

PID 制御について (公開資料) (About PID-Control)		作成 2022/11/12 ニューラルソフト有限公司 市来 博記
初版	—	2022/11/12

概要

本書は、フィードバック制御の基礎となっている PID 制御を解説するものです。

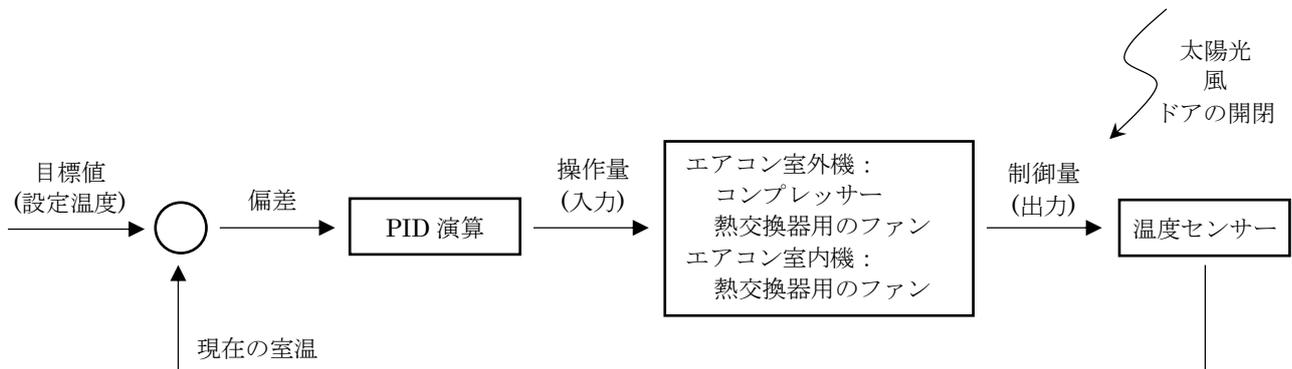
情報源

Wikipedia : PID 制御	ロボット・IT 雑食日記 : PID 制御について宇宙一わかりやすく解説してみる	HELLO CYBERNETICS : PID 制御の基本的な考え方	マクニカ 組込み技術ラボ わかりやすい PID 制御
------------------------------------	--	--	--

PID 制御とは

PID 制御は、古典制御論の枠組みで体系化されたもので長い歴史を持ち、様々な制御手法が開発・提案された今に至っても、産業界で主力の制御手法となっています。制御に必要な基本要素は、制御対象の現在値（制御量又は出力値）と目標値との偏差、その積分値（時間領域における偏差の累積値*1）、および微分値（偏差が変化する速さ＝単位時間（十分に短い時間）当たりの偏差の変化量）の 3 つで、これらの値から演算して得られる値が制御の操作量（入力）になります。又、理想的な環境以外（実世界）では制御を妨げる要素が存在し、それを外乱と呼びます。室内の温度制御で例えると、制御対象の現在値は室温、目標値はエアコンの設定温度になり、設定温度と室温の差を元に、時間領域における差の累積値と差の変化する速さを考慮に入れた演算によりエアコンの制御操作量*2 を決定します。（外乱の要素は、建物に照射される太陽光、建物の壁にあたる風、ドアの開閉などになります。）

PID 制御による温度制御のイメージ



*1 時間は連続しているので、厳密には積分値≠累積値です。

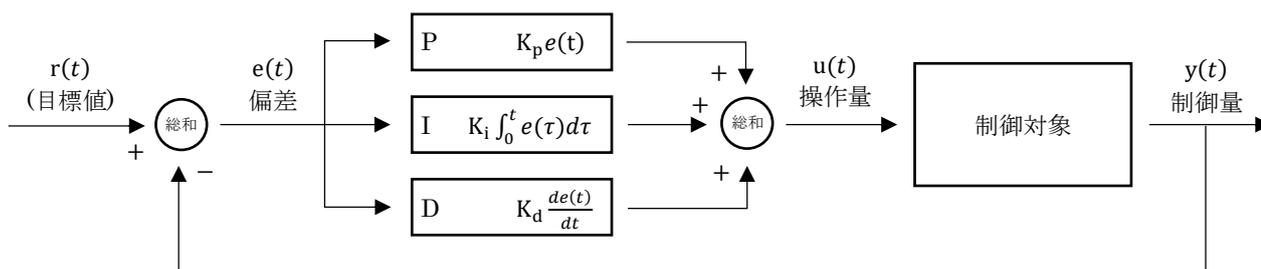
*2 エアコン室外機のコンプレッサーへのインバータ出力周波数や、室外機と室内機の熱交換器用のファンの回転速度を変更します。（操作量に応じて室内機から吹き出る温度の低い空気の量と温度を変更して、温度の高い室内の空気と置き換え拡散することで室温を設定温度に保ちます。）

