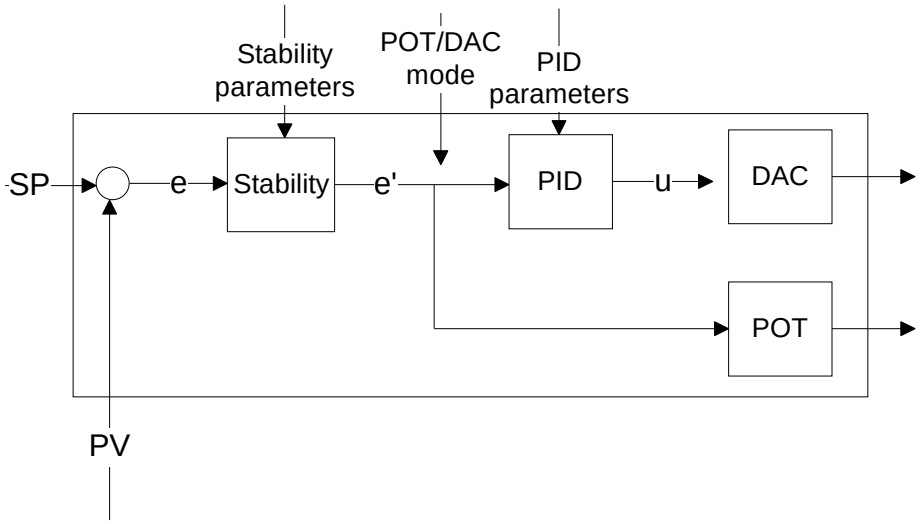
PID 제어에 대하여

작성 : 애니엘㈜ 2014-02-25

PID_제어원리와 설정방법 (1)



SP : 지정값 (input 또는 output)
PV : 프로세스에서 오는 feedback
e : [= SP – PV] 지정값과 feedback 사이의 편차
Stability : 다음 번에 PID 나 POT 로 가야하는 최대 input 을 계산함. ,
이 stability 에 의해 기울기가 정해져서 출력값의 ramp 로 나타남.
POT/DAC mode switch : 출력값이 analogy output (DAC)인지 접점출력인지 지정함.
PID : PID 제어 알고리즘
u : 다음 번 DAC output 을 발생하게 하는 input.
DAC : analogy output,
POT : digital increase/decrease output.

PID algorithm
P = Gain (비례)
I = Process delay (Integral) (적분)
D = Derivated (미분) Do not change it. Same as acc. pump on an old car carburettor.

I 설정방법
- 최초에는 I 값을 길게 주고 테스트를 시작한다.
- 발생하면 P 값을 줄이고 변화가 보이지 않으면 P 값을 올린다.
- 그렇게 하여 단 한번의 시도에도 바로 목표치에 바로 도달하도록 최적의 P 값을 찾는다.
- 길게 줬던 I 값을 점점 줄여서 overshoot 없을 때까지 최단 최적의 응답시간을 찾는다.

PID 제어에 대하여

작성 : 애니엘㈜ 2014-02-25

PID_제어원리와 설정방법 (2)

아래 글은 http://maxpulse.tistory.com/139 에서 퍼온 내용입니다.

