|  |
| --- |
| **제어공학[7-1] PID제어기 요약**  [제어공학](https://blog.naver.com/PostList.naver?blogId=kckoh2309&categoryNo=56&from=postList) / [IT강좌](https://blog.naver.com/PostList.naver?blogId=kckoh2309&categoryNo=28&parentCategoryNo=28&from=postList)   2020. 10. 31. 17:03 |

**비례-적분-미분 제어기(Proportional-Integral-Differential controller)**

**PID 제어(PID control)**는 실제 산업 응용분야에서 가장 많이 사용되는 대표적인 형태의 제어기이다. PID 제어기는 기본적으로 [피드백(feedback)](https://ko.wikipedia.org/wiki/%ED%94%BC%EB%93%9C%EB%B0%B1)제어의 구조를 가지고 있으며, 제어하고자 하는 대상 플랜트의 출력(output)을 측정하여 이를 기준입력(reference input) 혹은 원하고자 하는 입력(Desired input)과 비교하여 오차(error)를 계산하고, 이 오차값을 기반으로 하여 제어에 필요한 제어입력을 출력한다.

표준적인 형태의 PID 제어기는 아래의 식과 같이 세개의 항을 더하여 제어값(MV : Manipulated Variable)을 계산하도록 구성이 되어 있다.



이 항들은 각각 오차의 비례(proportional), 오차의 적분(integral), 오차의 미분(derivative)에 비례하기 때문에 비례-적분-미분 제어기 (Proportional–Integral–Derivative controller)라는 이름을 가진다.

또한 각 항들을 따로 분리하여, 비례제어(P-control), 적분제어(I-control), 미분제어(D-control)이라고 부른다. 각 제어의 효과는 다음과 같다.

* **비례제어** : 간단하게 설명하자면 목표와 지금 상태의 차이가 나는 만큼 제어입력을 넣는 것이다. 현재 상태에서의 오차를 줄여주고, 응답성을 개선하는 효과를 가진다. 현재를 중요시여긴다. 그러나 시스템 상대안정성은 나빠진다. 결정적인 것인 정적인 외부 부하가 걸리게 되면, 정상상태 오차가 존재한다. 왜냐하면, 그 부하토크에 평형을 이루는 제어입력은 오차에 비례하기때문에 결국 오차가 있어야 그 부하토크를 견딜 수 있는 것이기 때문이다. (Spring역할)
* **적분제어** : 비례제어가 현재만을 고려한다면, 반면 적분제어는 과거를 중시한다. 어떤 사람이 큰 실수를 했다고 치자. 그러나 과거의 공이 크면 한번 용서가 되는 것과 유사하다. 오차의 과거 누적을 기반으로 하기 때문에 경사입력 또는 외란에 대해 정상상태(steady-state) 오차를 없애는 작용을 한다. 역시 반면 시스템 안정성은 나빠진다.
* **미분제어**: 미분제어는 현재의 오차의 절대값보다는 증가율을 따지기 때문에, 미래 즉 추세를 반영한다. 예를 들면 자동차의 경우, 목표위치에 도달하기 전이라도, 남은 거리에 비해 속도가 너무 높으면 속도를 줄여주어야 한다. 만약 비례제어 였다면, 오히려 속도를 높였을 것이다. 출력값의 급격한 변화에 제동을 걸어 [오버슛(overshoot)](https://ko.wikipedia.org/wiki/%EC%98%A4%EB%B2%84%EC%8A%88%ED%8A%B8_(%EC%8B%A0%ED%98%B8))을 줄이고 안정성(stability)을 향상시킨다. 그러나 기본적으로 미분기를 사용하기 잡음에 민감한 단점을 지녀, 단독적으로 사용을 못하고 1차 노이즈 필터와 함께 사용한다.  재생 수138

PID 제어기는 경우에 따라서는 약간 변형된 형태로 사용하는 경우도 많다. 예를 들어, 비례항만을 가지거나(P제어), 혹은 비례-적분항만을 가지는 제어(PI제어), 비례-미분항만을 가진 제어(PD제어)의 형태로 단순화하여 사용하기도 한다.

**안티 와인드업(Anti-windup)**

계산된 제어값이 실제 구동기(actuator)가 작용할 수 있는 값의 한계보다 커서 구동기의 포화(saturation)가 발생하게 되는 경우, 오차의 적분값이 큰 값으로 누적되게 되어서, 정작 출력값이 설정값에 가까워지게 되었을 때, 제어값이 작아져야 함에도 불구하고 계속 큰 값을 출력하게 되어 시스템이 설정값에 도달하는 데 오랜 시간이 걸리게 되는 경우가 있는데, 이를 [적분기의 와인드업](https://en.wikipedia.org/wiki/Integral_windup)이라고 한다. 이를 방지하기 위해서는 적절한 안티 와인드업(Anti-windup) 기법을 이용하여 PID 제어기를 보완해야 한다.

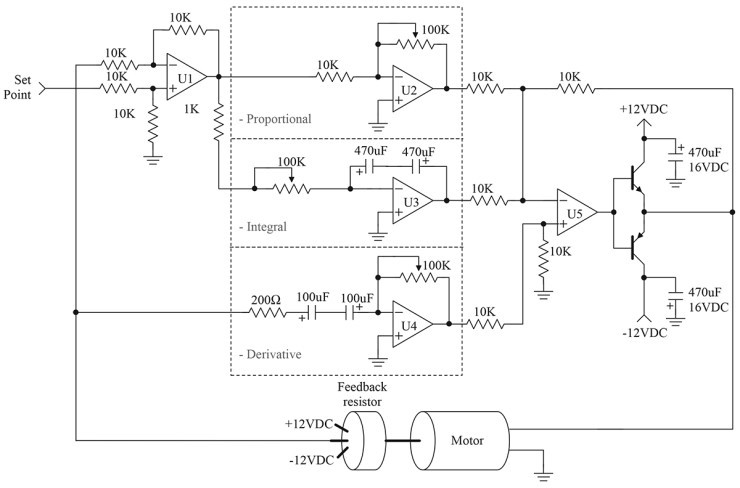
**PID 제어기의 자동 튜닝(Auto-tuning)**

제어 파라메터 를 이득값 혹은 게인(gain)이라고 하고, 적절한 이득 값을 수학적 혹은 실험적/경험적 방법을 통해 계산하는 과정을 튜닝(tuning)이라고 한다. PID 제어기의 튜닝에는 여러 가지 방법들이 있는데, 그중 가장 널리 알려진 것으로는 [지글러-니콜스 방법](https://en.wikipedia.org/wiki/Ziegler%E2%80%93Nichols_method)이 있다.

**모션제어기(Motion controller)**

Encoder, H-bridge, PWM amplifier, inner loop, outer loop, micro-controller, ADC(Analog-Digital Converter)

다음은 아날로그 Op-Amp로 구현한 PID제어기 회로도이다. [1]



[1] <https://www.nutsvolts.com/uploads/wygwam/NV_0105_Dahlen_Figure08.jpg>