PLC のディジタル PID の設計

◆ディジタル化には Z 変換を使うべきか？

音楽をディジタル録音する場合，例えば，サンプリング周波数を 48kHz，ローパスフィルタのカットオフを 16kHz とすると，Z 変換を使わないと正確なディジタルフィルタを設計できないであろう。理由は，サンプリング周期に対してローパスフィルタのタイムファクタが 3 倍しかないので離散値のラプラス変換と言われる Z 変換を使うことが必須である。

一方，プロセス制御では，PLC の PID のスキャンレートが 0.01 秒ないし 0.1 秒に対してプロセスのタイムファクタが 10 倍ないし 100 倍超なら次の微分積分の定義がそのまま適用できるのである。

$$\frac{dy}{dx}=\lim_{\Delta x\to 0}\frac{\Delta y}{\Delta x}$$

$$\int_a^b y(x)dx=\lim_{\Delta x\to 0}\sum_{i=1}^{n} y_i\Delta x$$

つまり，PLC の中ではディジタル演算なのだがアナログ理論がそのまま使えるのである。実際にブロック図ではアナログ扱いになっている。

◆基本形 PID

基本形 PID の制御則は，比例項，積分項，微分項を組み合わせて次式のようになる。比例定数 K_P をカッコの外にくくり出すのは伝統的な表現である。

$$\mathrm{MV}(t)=K_P\left\{\mathrm{EV}(t)+\frac{1}{T_I}\int_0^t \mathrm{EV}(\tau)d\tau+T_D\frac{d}{dt}\mathrm{EV}(t)\right\}$$

これをラプラス変換すれば線形になり EV がカッコの外にくくり出せるので次式のように伝達関数が算出できる。

$$\frac{\mathrm{MV}(s)}{\mathrm{EV}(s)}=K_P\left\{1+\frac{1}{T_I s}+T_D s\right\}$$

制御系全体のブロック図は次のようになる。

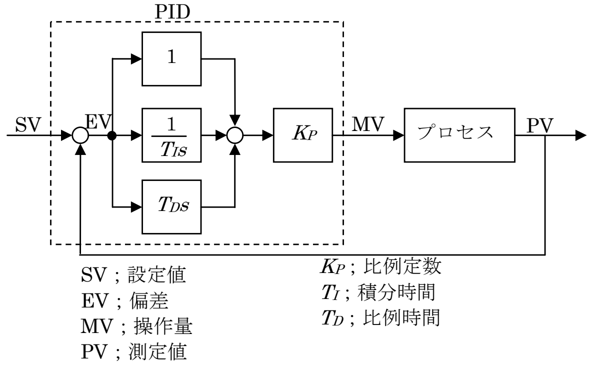

図　基本形 PID

◆PI-D と I-PD

基本形 PID の変形である PI-D，I-PD のブロック図を次に示す。

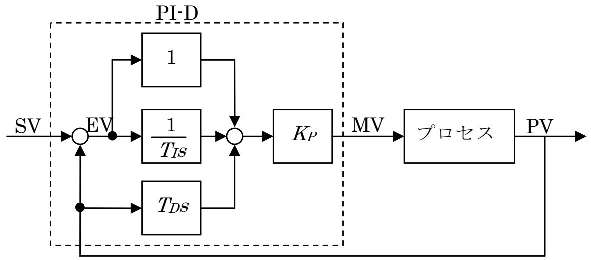

図　PI-D（測定値微分型，微分先行型）

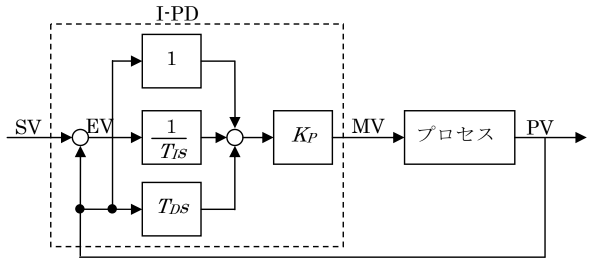

図　I-PD（測定値比例微分型，比例微分先行型）

PI-D は測定値微分型 or 微分先行型と言われる。I-PD は測定値比例微分型 or 比例微分先行型と言われる。

これらの変形がつくられた理由は，基本形 PID が SV の変化特にステップのときに MV の変化が敏感すぎてしまうからである。

電磁弁がステップに開くとウォーターハンマーという現象が起き故障の原因になりかねない。このような理由で，微分項または比例項＋微分項の入力から SV を除きその入力を PV だけとしたのである。