PID 制御の詳細

PID 制御の P、I、D は、制御に必要な要素から操作量を決定するための演算方法の頭文字となっています。

- P : Proportional (比例)
- I : Integral (積分)
- D : Differential (微分)

制御に必要な要素、演算処理、操作量、制御量の関係を表したブロック線図は以下のようになります。



PID 制御の詳細は、目標値 $r(t)$ と制御量 $y(t)$ (制御対象の出力値 = 現在値) の偏差 $e(t)$ を「P (関数)」、「I (関数)」、「D (関数)」を入力して、その出力の和を制御対象に入力する流れになっています。

P 制御

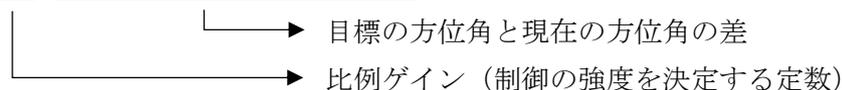
P 制御 (比例制御) は目標値と現在値の差に比例した操作量を制御対象に入力します。

$$u(t) = K_p e(t) = K_p (r(t) - y(t))$$

水平に設置された砲台の方位角を制御する例

モータに印加する電圧を V 、目標の方位角を θ_{Attack} 、現在方位角を $\theta_{Current}$ とすると、

$$V = K_p (\theta_{Attack} - \theta_{Current})$$



モータに印加する電圧 V は、目標の方位角と現在の方位角の差に比例した大きさになります。(符号はモータの回転方向を表します。)

目標の方位角と現在の方位角の差が大きいほど大きな電圧をモータに印加することになり、モータの回転速度が速くなります。(モータを回転させるトルクが大きくなります。)

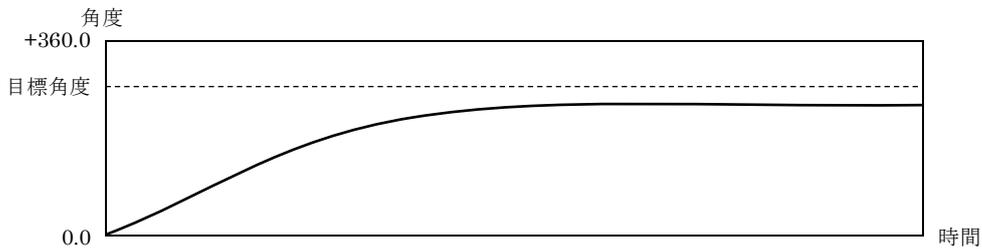
又、 K_p が大きいほど大きな電圧をモータに印加することになります。

P 制御の欠点

シンプルで直感的に分かり易い P 制御ですが、幾つかの欠点があります。

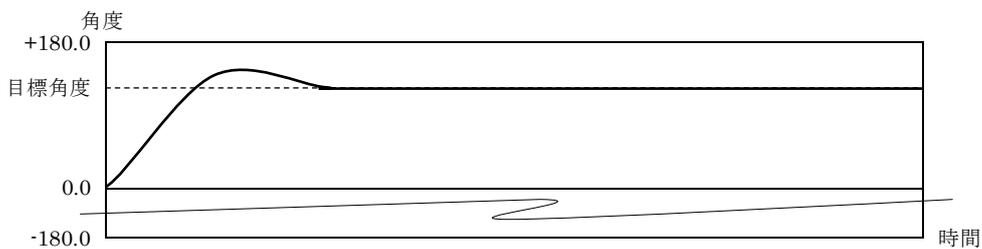
- K_p が小さいと現在値が目標値に近づいた時、制御の効力が消失する。
前の頁に示したモータ制御の場合、早い段階でモータを回転させるトルクがモータの回転抵抗より小さくなり（減速する）、目標角度に達する前に停止する可能性が大きくなります。

現在値が目標値に届かないことをアンダー シュートと言い、その時発生する偏差を定常偏差（残留偏差）と言います。 K_p が小さいと定常偏差は大きくなります。

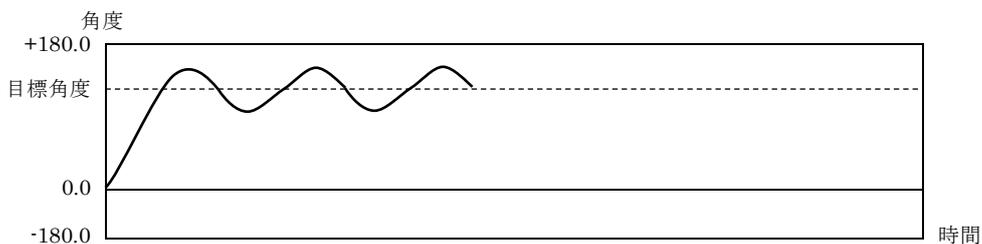


- K_p が小さいと現在値が目標値に近づくための時間が長くなる。
- K_p が大きいと現在値が目標値を大きく行き過ぎるようになります。
前の頁に示したモータ制御の場合、モータの回転速度が速くなること、モータが減速に転じるタイミングが遅くなることで現在角度が目標角度を大きく行き過ぎるようになります。

