**고급소프트웨어실습I HW9 보고서**

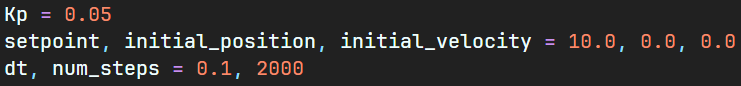
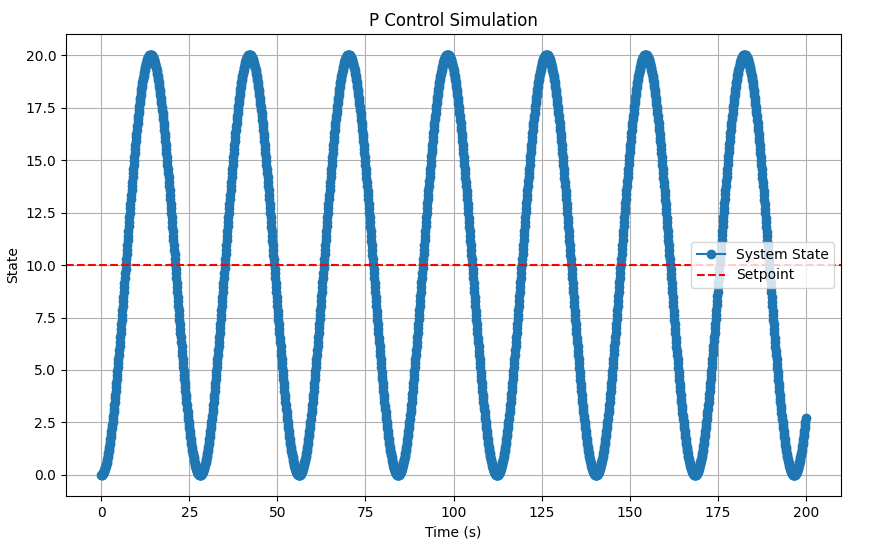
**20211584 장준영**