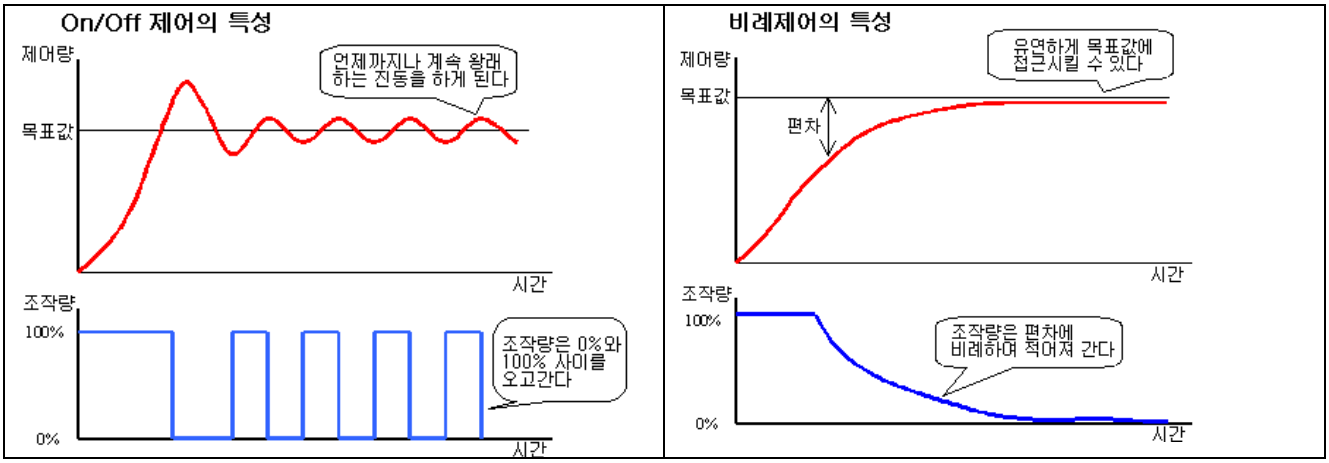
일단
P : 비례제어
I : 적분제어
D : 미분제어

여기서 비례, 적분, 미분 이란 것은
에러값(내가 원하는 제어 목표와 실제 제어 상태)에 대한 비례, 적분, 미분을 의미한다.

PID 제어는 모터의 속도/위치 제어, 보일러의 온도 제어등 여러가지 분야에서 쓰일수 있기 때문에
에러값이라하면 모터의 속도라거나, 보일러에서 끓는 물의 온도 등등이 된다.
그 에러값들은 제어 회로의 구성에 따라 전압의 아날로그 값이 되거나,
시간 간격에 따른 펄스의 개수, 혹은 펄스의 길이 이런 다양한 값이 될수 있다.

이런 에러값을 받아들여서 P, I, D 라는 방법을 활용하여 제어를 하게 된다.

1. P 제어 (비례 - Proportinal 제어)
에러값의 비례해서 제어량을 변화시키는 방법이다.



위 두개의 그림을 보면 On/Off 제어, 비례제어로 나뉜다.
하지만 On/Off 제어도 비례제어이다.

비례제어의 특성중 하나는 정상상태 오차(Steady State Error)가 없어지지 않는다는 것이다.

On/Off 제어를 비례제어인데 Gain(에러값에 대한 제어량에 기준값)이 무척 큰 것이라고 생각하면 일정 크기의 에러가 계속 존재하게 된다.
게인을 줄인다고 해도 그 에러는 계속 존재한다.
예를 들어 10 의 오차가 있을때 게인값을 적용하면 -1 의 오차가 남는다면(오버되는 경우) 다시 -1 의 오차를 보정하면 +0.1 의 오차가 남게되고 반복하게 된다.

다만 게인이 일정값 이하로 줄어들면 아래 그림처럼 약간의 편차가 계속 남은채로 목표값을 따라가게 된다.
예를 들어 속도의 오차가 10 이고, P 제어를 적용하면 9 의 오차를 줄일수 있다고 하자.
그러면 1 의 오차가 남는다. 그러나 다시 P 제어를 하면 0.9 의 오차를 줄이고 0.1 에서는 0.09.....0.01 의 오차에서 0.009 이런식으로 오차가 계속 남게 된다.