◆完全微分と不完全微分

基本形 PID の微分項は完全微分と言われ，ステップ入力のとき出力がインパルス（幅 0 高さ∞）になってしまい，物理的に実現もできないし，できたとしても操作量として適当ではない。そこで，微分項に 1 次遅れ（ローパスフィルタ）を加味した不完全微分を採用することが多い。

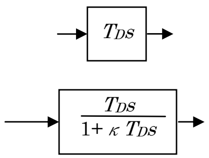

図　完全微分と不完全微分

PLC では，次のように書くことが多い。

$$\kappa = \frac{1}{K_D}, \quad \kappa T_D = \frac{T_D}{K_D}$$

K_D：微分ゲイン

κ ：特に名称なし（不完全微分係数なんてどうですかね）

不完全微分のステップ応答は 1 次遅れの上下反転となる。

PLC のディジタル PID の設計

◆基本形 PID 演算式

微分積分のディジタル化を次のように定義する。

$$\frac{d\mathrm{EV}(t)}{dt} = \frac{\Delta \mathrm{EV}_n}{\Delta t} = \frac{\mathrm{EV}_n - \mathrm{EV}_{n-1}}{\Delta t}$$

$$\int_0^t \mathrm{EV}(\tau)d\tau = \sum_{i=0}^{n} \mathrm{EV}_i \Delta t$$

これらを基本形 PID に代入すると次式が得られる。

$$\mathrm{MV}_n = K_P \left\{ \mathrm{EV}_n + \frac{\Delta t}{T_I}\sum_{i=1}^{n}\mathrm{EV}_i + \frac{T_D}{\Delta t}\left(\mathrm{EV}_n - \mathrm{EV}_{n-1}\right) \right\}$$

上式はいわゆる位置型アルゴリズムである。実際の PLC の中では，リアルタイムに演算しなければならないので，速度型アルゴリズムにしなければならない。上式の 1 スキャン前は，

$$\mathrm{MV}_{n-1} = K_P \left\{ \mathrm{EV}_{n-1} + \frac{\Delta t}{T_I}\sum_{i=1}^{n-1}\mathrm{EV}_i + \frac{T_D}{\Delta t}\left(\mathrm{EV}_{n-1} - \mathrm{EV}_{n-2}\right) \right\}$$

であり，その差分は，

$$\Delta \mathrm{MV}_n = \mathrm{MV}_n - \mathrm{MV}_{n-1}$$

$$\Delta \mathrm{MV}_n = K_P \left\{ \left(\mathrm{EV}_n - \mathrm{EV}_{n-1}\right) + \frac{\Delta t}{T_I}\mathrm{EV}_n + \frac{T_D}{\Delta t}\left(\mathrm{EV}_n - 2\mathrm{EV}_{n-1} + \mathrm{EV}_{n-2}\right) \right\}$$

となる。実際の PLC の中では，リアルタイムに速度型アルゴリズムの次式を演算している。

$$\mathrm{MV}_n = \mathrm{MV}_{n-1} + \Delta \mathrm{MV}_n$$

◆不完全微分の演算式

不完全微分の伝達関数は次式である。

$$\frac{Y(s)}{\mathrm{EV}(s)} = \frac{T_D s}{1 + \kappa T_D s}$$

上式の分母をはらって，微分方程式に戻すと，

$$Y(s) + \kappa T_D s Y(s) = T_D s \mathrm{EV}(s)$$

$$y + \kappa T_D \frac{dy}{dt} = T_D \frac{d}{dt}\mathrm{EV}$$

となり，ディジタル化すると，

$$y_n + \kappa\frac{T_D}{\Delta t}\left(y_n - y_{n-1}\right) = \frac{T_D}{\Delta t}\left(\mathrm{EV}_n - \mathrm{EV}_{n-1}\right)$$

