

計装豆知識

PID制御のアルゴリズム

PID制御は、プロセス産業をはじめとする様々な分野に広く普及している、最も標準的な制御方式です。PID制御のアルゴリズム（演算方式）は、P（比例項演算）、I（積分項演算）、D（微分項演算）の組合せによって構成されますが、各演算の組合せ方式によって制御出力の挙動が変わり、制御の効果が少しずつ異なってきます。ここでは、代表的なPID制御のアルゴリズムを取上げ、その特徴と動作の概要について説明します。

PID制御（基本形）

PID制御では設定値（SP）と測定値（PV）の偏差（e）に対して、**P**（比例項演算）、**I**（積分項演算）、および**D**（微分項演算）を行い、それぞれの演算結果を加算して制御出力（MV）を算出します。この制御方式のアルゴリズムを図1に、演算式を下記(1)式に示します。

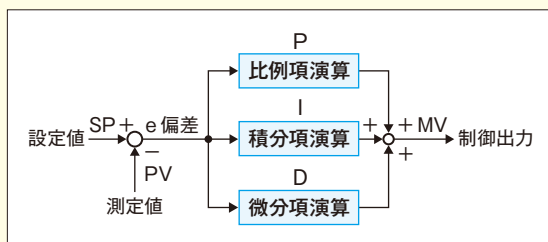


図1 PID制御（基本形）のアルゴリズム

$$MV(t) = \underbrace{K_p e(t)}_{\text{比例項演算}} + \underbrace{\frac{K_p}{T_i} \int e(t) dt}_{\text{積分項演算}} + \underbrace{k_p T_d \frac{de(t)}{dt}}_{\text{微分項演算}} \quad \dots (1)$$

MV(t): 制御出力（時間変数） **e(t)**: 偏差（= SP(t) - PV(t） **Kp**: 比例ゲイン（= $\frac{100}{PB}$ ）
Ti: 積分時間 **Td**: 微分時間 **PB**: 比例帯

PID制御では設定値、および測定値のいずれの変更に対しても制御出力が同様の挙動を示します。このため、設定値を変更する場合、値を急変させると微分項演算の働きで制御出力が突変することがあります（微分キック）。PID制御は主に設定値変更に対する制御ループの追従性を重視する場合に用いられます。PID制御は最も一般的な制御方式であり、多くの調節計やDCSに採用されています。

PI-D制御（微分先行形PID制御）

PI-D制御では、偏差に対して比例項演算、積分項演算、そして測定値に対して微分項演算を行い、それぞれの演算結果を加算して制御出力（MV）を算出します。この制御方式のアルゴリズムを図2に、演算式を下記(2)式に示します。

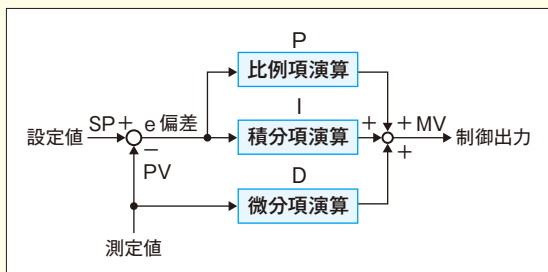


図2 PI-D制御（微分先行形PID制御）のアルゴリズム

$$MV(t) = \underbrace{K_p e(t)}_{\text{比例項演算}} + \underbrace{\frac{K_p}{T_i} \int e(t) dt}_{\text{積分項演算}} + \underbrace{k_p T_d \frac{dPV(t)}{dt}}_{\text{微分項演算}} \quad \dots (2)$$

PI-D制御では、測定値の変更のみに対して微分項演算を行い、設定値の変更に対しては微分項演算を行いません。このため、PID制御に比べ設定値の変更時の微分キックが低減されます。PI-D制御は、たとえばカスケード制御系の下流側（セカンダリループ）などに使用され、制御系全体の追従性を良くします。測定値の変動に対してはPID制御と同様に動作するので、PID制御と同様に多くの調節計やDCSに採用されています。