**0. 코드 실행**

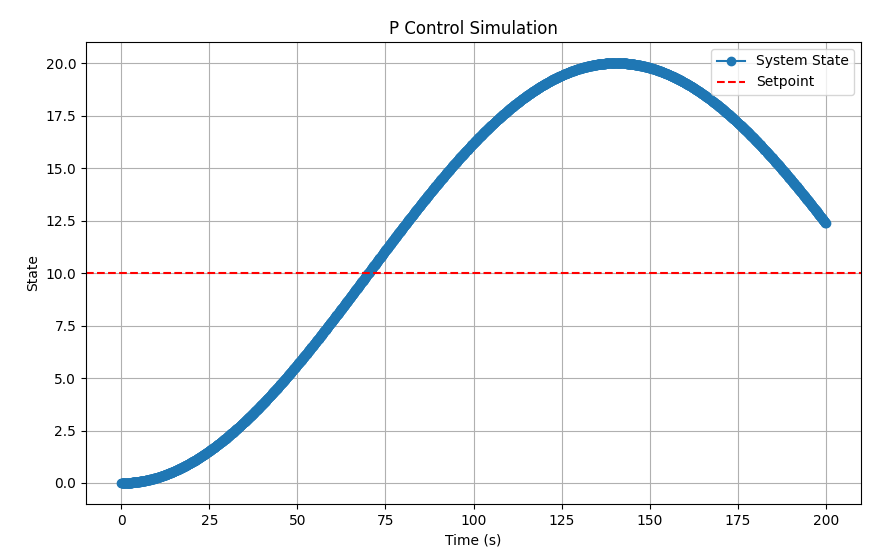
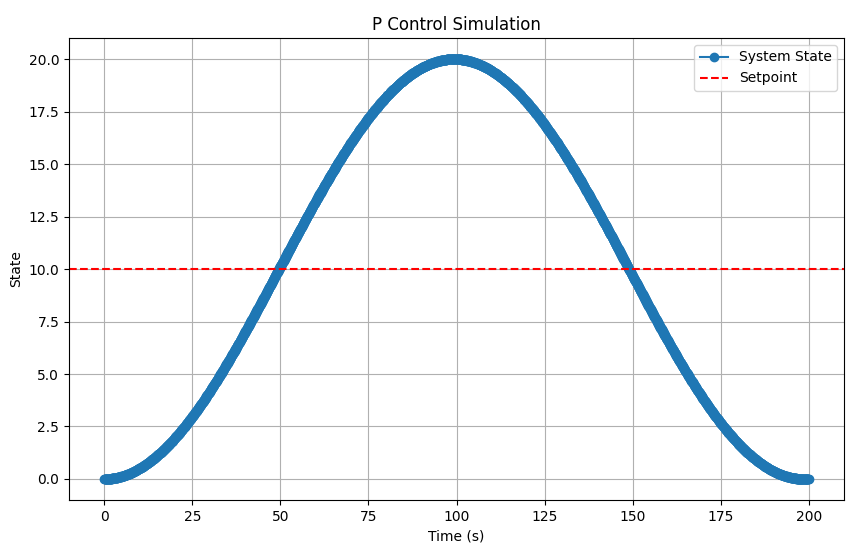
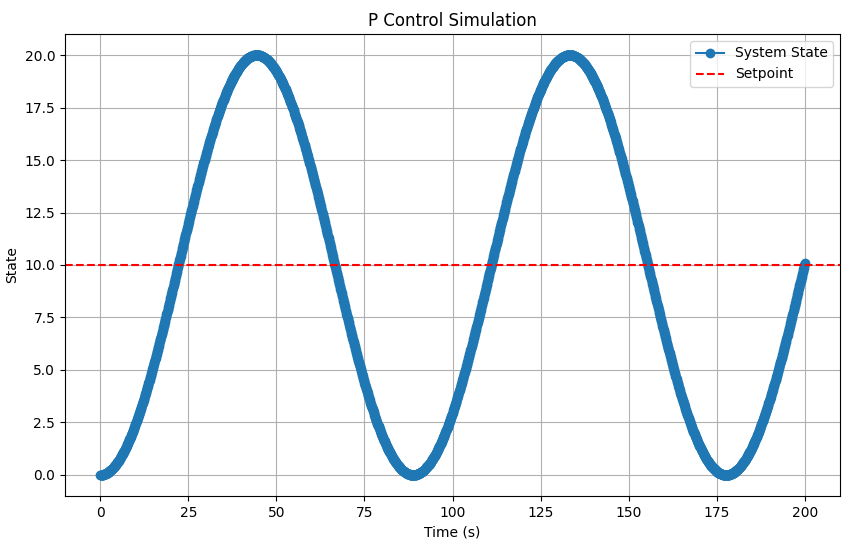
나의 코드는 ‘python <code file name> --<parameter : p/pd/bb>’의 명령어로 실행하면 세 가지 제어 방식에 대한 그래프를 확인할 수 있다. 코드는 필요한 parameter를 받아 각각의 동작을 실행하고 계산한 값을 그래프의 점으로 반환한다.

**1. P Control**

P control은 배운 피드백 제어 방식 중 가장 간단한 형태로, 목표값과 실체값의 오차(error)에만 조정값을 비례시킨다. 응답이 빠르지 않고, 진동이 계속되기 때문에 오차와 응답 속도가 중요하지 않고 단순한 계산을 해야 하는 상황에 사용한다.