となる。整理して，

$$y_n = \frac{\kappa T_D}{\Delta t + \kappa T_D} y_{n-1} + \frac{T_D}{\Delta t + \kappa T_D}\left(\mathrm{EV}_n - \mathrm{EV}_{n-1}\right)$$

が得られる。1 スキャン前は，

$$y_{n-1} = \frac{\kappa T_D}{\Delta t + \kappa T_D} y_{n-2} + \frac{T_D}{\Delta t + \kappa T_D}\left(\mathrm{EV}_{n-1} - \mathrm{EV}_{n-2}\right)$$

となり，その差分は，

$$y_n - y_{n-1} = \frac{\kappa T_D}{\Delta t + \kappa T_D}\left(y_{n-1} - y_{n-2}\right) + \frac{T_D}{\Delta t + \kappa T_D}\left(\mathrm{EV}_n - 2\mathrm{EV}_{n-1} + \mathrm{EV}_{n-2}\right)$$

$$\Delta y_n = \frac{\kappa T_D}{\Delta t + \kappa T_D}\Delta y_{n-1} + \frac{T_D}{\Delta t + \kappa T_D}\left(\mathrm{EV}_n - 2\mathrm{EV}_{n-1} + \mathrm{EV}_{n-2}\right)$$

となる。実際の PLC の中では，リアルタイムに速度型アルゴリズムの次式の演算している。

$$\mathrm{MV}_n = \mathrm{MV}_{n-1} + \Delta\mathrm{MV} = \mathrm{MV}_{n-1} + K_P\left\{\left(\mathrm{EV}_n - \mathrm{EV}_{n-1}\right) + \frac{\Delta t}{T_I}\mathrm{EV}_n + \Delta y_n\right\}$$

◆PI-D と I-PD の演算式

PI-D の演算式は微分項の EV を－PV に置き換えるだけである。

I-PD の演算式は比例項と微分項の EV を－PV に置き換えるだけである。

1 例として，不完全微分 PI-D の演算式は次のようになる。

$$\Delta\mathrm{MV} = \mathrm{MV}_n - \mathrm{MV}_{n-1} = K_P\left\{\left(\mathrm{EV}_n - \mathrm{EV}_{n-1}\right) + \frac{\Delta t}{T_I}\mathrm{EV}_n + \Delta y_n\right\}$$

$$\Delta y_n = \frac{\kappa T_D}{\Delta t + \kappa T_D}\Delta y_{n-1} + \frac{T_D}{\Delta t + \kappa T_D}\left(-\mathrm{PV}_n + 2\mathrm{PV}_{n-1} - \mathrm{PV}_{n-2}\right)$$

PLC では，次のように書くことが多い。

$$\kappa = \frac{1}{K_D}, \quad \kappa T_D = \frac{T_D}{K_D}$$

K_D：微分ゲイン

κ ：特に名称なし（不完全微分係数なんてどうですかね）

◆PV のローパスフィルタ

プロセスから戻る電気信号にノイズが乗る場合には，PV にローパスフィルタを付けるのが望ましい。ローパスフィルタは 1 次遅れとする。

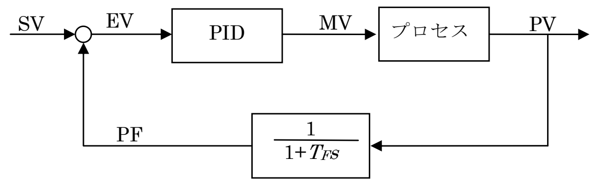

図　PV のローパスフィルタ

PV のローパスフィルタの伝達関数は次式である。

$$\frac{\mathrm{PF}(s)}{\mathrm{PV}(s)} = \frac{1}{1 + T_F s}$$

上式の分母をはらって，微分方程式に戻すと，

$$\mathrm{PF}(s) + T_F s\mathrm{PF}(s) = \mathrm{PV}(s)$$

$$\mathrm{PF} + T_F\frac{d}{dt}\mathrm{PF} = \mathrm{PV}$$

となり，ディジタル化して整理すると次式が得られる。

$$\mathrm{PF}_n + \frac{T_F}{\Delta t}\left(\mathrm{PF}_n - \mathrm{PF}_{n-1}\right) = \mathrm{PV}_n$$