現在値が目標値を超えることをオーバー シュートと言います。



- K_p が大きすぎると現在値が目標値の上下で振動する現象が発生します。



PI 制御

PI 制御は、P 制御の欠点を解決する為、P 制御に I（積分）制御を追加した制御です。
以下に I 制御関数を示します。

$$K_i \int_0^t e(\tau) d\tau$$

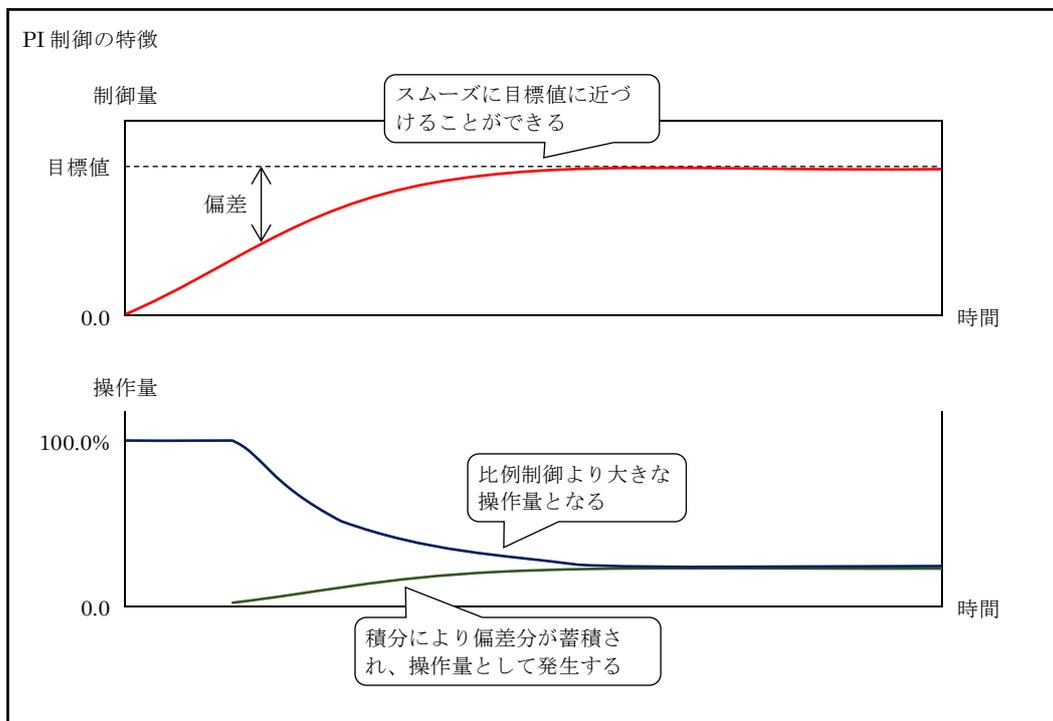
→ 時間領域における偏差の積分値（≒累積値）
→ 積分ゲイン（制御の強度を決定する定数）

時間領域における偏差の積分値に定数 K_i を掛けたものが積分制御関数の出力値になります。

通常、I 制御は P 制御の欠点を補うために使用され、単独では使用されることはありません。

PI 制御（比例積分制御）は、比例制御関数と積分制御関数の出力値の和を操作量として制御対象に入力します。

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau$$



PI 制御では操作量算出時に、瞬時の偏差に偏差の累積値を加味することで現在値を目標値に到達させることができます。

PID 制御

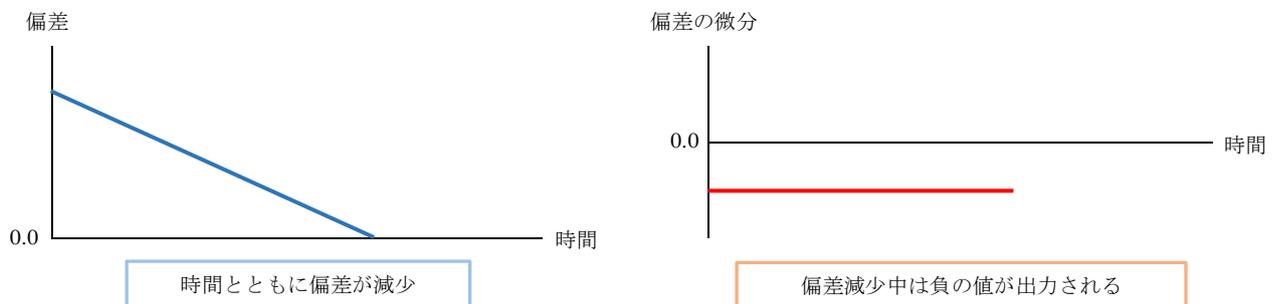
PI 制御は、現在値を目標値に到達させることができますが、目標値に高速に近づけるために、P 制御の成分を大きくしすぎるとオーバーシュートが発生することがあります。このオーバーシュートを抑えるための負成分が D 制御(微分制御)になります。

