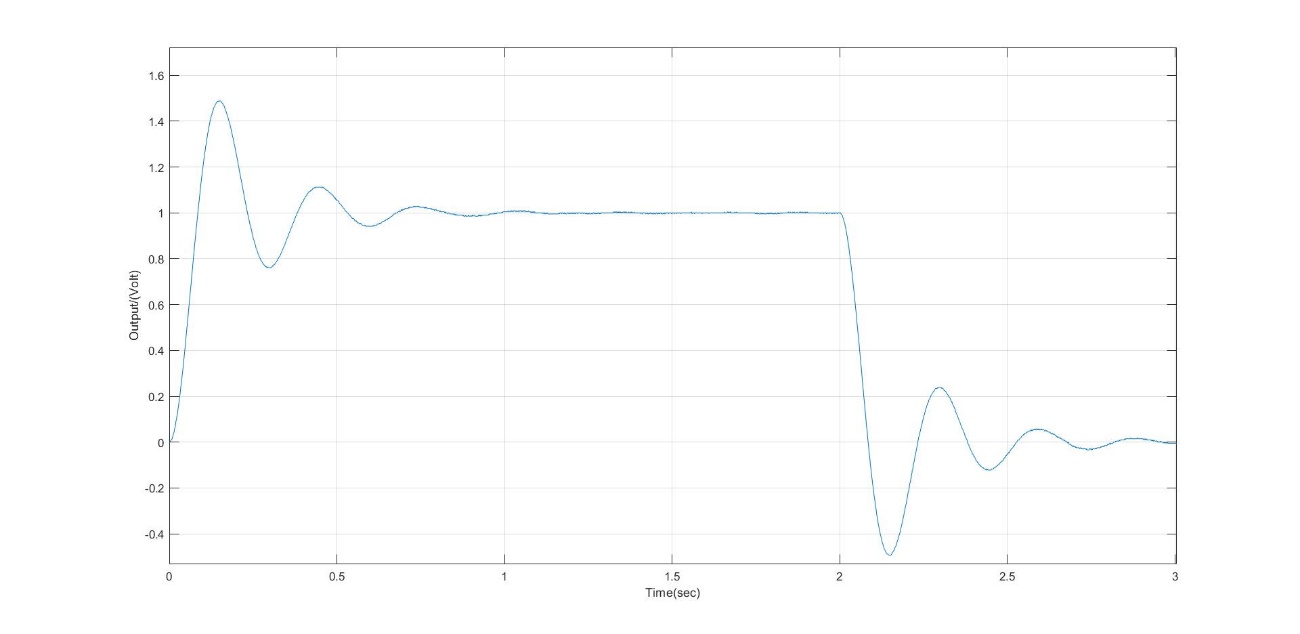
***Kp +Kds +Ki/s***

비례, 적분, 미분 제어를 모두 사용하는 PID 제어기는 에러의 크기, 누적된 양, 변화량을 고려하여 제어하므로 반응속도가 빠르고, 안정성도 좋으며, overshoot도 감소한다.

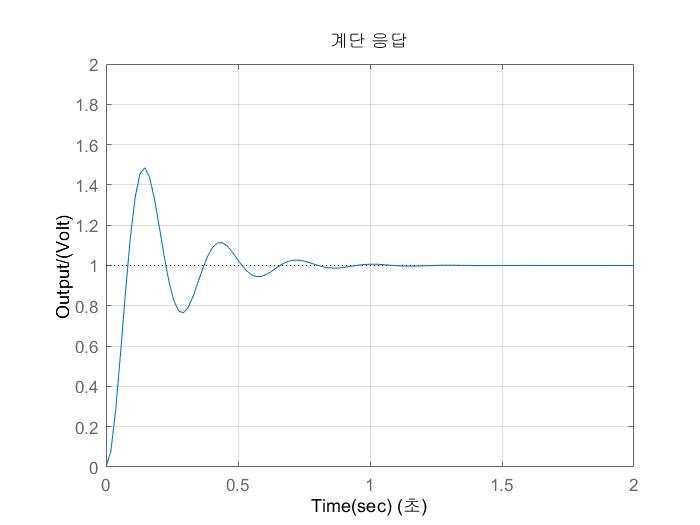
**Chapter 2. 실험 결과(Experimental Results)**