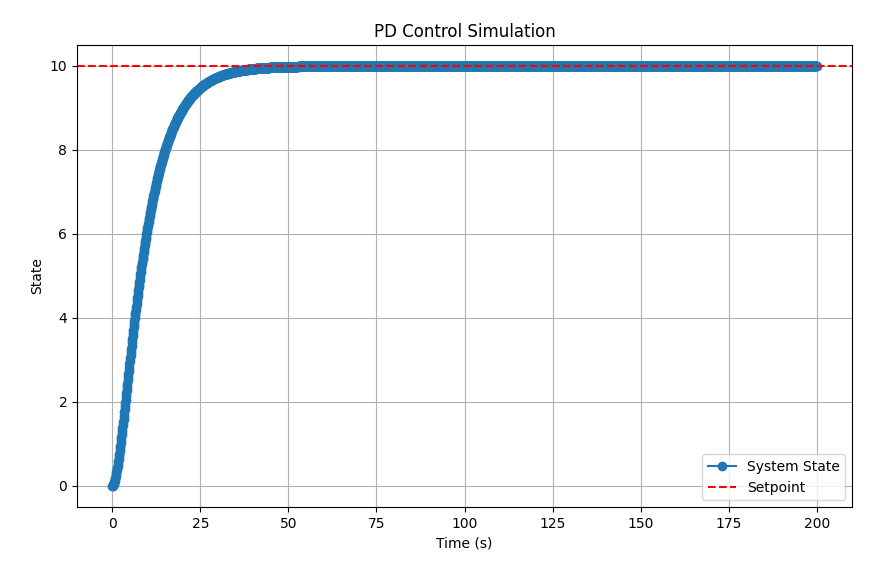
아래 이미지와 같은 파라미터의 값을 갖는 p control의 그래프이다. 목표값과 실제값의 오차가 커질 수록 조종값이 커져 계속 목표값 근처를 진동하는 것을 볼 수 있다.

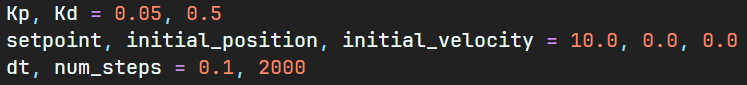


제어 계수인 Kp를 줄여감에 따라, 값을 조정하는 양이 줄어들어 폭이 큰 진동 형태의 그래프가 나오는 것을 볼 수 있다.

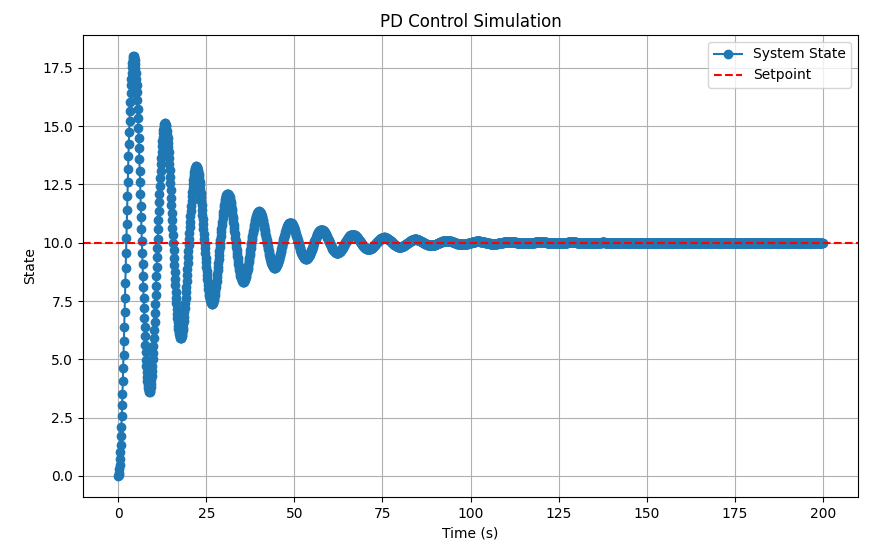
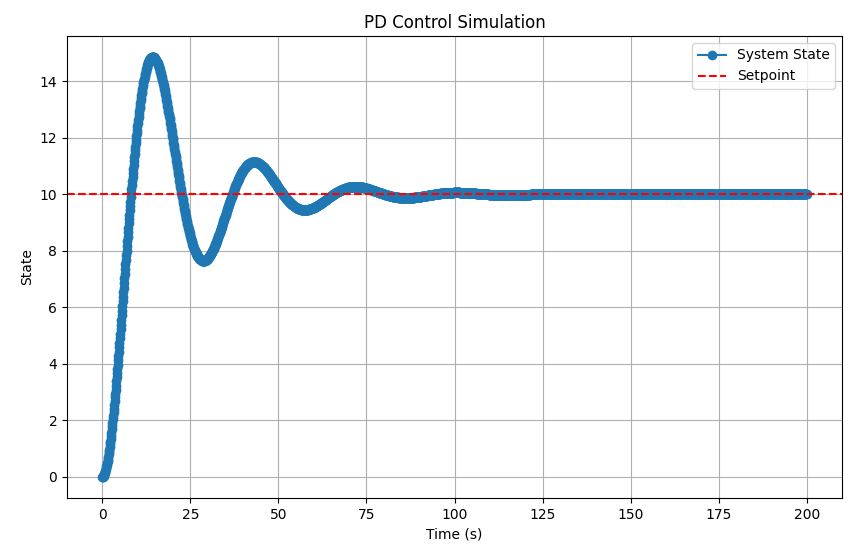
**2. PD Control**

