プロセス制御工学
6. PID制御

京都大学　加納　学

Division of Process Control & Process Systems Engineering
Department of Chemical Engineering, Kyoto University

manabu@cheme.kyoto-u.ac.jp
http://www-pse.cheme.kyoto-u.ac.jp/~kano/

PID制御の基礎

- 比例(P)動作
  偏差の大きさに応じて操作変数を調節する.

$$u(t) = K_P e(t)$$

- 積分(I)動作
  偏差が存在する限り操作変数を変化させ続ける.

$$u(t) = \frac{1}{T_I}\int_0^t e(\tau)d\tau$$

- 微分(D)動作
  偏差の変化速度に応じて操作変数を調節する.
  予測に基づいて制御を行う効果がある.

$$u(t) = T_D \frac{de(t)}{dt}$$

PID制御則

時間領域での表現

$$u(t) = K_P\left(e(t) + \frac{1}{T_I}\int_0^t e(\tau)d\tau + T_D\frac{de(t)}{dt} \right) + u_0$$

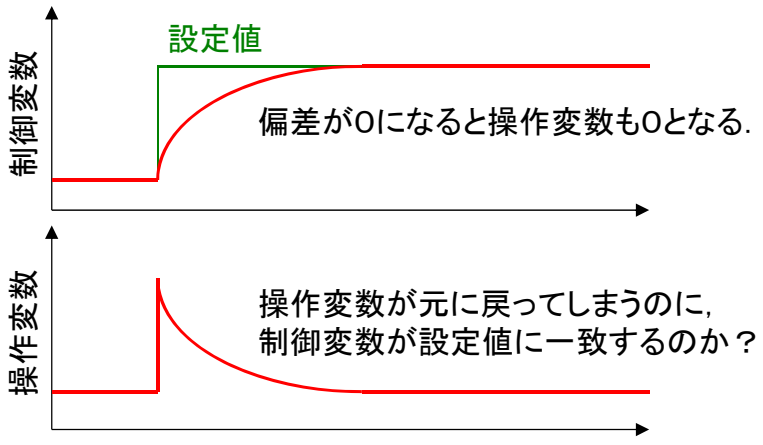
伝達関数による表現

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

K_P 比例ゲイン　T_I 積分時間　T_D 微分時間

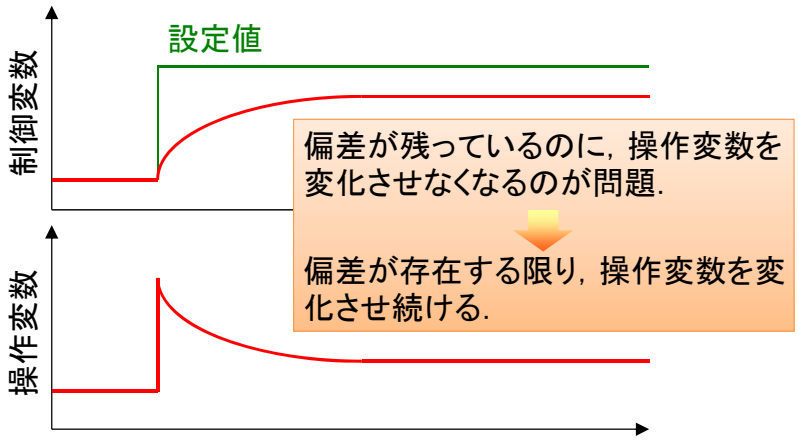
比例制御の仕組み

$$u(t) = K_P e(t)$$



比例制御の問題点と解決策

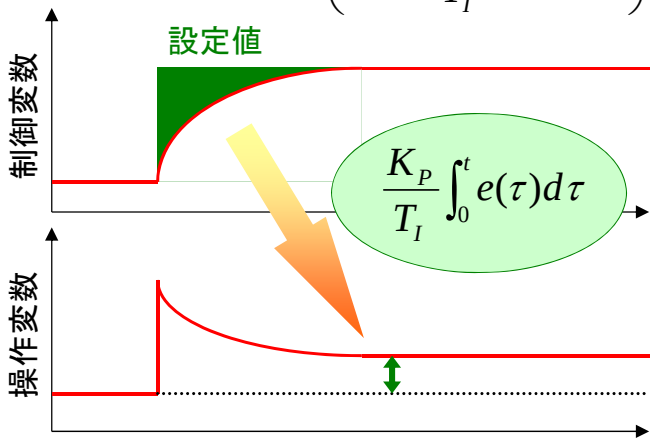
$$u(t) = K_P e(t)$$