以下に D 制御関数を示します。

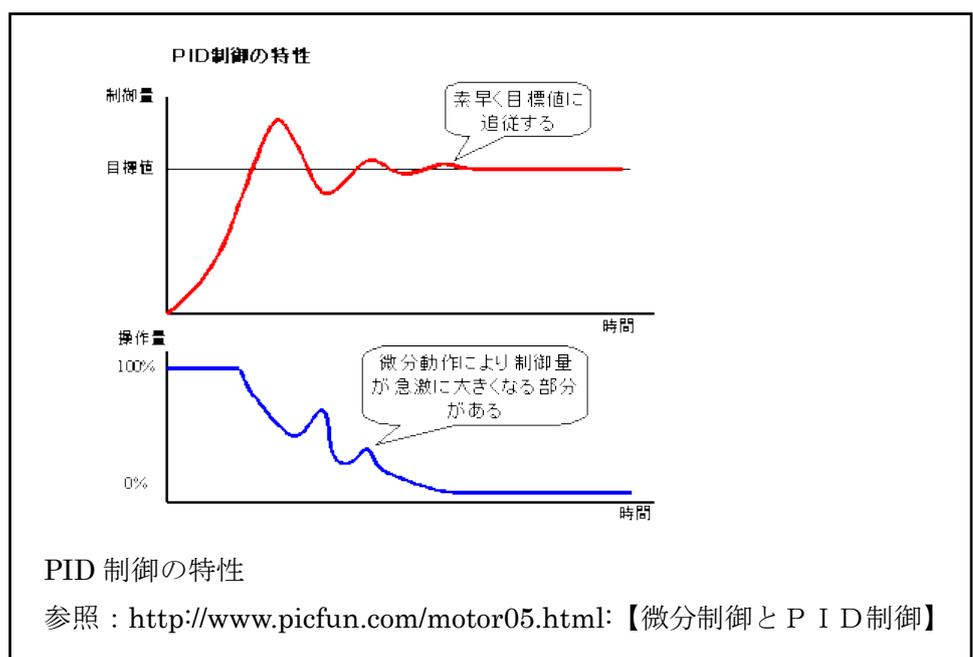
$$K_d \frac{de(t)}{dt}$$

→ 時間領域における偏差の微分 (偏差の変化の傾き)
→ 微分ゲイン (制御の強度を決定する定数)

時間領域における偏差の微分値に定数 K_d を掛けたものが微分制御関数の出力値になります。



D 制御は時間経過による偏差の変化、すなわち瞬間的な変化に反応します。微分をするということは傾きを求めることであり、傾きを求めることで、少し先の未来に偏差がどのような値になっているのかを予測することができます。このことから、PI 制御に D 制御を加えることで素早く目標値に追従し、外乱などの変化に強い制御を行うことができます。



PID 制御は D 制御の役割によって、より積極的に制御量を調整することができます。
操作量は比例制御関数、積分制御関数、微分制御関数の出力値の和になります。

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}$$

PID 制御の C 言語プログラム例

(参照 : <https://emb.macnica.co.jp/articles/15859/> : 【PID 制御の C 言語プログラム例】)

PID 制御のプログラムは非常にシンプルになっており、以下のように記述することで実現できます。

```
/* feedback_val : 現在値 */
/* target_val : 目標値 */
/* integral : 偏差の積分値 */
float pid_culc(float feedback_val, float target_val)
{
    float p, i, d;

    error[0] = error[1]; /* error[0] : DELTA_T 時間前の偏差 */
    error[1] = feedback_val - target_val; /* error[1] : 現在の偏差 */
    integral += (error[1] + error[0]) / 2.0 * DELTA_T;

    p = KP * error[1];
    I = KI * integral;
    d = KD * (error[1] - error[0]) * Inverse(DELTA_T);

    return p + i + d;
}
```

所感

PID 制御は非常にシンプルな制御である反面、P、I、D のゲイン値に最適な値を設定することが非常に重要であることが最認識できました。基本的にゲイン値は、経験則に基づいた試行錯誤的な方法で決定することになると思いますが、幾つかの調整法が提唱されています。これらの調整法についての解説は、この資料の次の版で追記します。