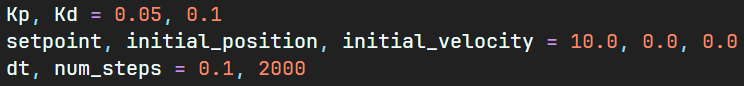
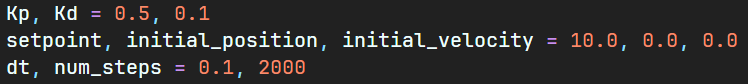
PD control은 P control에 D(derivative) 요소를 더한 것으로, 오차의 변화율(미분)에도 반응하여 조정값을 정하는 방식이다. 오차가 static 하다면 그 점으로 수렴하고, 오차가 변동해도 적절히 따라가기 때문에, D 요소를 통해 계속된 진동을 없애고 목표값에 정밀하게 수렴할 수 있다.





아래 이미지와 같은 파라미터의 값을 갖는 pd control의 그래프이다. Kd가 0.5로 큰 값을 갖기 때문에, 오차의 변화율에 대한 강한 의존도를 갖게 되어 진동이 매우 약해지거나 사라진다.

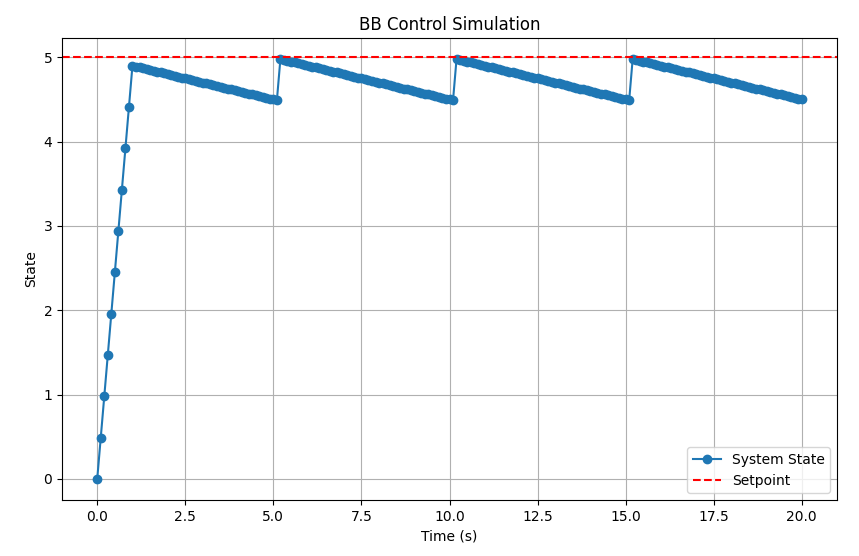


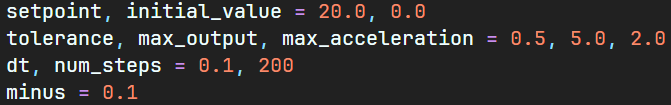
 

이번에는 Kd의 값을 줄이고 Kp의 값을 늘리는 두 경우의 파라미터를 넣어보았다. 여전히 오차의 변화율과 상관관계가 있지만 그 값이 작아져 관여하는 양이 줄어든다. 따라서, 목표값 근처에서의 진동 후 수렴을 확인할 수 있다.