I-PD制御（比例微分先行形PID制御）

I-PD制御では、偏差に対して積分項演算を行い、測定値に対して比例項演算、および微分項演算を行い、それぞれの演算結果を加算して制御出力を算出します。この制御方式のアルゴリズムを図3に、演算式を(3)式に示します。

I-PD制御では設定値の変更に対しては積分項演算のみを行うので、たとえば設定値をステップ状に変更しても出力はランプ状に緩やかに変化します。したがって、設定値変更に対する微分キックが最も抑制されます（前述、PI-D制御では設定値変更に対して比例項演算も行います）。一方、測定値の変動に対しては比例、積分、微分各項の演算を行うので良好な制御性が得られます。

I-PD制御は設定値変更に対する追従性はPID制御に劣りますが、応答が穏やかなので、安定性や安全性を重視した制御目的に採用されます。

$$MV(t) = \underbrace{K_p PV(t)}_{\text{比例項演算}} + \underbrace{\frac{K_p}{T_i} \int e(t) dt}_{\text{積分項演算}} + \underbrace{k_p T_d \frac{dPV(t)}{dt}}_{\text{微分項演算}} \quad \dots (3)$$

微分動作の完全微分形と不完全微分形について

前述してきた各種のPIDアルゴリズムにおける演算式では、微分項演算を理想的な微分演算として示してあります（式(1)～(3)参照）。この微分項の演算方式を「完全微分形」といいます。

しかし、実際のプロセスにおいては測定値（計測信号）に重畳している高周波ノイズが完全微分によって過度に増幅されて制御系を不安定にしたり、設定値を変更した場合の微分キックを過大にしたりというような問題が生じます。そこで、この問題を解決するのが「不完全微分形」による演算方式です。不完全微分形の演算式は完全微分演算と1次遅れ演算を組合せた形になっていて、微分項演算の入力信号に含まれる高周波成分を除去し、信号の突変を抑える効果があります。完全微分形と不完全微分形の演算式をPID制御方式（基本形）を例にとり下記に示します。

● 完全微分形のPID制御の演算式（(1)式をラプラス変換）

$$MV(s) = K_p e(s) + \frac{K_p}{T_i s} e(s) + \underbrace{K_p T_d s e(s)}_{\text{微分項演算}} \quad \dots (4)$$

s: ラプラス演算子

● 不完全微分形のPID制御の演算式

$$MV(s) = K_p e(s) + \frac{K_p}{T_i s} e(s) + \underbrace{\frac{K_p T_d s}{1 + \frac{T_d s}{m}} e(s)}_{\text{微分項演算}} \quad \dots (5)$$

m: 微分ゲイン（8～10の定数）

図4に偏差をステップ状に変化させた場合の完全微分形と不完全微分形の制御出力の応答を示します。一般に、実際の調節計やDCSにおいては、不完全微分形の微分演算が採用されています。

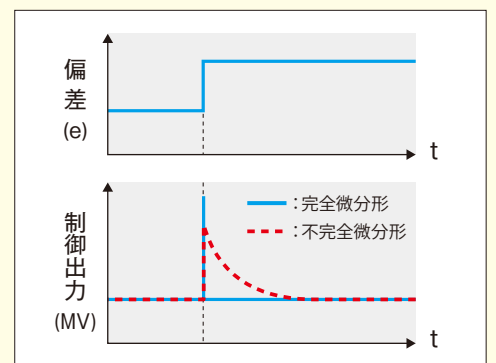


図4 完全微分形と不完全微分形の制御出力のステップ応答

＜参考文献＞

高津春雄 編著：プロセス制御（コロナ社）

千本 資・花淵 太 共編：計装システムの基礎と応用（オーム社）

広井和男：PID制御のお話

（MGTrend（旧誌名：MST、エムエスツデー）2004年2月号～12月号に連載）

【(株)エムジー 広報部】