$$\mathrm{PF}_n = \frac{T_F}{\Delta t + T_F}\mathrm{PF}_{n-1} + \frac{\Delta t}{\Delta t + T_F}\mathrm{PV}_n$$

$$\mathrm{PF}_n = \frac{T_F}{\Delta t + T_F}(\mathrm{PF}_{n-1} - \mathrm{PV}_n) + \mathrm{PV}_n = \alpha(\mathrm{PF}_{n-1} - \mathrm{PV}_n) + \mathrm{PV}_n$$

$$\text{where} \quad \alpha = \frac{T_F}{\Delta t + T_F} < 1, \ \text{where} \ T_F \gg \Delta t$$

α は 1 に近くなる。1 次のローパスフィルタでは気休めに過ぎないであろう。あまり使われていないと考える。使わないときは，$\alpha = 0$ とする。

◆プロセス特性

PID パラメータの最適調整のためにはプロセス側の特性を特定しなければならないが，普通はステップ応答を次図のジグモイド曲線と仮定する。

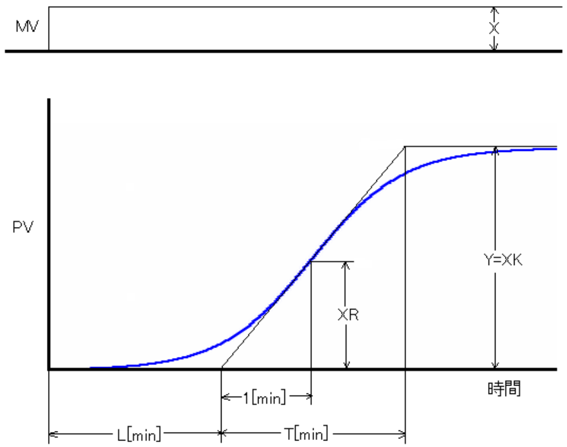

図　プロセスの応答特性

上図の伝達関数は次のように 1 次遅れとむだ時間の結合となる。

$$P(s)=\frac{K}{1+Ts}e^{-LS}$$

K：プロセスゲイン

T：時定数（1 次遅れの）

L：むだ時間

多くの専門書では単位ステップ応答（$X=1$）とするので縦軸の表現が上図とは少し異なる。しかし，MV の単位ステップとは何かを定義することは困難である。シンプルに MV=X としたとき PV=Y（定常値）になれば，プロセスゲイン K は次式となる。

$$K=\frac{Y}{X}=\frac{\mathrm{PV}}{\mathrm{MV}}$$

多くの専門書ではジグモイド曲線の傾きを反応速度 R としている。単位ステップをきちんと定義できないので混乱の基になるが次式の関係がある。

$$R=\frac{K}{T},\quad RL=\frac{KL}{T}$$

R：反応速度

K：プロセスゲイン

T：時定数（1 次遅れの）

L：むだ時間

PV が定常値に達するまで待てない場合はプロセスゲイン K の代わりに反応速度 R が使える。上図で反応速度 R を算出するときは上昇途中の Y を時間と X で除することで得られる。

一般的に，むだ時間 L が小さいほど，時定数 T が大きいほど，つまり，L/T が小さいほど制御が容易である。

◆PID パラメータ最適調整

プロセス側の特性が既知ならば数値シミュレーションなどで PID パラメータは最適調整できる。多くの専門書に掲載されている代表的なものが次表である。係数は無次元であることに注意されたい。