PID 제어에 대하여

작성 : 애니엘㈜ 2014-02-25

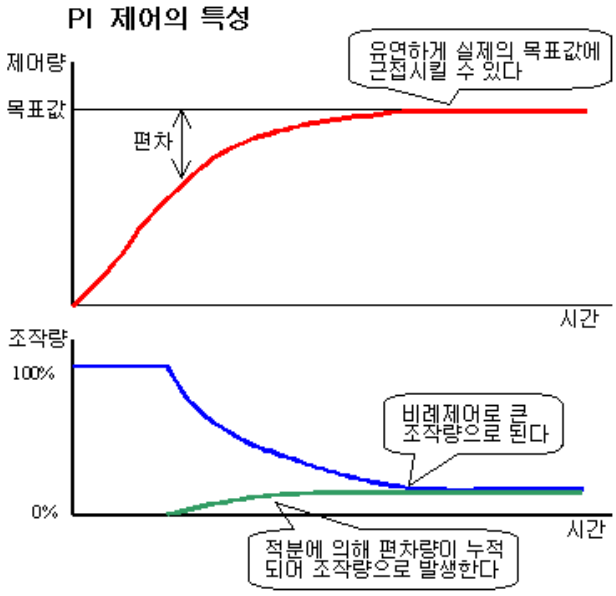
만약 게인값을 딱 맞추면 오차가 없어지지 않겠느냐라고 생각한다면
오차가 없어지는 순간 제어할수 있는 값이 없어지므로 다시 오차가 발생하게 된다.

* P 제어의 오차가 없어지지 않는 이유는 정확히 찾을수 없었으나 나의 생각은 이렇다는 이야기임..--;;

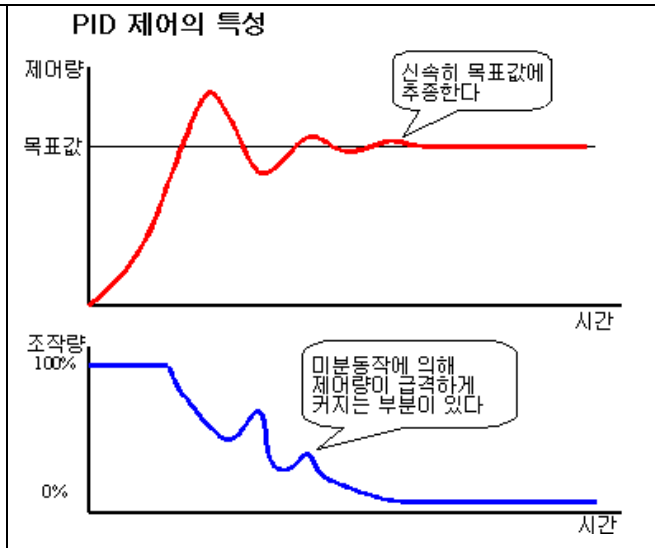
2. I **제어** (적분 Integral 제어)
에러값을 적분한 값을 가지고 제어를 하는 경우.

위에서 비례 제어의 특성 그래프를 보면 일정 크기의 정상상태 오차가 계속 남아있게 된다.
이는 P 제어로는 처리 할 수 없는 작은 오차(잔류편차)이므로 P 제어 만으로는 없앨수가 없다.

즉 이런 미소한 잔류편차를 시간단위로 적분하여 그 값이 어떤 크기가 되면 조작량을 증가시켜
편차를 없애는 방식이 I 제어이다.



위 그림을 보면 파란선은 P 제어에 의한 조작량이고,
녹색선이 I 제어에 의한 조작량이다.
그림을 보면 I 제어는 일정 시간동안 오차가 누적되어 일정값을
넘어서면 시작하게 된다.



위 그림은 PID 제어에 대한 그림이다. D 제어만 따로 놓고 보기는 좀
힘들기 때문에 PID 제어를 통틀어서 본다.
D 제어는 오차의 변화의 반대방향으로 제어가 이루어지게 된다.
즉 오차의 값을 미분한(즉 빨간선의 기울기)의 반대방향으로
조작량을 변화시킨다.
그렇기 때문에 빨간선과 파란선을 비교하면 서로 반대의 모양으로
급격하게 조작량이 변하는 것을 볼수 있다.

3. D **제어**
미분 - 즉 오차값의 변화를 보고 조작량을 결정하는 방법이다.

4. **조작량의 결정**
실제 제어기는 아날로그 값으로 제어하는 것처럼 되어 있지만 실제 컴퓨터를 사용하기 위해서는 샘플링을 통하여
들어온 이산값의 오차들을 이용하여 조작량을 결정하게 된다.

조작량 = Kp * **편차** + Ki * **편차의 누적값** + Kd * **편차의 변화량**
여기서 kp, Ki, Kd 가 제어기의 특성을 결정하게 된다.