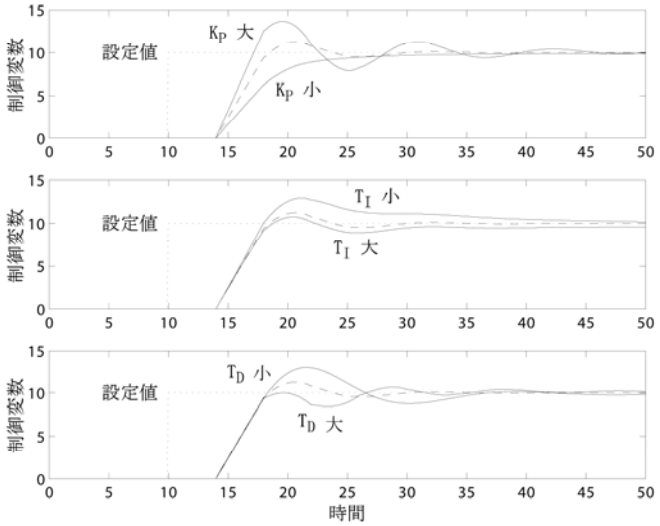
比例制御だけでは定常偏差が残ってしまう．

積分制御の仕組み

$$u(t) = K_P\left(e(t) + \frac{1}{T_I}\int_0^t e(\tau)d\tau \right)$$



PID制御パラメータの働き



PID制御パラメータと制御性能

	比例ゲインを増加 K_P	積分時間を減少 T_I
立上がり時間	短くなる	変わらない（短くなる）
行過ぎ量	大きくなる	大きくなる
整定時間	最小となる値がある	最小となる値がある

ZN 限界感度法

制御則	比例ゲイン K_P	積分時間 T_I	微分時間 T_D
P	$0.5K_C$	―	―
PI	$0.45K_C$	$0.833T_C$	―
PID	$0.6K_C$	$0.5T_C$	$0.125T_C$

K_C 限界感度
制御系が安定限界にあるとき，すなわち一定振幅の持続振動が起こるときの比例ゲイン

T_C 振動周期

ZN ステップ応答法

制御則	比例ゲイン K_P	積分時間 T_I	微分時間 T_D
P	T/KL	―	―
PI	$0.9\ T/KL$	$3.33L$	―
PID	$1.2\ T/KL$	$2L$	$0.5L$

プロセスの動特性が1次遅れ要素とむだ時間で表される場合

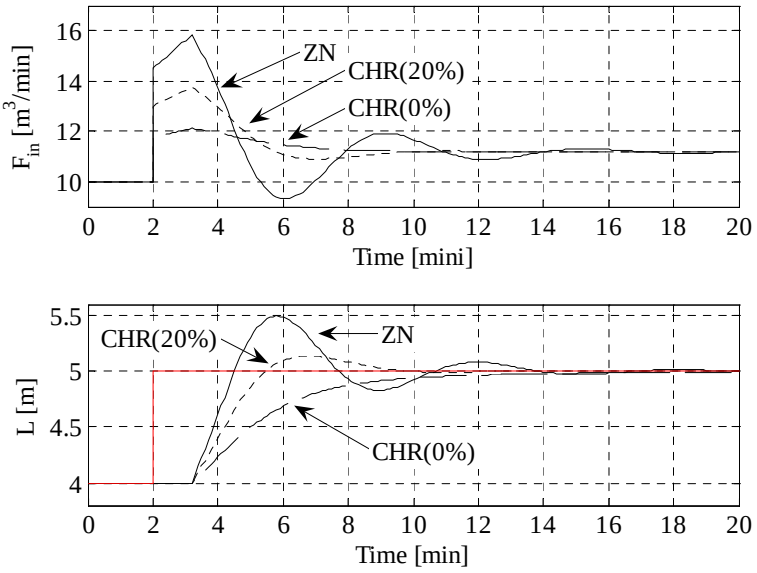
$$P(s) = \frac{K}{Ts+1} e^{-Ls}$$

CHR法

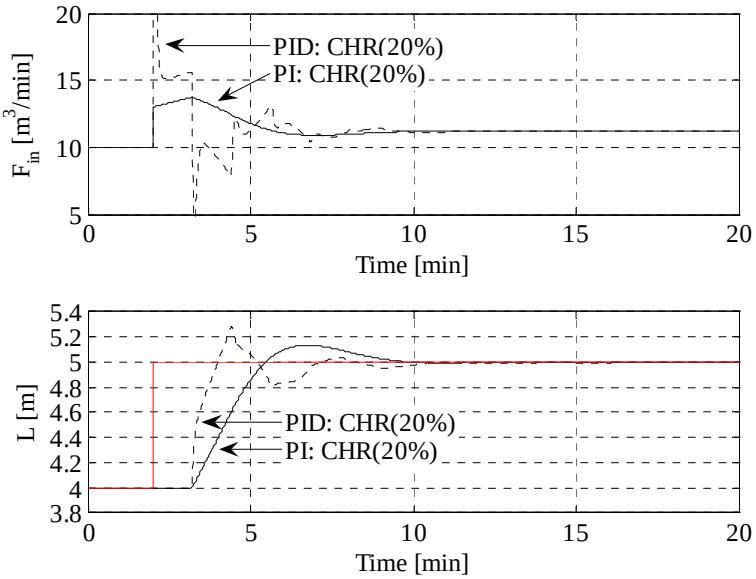
- Chien，Hrones，Reswickは，目標値と外乱のステップ状変化に対して，行過ぎ量を0%とする場合と20%とする場合の合計4通りの組み合わせを考え，調整方法を提案した．
- この調整方法は，提案者の名前にちなんでCHR法と呼ばれ，制御変数が定常値に到達するまでの時間を最小にすることを目的としている．