提案者	タイプ	PID	K_P	T_I	T_D
Ziegler & Nichols	目標値追従 外乱抑制	P	$1.0/RL$		
		PI	$0.9/RL$	$3.3L$	
		PID	$1.2/RL$	$2.0L$	$0.5L$
Chien Hrones & Reswick （CHR 法）	目標値追従 行過ぎ 0%	P	$0.3/RL$		
		PI	$0.35/RL$	$1.2T$	
		PID	$0.6/RL$	$1.0T$	$0.5L$
	目標値追従 行過ぎ 20%	P	$0.7/RL$		
		PI	$0.6/RL$	$1.0T$	
		PID	$0.95/RL$	$1.35T$	$0.47L$
	外乱抑制 行過ぎ 0%	P	$0.3/RL$		
		PI	$0.6/RL$	$4.0L$	
		PID	$0.95/RL$	$2.4L$	$0.4L$
	外乱抑制 行過ぎ 20%	P	$0.7/RL$		
		PI	$0.7/RL$	$2.3L$	
		PID	$1.2/RL$	$2.0L$	$0.42L$

Ziegler & Nichols 法では，オーバーシュートの第 2 波の行過ぎが第 1 波のそれの 1/4 になるようにパラメータが選ばれている。CHR 法では，オーバーシュートの第 1 波の行過ぎが 0%または 20%になるようにパラメータが選ばれている。

◆PID パラメータの手動調整

いろいろな理由でプロセス側の特性をきちんと調査できないことも多い。その場合は PID パラメータを手動調整することになる。

① P→I→D の順で調整する。予め，P は最小，I は最大，D は最小にしておく。理由は，この状態は PID がまったく効いていない状態だからである。

② P（比例定数 K_p）は小さい数字から大きい数字へと変える。測定値が振動ぎみになってきたら少し戻す。

③ I（積分時間 T_I）は大きい数字から小さい数字へと変える。測定値の振動ぎみなってきたら少し戻す。

④ D（微分時間）は小さい数字から大きい数字へと変えていく。測定値の振動ぎみなってきたら少し戻す。

⑤ 目標値到達時間を重要視するか定常値制定時間を重要視するかにより PID パラメータを微調整する。

■PLC のディジタル PID の設計の実際

◆物理空間と PID の接続の設計

PLC のディジタル PID は PLC メーカーの既製品であり，ユーザーが設計・製作することはない。ユーザーは PV，SV，MV の物理量と内部変数の関係を設計しなければならない。次図は物理空間と PID の接続の概略である。

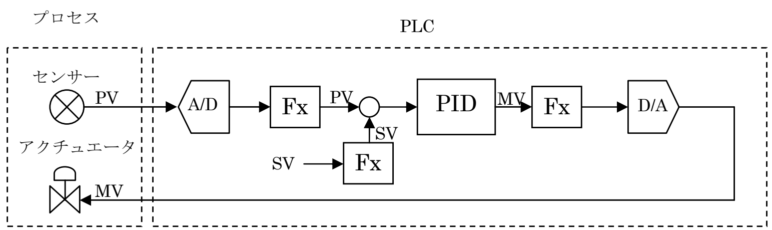

図　PID 周辺ブロック図

PID の入力は PV と SV であり出力は MV であり，普通は 2Byte の整数である。一方，物理空間にも，PV と SV と MV があり，SV は普通パソコン（旧くは CRT と呼ばれていた）から設定される。　PID の入出力の範囲は，PLC の機種によっては 0～2000 のように固定のものもあるが，任意に設定できる機種もあり普通は 2Byte（-32768～0～+32767）に収まればよい。

◆AD 変換，DA 変換

AD 変換のアナログ入力と DA 変換のアナログ出力の電気信号は，1～5V，4～20mA などいろいろな仕様があり，これをセンサー・アクチュエータの物理量の仕様により，例えば，0～20L/min，0～100%などに割り当てる。次に，PID の PV を 0～200，0～2000，0～10000 など任意に選ぶことができる。また，物理量の MV が 0～100%であれば，PID の MV は 0～1000，0～2000，0～10000 など任意に選ぶことができる。