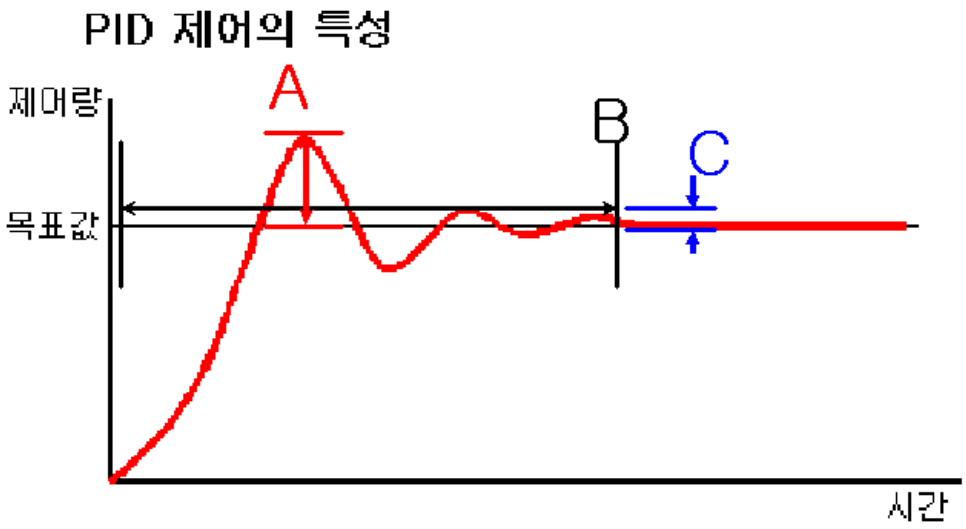
5. **제어기의 특성과** PID **각 요소들의 의미**.

제어기의 특성은 몇가지 지표로 결정이 되는데

PID 제어에 대하여

작성 : 애니엘㈜ 2014-02-25

그것은 제어기마다 다르다.
절대 목표값을 넘지 말아야하는 제어기도 있을수 있고,
무엇보다 빠르게 목표값에 도달해야 하는 제어(빠른응답)도 있을수 있다.
오차가 작은게 무엇보다 중요한 제어도 있을수 있다.



큰 지표가 되는 것은 3 가지이다.
(실제 제어공학을 배우면 특성을 나타내는 값들이 이것들보다 더 많고, 측정 방법도 조금 다르다.)

A. Overshoot - 오버슈트
목표값에 비해서 최고로 오차가 커지는 부분이 얼마인가를 보는 것이다.
이 값이 너무 커지면 시스템에 무리를 줄 수도 있다.
예를 들어 제어량이 전압이고 TR 을 동작시키는데 오버슈트가 너무 커지면 TR 의 동작 영역을 벗어나서 오동작을 하거나
TR 이 과전압에 의해 손상될수 있는 상태이다.

B. 목표값 도달 시간.
제어는 어차피 완전히 100% 수렴할수는 없다. 그래서 목표값의 몇%에 들어가면 제어가 완료된 것으로 본다.
어쨋든 제어가 완료되었다고 판단되는 시간.
그 시간이 짧을 수록 좋은 제어기이다.

C. 정상상태 오차
제어량이 목표량의 일정범위에 도달하였으나 없어지지 않고 남아있는 오차이다.

PID 와의 제어기의 특성

P 제어 : 목표값 도달 시간(B)을 줄인다.
I 제어 : 정상상태 오차(C)를 줄인다.
D 제어 : 오버슈트(현재치의 급변이나 외란-A)을 억제한다.

간단히 P, I, D 각자는 제어기의 특성에 대해서 위와 같은 효과를 발휘한다고 볼수 있다.
그러나 이는 단순한 문제는 아니며 P,I,D 의 조작량은 서로 영향을 미친다.

일반적으로 각 제어량의 게인을 조절하여 제어기의 특성을 셋팅하게 되는데
이는 이론적으로 어느정도 게인값을 결정한뒤 튜닝을 통하여 최적(이라고 생각되는) 게인값을 제어기에 적용하게 된다.