[Exercise 1 : PD-Controller]

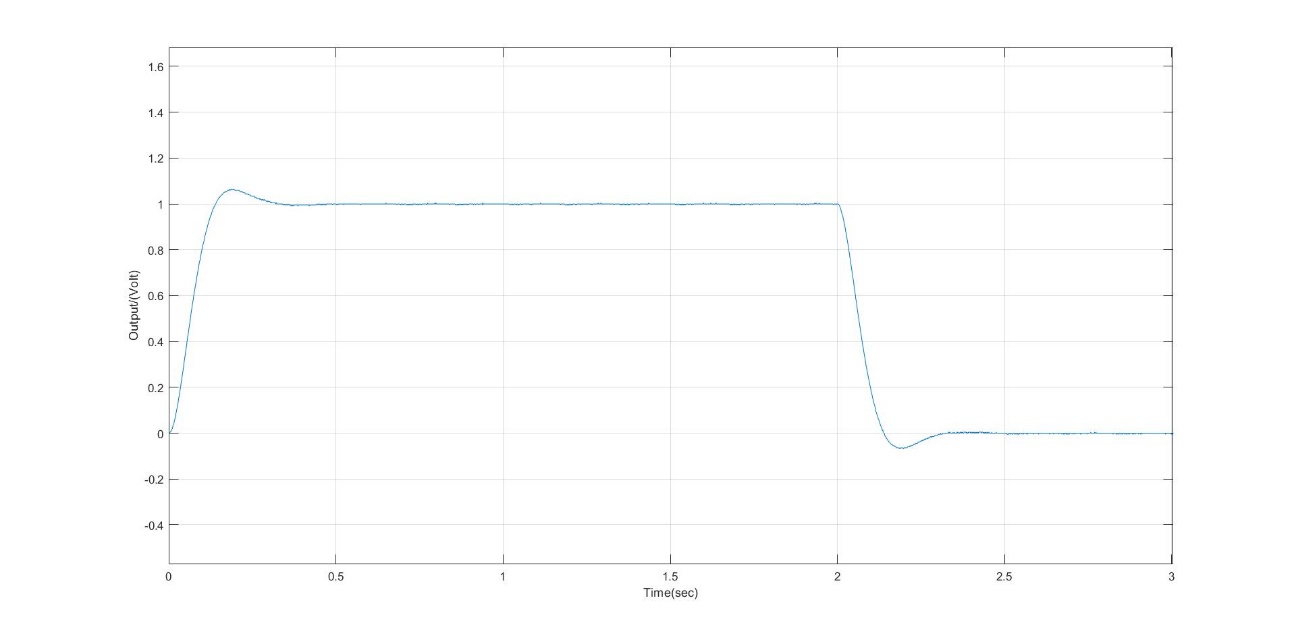
아래 사진은 Kp=5, Kd=0.0 일 때의 실행 결과를 plot한 결과이다. Kd = 0으로 P 제어기로 동작한다.