物理空間の PV，MV と PID 内部変数の PV，MV の関係を工学値変換係数と呼ぶ。パソコンで設定される物理量の SV と PID の SV の関係（工学値変換係数）は PV のそれと同じにしなければならない。ただし，PV のフルスケール

とは別に SV にリミットを設定されることが多い。

◆AD 変換・DA 変換のスケーリングの設計の実際

AD 変換のディジタル出力と DA 変換のディジタル入力は，普通は 4000 とか 12000 などのように固定されている。そこで，スケーリングして PID 内部変数に変換するのである。例えば，物理量 PV が 0～20L/min，PID の PV が 0～2000 の場合，AD 変換のアナログ入力が MAX のときのディジタル出力が 12000 であれば，スケーリングで 1/6 にすればよいのである。

SV については，普通はパソコン（CRT ともいう）で 20.00L/min と入力したときにパソコン側で工学値変換して PLC のレジスタに 2000 を渡すように設計するのである。

このようにスケーリングしたあとは，AD 変換のアナログ入力と DA 変換のアナログ出力の電気信号とその変換の仕様は工学値変換係数と全く関係なくなるので，電気信号とその変換の仕様はいつでも仕様変更できるのである。

◆工学値変換係数の設計

PID の入力と出力のフルスケールを一致させる設計方法と一致させない設計方法がある。工学値変換係数を単位ごとに統一すると必然的に後者になる。前者は PID のパラメータの設計やスケーリングの設計が統一できるが工学値変換係数の設計が複雑になる。

最初に，PID の入力と出力の範囲が同じでない例をあげる。

	物理量	PID 内部変数	工学値変換係数
PV，SV	0～20.00L/min	0～2000	0.01
MV	0～100.0%	0～1000	0.1

工学値変化係数は 10 の倍数が望ましい。たまたま，工学値変換係数が同じになる場合もある。

アクチュエータは比例電磁弁やヒーターなどがあり，操作される物理量は流量や電力（温度）であるが，実際には，バルブ開度やヒーター電力をパーセント表示することが多い。理由は，仕様変更しても設計のやり直しがない

ようにするためである。また，PID ループが複数ある場合は，物理量と PLC 内部変数の関係（工学値変換係数）は単位ごとに統一した方がよい。このような設計では必然的に PID の入力と出力の範囲が同じでなくなる。

物理量のプロセスゲイン K は，

$$K = \frac{\mathrm{PV}}{\mathrm{MV}} = \frac{20\mathrm{L/min}}{100\%} = 0.2\mathrm{L/min/\%}$$

であるが，内部変数のプロセスゲイン K は，工学値変換係数を考慮して，

$$K = \frac{\mathrm{PV}}{\mathrm{MV}} = \frac{20\mathrm{L/min}\ /\ 0.01}{100\%\ /\ 0.1} = \frac{2000}{1000} = 2$$

となる。最適調整で使うプロセスゲイン K は後者であることに注意する。

例えば，PID 内部変数の PV，MV を 0～2000 として，最適調整して得られた比例定数を K_P とすると，あとで MV を 0～1000 に仕様変更すれば，比例定数は元の K_P の 1/2 になるのである。

◆スケーリングの Fx と内部変数の整数演算

PLC 内部変数の PV，MV の工学値変換のスケーリングに乗除算ではなく関数発生 Fx を推奨する理由は，非直線性補正をしたい場合があるからと乗除算より関数発生 Fx の方が汎用性があるからである。

一方，PLC の演算速度の関係でスケーリングに関数発生 Fx を使いたくない場合は，また，スケーリング以外のすべての演算は整数演算を推奨する。理由は PLC の内部変数はすべて整数だからである。整数演算の例として物理空間で次式の割合演算を考える。

$$Z = \frac{X}{X+Y} \times 100$$

よくやる方法は，すべての数値を浮動小数点にしてから演算して整数に戻す方法である。しかし，内部変数 X，Y，Z，X+Y が 2Byte（-32768～0～+32767）を越えなければ整数演算が使える。