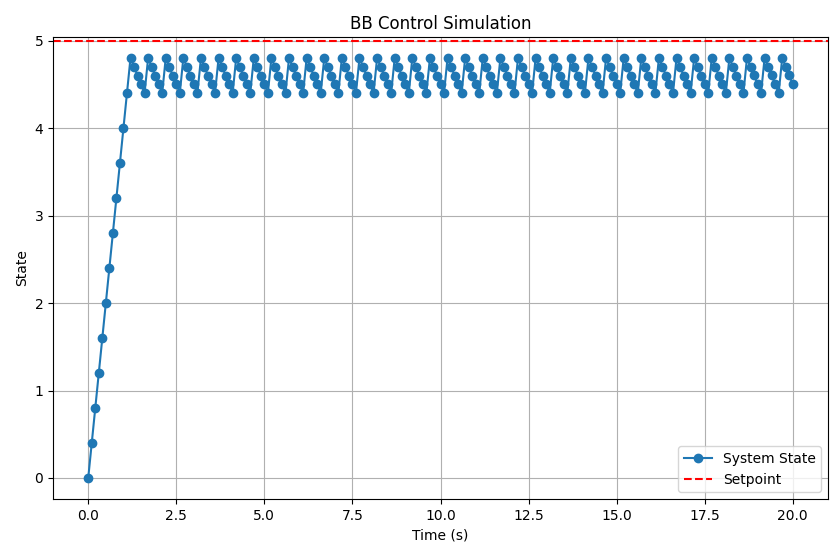
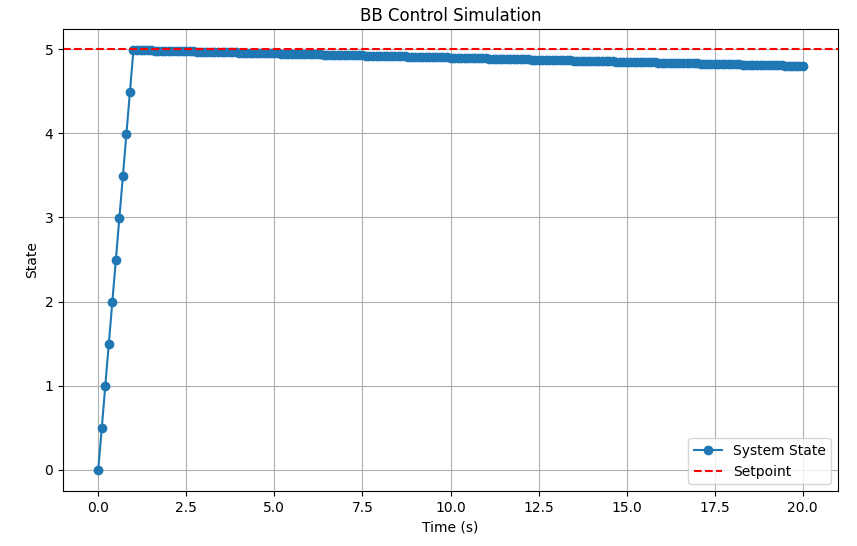
**3. Bang-bang Control**

Bang-bang control은 On/Off 두 상태를 계속하여 전이하면서 목표값을 유지하는 방식이다. 구현이 쉽고 빠르게 목표값에 도달할 수 있지만, 상태 전이가 짧은 주기로 반복되면서 시스템에 큰 스트레스를 줄 수 있으며 정밀한 제어에는 알맞지 않다.





아래 이미지와 같은 파라미터 값을 갖는 bb control의 그래프이다. On/Off를 조절하는 방법이 조금 모호할 수 있는데, 강의자료에서 설명한 방식대로 특정 값을 빼는 방법으로 조절하였다(minus가 그 양을 나타내는 파라미터). 목표값까지 계속해서 증가하다가(On state), 목표값을 초과할 수 있는 때가 되면 값을 줄임으로써(Off state) 목표값에 빠르게 도달해 근처에서 머물 수 있게 해준다.



좌측 사진은 minus 값을 줄인 그래프이고, 우측은 늘린 그래프이다. 임계값 근처에서 줄어드는 양에 따라 On/Off state 전환의 빈도가 달라진다. 만약 우측 같은 상황이 된다면, 전환의 빈도가 너무 높아 시스템에 무리를 줄 수 있으므로 bb control을 사용할 땐 목표값이 어떻게 변하는지 분석하여 파라미터를 설정해야 한다.