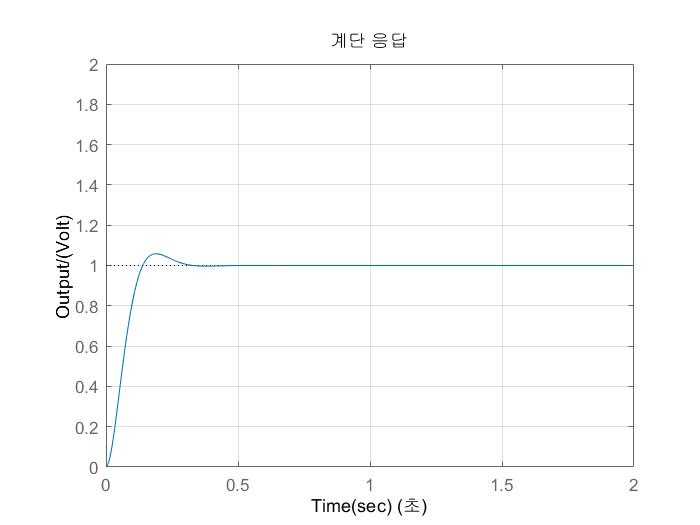
아래의 Matlab Simulation 결과와 비교하면 같음을 알 수 있다.



다음은 Kp=5,Kd=0.2 일 때 실행한 데이터들로 plot한 결과이다. Kd = 0.2로 PD 제어기로 동작한다. 이전 P제어기에 비해 overshoot이 줄고 목표 값에 더 빨리 도달하였다.



아래의 Matlab Simulation 결과와 비교하면 같음을 알 수 있다.



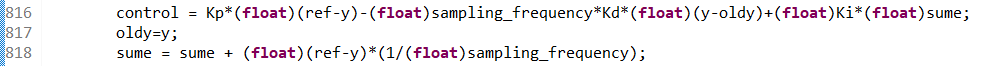
[Exercise 2: PID-Controller]

텍스트이(가) 표시된 사진

자동 생성된 설명 텍스트이(가) 표시된 사진

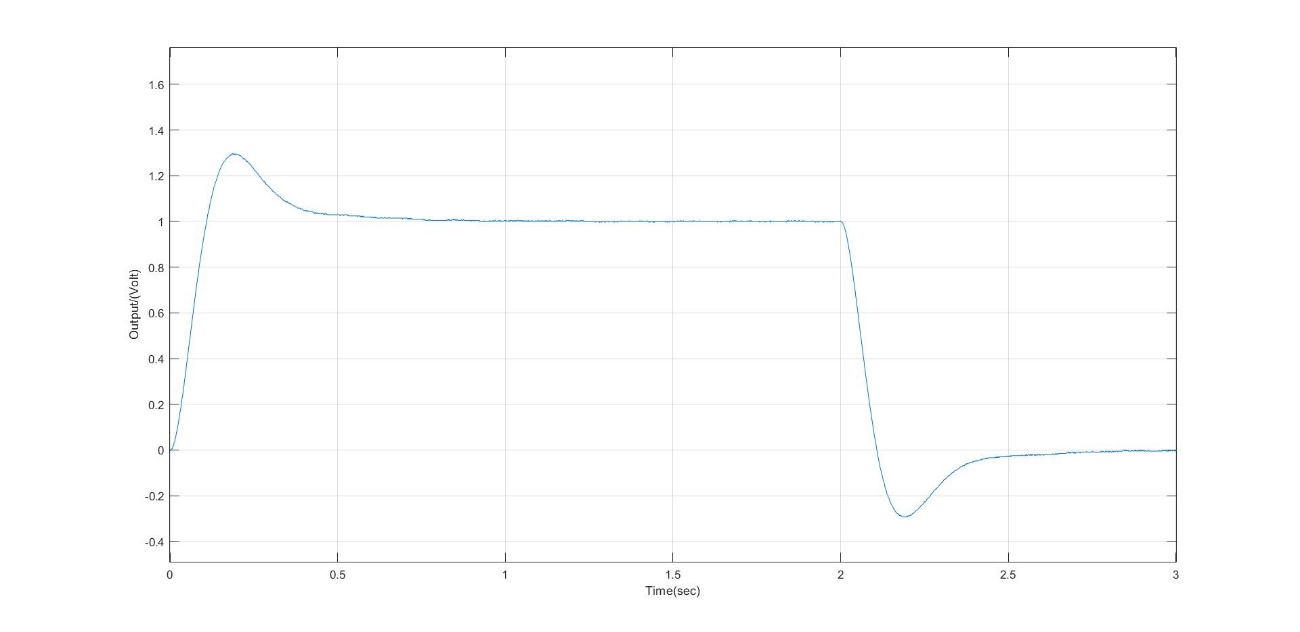
자동 생성된 설명

PID 제어기 설계를 위해 Ki=20.0 변수와 integrator에서 e(t)값을 누적해 더해줄 sume 변수를 추가하였다.