	物理量	PID 内部変数	工学値変換係数
X，Y	0～20.00L/min	0～2000	0.01
Z	0～100.0%	0～1000	0.1

上式と上表から変数の適当な値を使って次式の A を求める。

$$500 = \frac{2000 \times A}{2000 + 2000}, \quad A = 1000$$

実際の PLC では，X×1000 を先にやり 4Byte にして，X＋Y の 2Byte で除すればよい。

ほとんどすべての内部変数が 2Byte を超えないので必ずこの演算パターンが使えるはずである。

この方法は，数値演算コプロが高価だったころ，工学演算をすべて整数演算にするテクニックであり，三角関数も整数演算でやったものである。

◆バンプレス切替

PID 制御を自動，MV 直接設定を手動と言うことにすると，自動と手動の切替時に MV が変化しないようにすることをバンプレス切替と言う。その実現にはトラッキングという手法を使う。

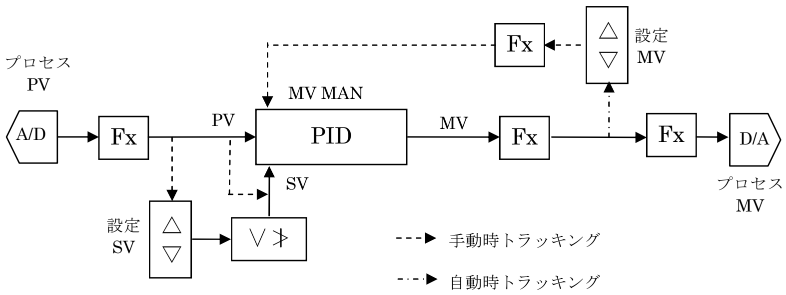

図　トラッキングのブロック図

PLC の PID ループはその手動時には，SV が PV に書き換えられ（偏差が 0 になる），また，MV MAN が MV に出力される。さらに，プログラムで自動の設定 SV にも PV を書きこんでおく。こうしておけば，手動から自動への切替時に MV が変化しなくなる。

自動時には，プログラムで手動の設定 MV に PID の MV を書きこむ。こうすれば，自動から手動への切替時に MV は変化しない。

さらに，自動の設定 SV をステップ状に変化したときは PID の SV の変化率を制限する。そうすることによって，手動から自動へ切替すると，その瞬間は MV は変化せず，自動の設定 SV が手動の設定 MV と異なる場合でも，その後は PLC の SV は手動の SV（つまり PV）から自動の設定 SV に変化率が制限されながら到達する。

◆非直線性補正

次図のようにバルブ開度と流量の関係が直線になるとは限らない。0 付近に不感帯があるものもある。本来，PID 制御はそのような非直線性の特性があっても問題なく制御でき， PV は SV に必ず一致させることができるのである。が，どの部分でも最適調整したい場合は非直線性補正をする必要がある。

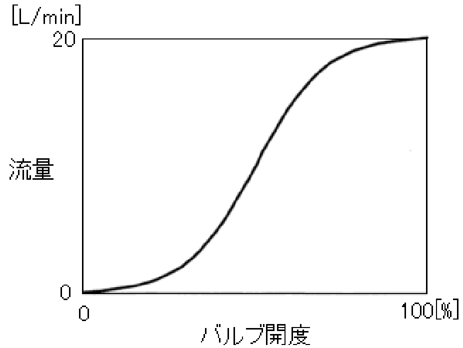

図　バルブ開度と流量の特性曲線の 1 例

1 例をあげる。

	物理量	PID 内部変数	工学値変換係数
PV，SV	0～20.00L/min	0～2000	0.01
MV	0～100.0%	0～1000	0.1

次図のブロック図を設計するとき最初に決めるべきは，自動の設定 SV と手動の設定 MV の工学値変換係数であろう。つまり，物理量と内部変数の関係を決めるのである。複数の PID ループがある場合，単位ごとに工学値変換係数を統一すれば必然的に PV と MV の内部変数の範囲が異なることになり，ス

ケーリング（関数発生）Fx を２連にしなければならない。PV と MV の内部変数の範囲が同じであればスケーリング（関数発生）Fx を１連できるが，工学値変換係数の設計が複雑になる。

この例の設計では，PID の MV はバルブ開度であり，最初の Fx で流量に直線変換して，次の Fx で特性曲線の縦軸を入力，横軸を出力としてバルブ開度に変換する。例えば，0 付近で MV が少し上がればプロセス MV が大きく上がるような非直線性補正をしてくれるのである。MV MAN のトラッキングの Fx はその逆になる。

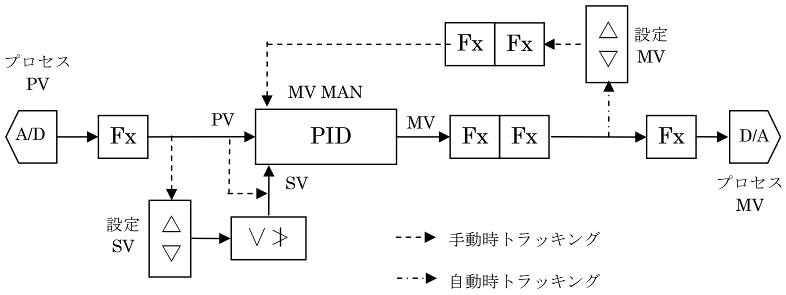

図　非直線性補正

もし，PV と MV の内部変数の範囲を同じにすれば，スケーリングの関数発生 Fx は 1 連でよい。PV と MV の内部変数の範囲を一致させるメリットは，内部変数のプロセスゲイン K がすべての PID ループで 1 となり，比例定数 K_P の設定が判り易くなることである。

ただし，このような設計では，工学値変換係数（特に手動の設定 MV）や PID パラメータ（特に MV 制限）や DA 変換に渡す Fx の設計が PID ループごとに変わり混乱をまねくおそれがある。特に，MV の直接設定（手動のこと）（リミットも）の単位をどう表現するかが悩ましいことになる。

さらに，Fx を減らしたいときは次図の構成でもよいが，PID ループごとに注意深く設計してほしい。

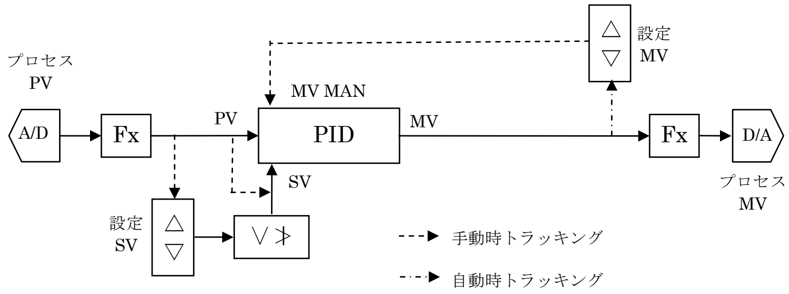

図　Fx の少ない構成

もし，PV と MV の内部変数の範囲を同じにする設計を採用するならば，MV の工学値の単位を PV のそれと同じにするという設計も可能である。つまり, MV のバルブ開度やヒーター出力の工学値の単位をパーセントとするのではなく，PV と同じ L/min や kW などとするのである。

このようにすれば，バルブ開度・流量特性やヒーター出力・温度特性の横軸と縦軸の単位は同じになり，ブロック図の中で工学値の単位が見かけ上統一されることになる。そして，非直線性補正も横軸と縦軸の単位は同じまま設計することになる。これはプロセスゲイン K が 1 に統一されたことを意味する。

その代償として，物理空間と内部変数の関係の設計が混乱しやすくなるので注意されたい。特に，MV の直接設定（手動のこと）（リミットも）の単位をどう表現するかが悩ましいことになる。単位変換のためにスケーリングするのは本末転倒になる。

非直線性補正が Fx の設計の混乱を招いているのである。念のためにもう一度言うが，非直線性補正がなくても PID 制御は正常に働くのである。