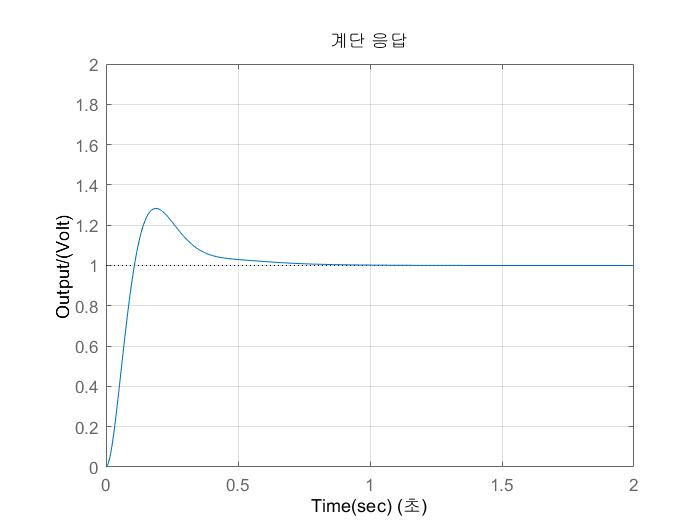


Control 식에 적분기에 대한 식을 추가하였다.

아래는 실행해서 저장한 데이터를 이용해 plot한 결과이다. PD 제어기에 비해 overshoot이 증가하고 속도도 느려짐을 확인할 수 있다.



아래의 Matlab Simulation 결과와 비교하면 같음을 알 수 있다.

>

**Chapter 3. 결론 및 Discussion**

실험결과를 통해 제어시스템공학에서 배웠던 비례, 적분, 미분 제어기의 특성을 확인할 수 있었다. 단순히 이론적, 수학적으로만 이해했던 것들을 눈으로 확인할 수 있어 이해에 많은 도움이 되었다. 비례 제어기를 단독으로 사용하는 것보다 미분, 적분 제어기를 함께 사용했을 때 overshoot과 정상상태 오차 측면에서 좋은 성능을 보여주는 것을 확인할 수 있었다. 또한, PID 제어기를 사용하면 PD 제어기를 사용할 때보다 overshoot이 증가하는데 왜 사용하는지에 대해서도 궁금하였는데, ramp 입력에서의 정상상태 오차를 줄일 수 있기 때문임을 수업 시간에 교수님의 설명을 통해 알 수 있었다. 제어공학에서 시스템 type을 배웠음에도 이유를 모르고 지나갔는데 이번 실습을 통해 type을 배웠던 이유를 알게 되었다.

sampling\_frequency가 정수형으로 선언되어 있어서 적분기에 대한 식을 수정할 때 (float)를 붙여주어야 했다. 처음엔 직접 0.001을 넣어 실행해보아서 문제를 인지하지 못했는데 1/sampling\_frequency로 바꾸자 결과가 제대로 나오지 않았다. float형으로 바꿔줘야 함을 알게된 뒤에도 (float)(1/sampling\_frequency)처럼 밖에 붙여주었더니 결과가 제대로 나오지 않았다. 당황스러웠지만 (float)를 샘플링주파수 바로 앞에 붙이니 정상적인 결과가 나오는 것을 확인했다. 당시에는 무엇이 문제인지 알 수가 없어서 적잖이 당황했었다.

**Appendix:**

과제 제출 시 별도로 소스코드 파일 첨부하였습니다.