テキスト参照

調整方法の比較（PI制御）



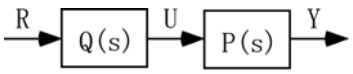
PI制御とPID制御の比較



内部モデル制御（IMC）法

- 設定値変更に対して理想的な開ループ制御を考える.
- コントローラQ(s)をプロセスP(s)の逆数として設計すると，制御変数を設定値に完全に一致させることができる.

$$Y = PQR = PP^{-1}R = R$$

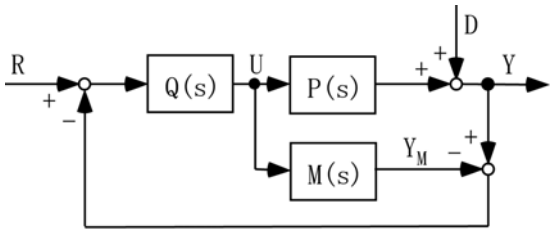


(a) 開ループ制御

- ただし，このままでは，外乱やモデル誤差（プロセスとモデルのずれ）が存在する場合に，制御変数を設定値に一致させることができない.

内部モデル制御(IMC)法

- プロセスP(s)とモデルM(s)を並列に配置し，それらの出力の差をコントローラに戻す．
- M=Pであり，かつ外乱が存在しなければ，このフィードバック制御系は理想的な開ループ制御系と等しくなる．



(b) 内部モデル制御 (IMC)

内部モデル制御(IMC)法

- 完全な制御を行うためには，IMCコントローラQ(s)をモデルM(s)の逆数として設計すればよい．しかし，現実には，モデルの逆数としてコントローラを設計できない．
- 例えば，プロセスがむだ時間を有する場合，むだ時間の逆数は未来の予測を意味するため，その実現は不可能である．
- そこで，IMCコントローラQ(s)にモデルの逆数をそのまま利用するのではなく，以下のような工夫を施す．
  - モデルの最小位相(逆数が不安定とならない)要素のみの逆数をとる．なお，逆数をとらない部分は全域通過フィルタとなるようにする．
  - 低域通過フィルタF(s)を用いる．

内部モデル制御（IMC）法

モデル $M(s) = M_M(s)e^{-Ls}$

IMCフィルタ $F(s) = \dfrac{1}{(\lambda s+1)^n}$

IMCコントローラ $Q(s) = F(s)M_M^{-1}(s)$

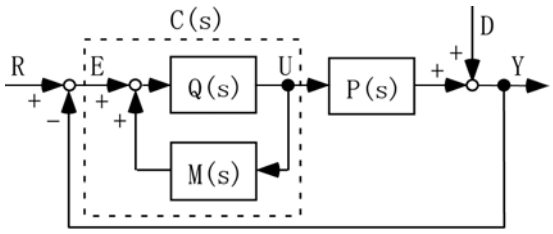
制御応答 $Y = PQR = \dfrac{1}{(\lambda s+1)^n}e^{-Ls}R$

- 設定値変更に対する制御変数の応答は，むだ時間だけ遅れるものの，プロセスには依存せず，フィルタ時定数λによって完全に決定される．

内部モデル制御（IMC）法

- 設定値変更に対する制御変数の応答は，むだ時間だけ遅れるものの，プロセスには依存せず，フィルタ時定数λによって完全に決定される．
- ステップ状設定値変更に対しては，制御変数は振動せずに設定値に漸近し，フィルタ時定数λを小さくすれば応答は速く，大きくすれば応答は遅くなる．
- 内部モデル制御を利用する場合には，モデルさえ与えられれば，後はフィルタ時定数λを調整するだけでよい．さらに，フィルタ時定数λが応答の速さに対応しているため，直感的に調整を行うことができる．

内部モデル制御(IMC)法



(c) IMCと等価なフィードバック制御

$$C = \frac{Q}{1 - QM}$$

内部モデル制御(IMC)法

モデル	比例ゲイン K_P	積分時間 T_I	微分時間 T_D
$\frac{K}{Ts+1}$	$\frac{T}{\lambda K}$	T	—
$\frac{K}{(T_1 s+1)(T_2 s+1)}$	$\frac{T_1+T_2}{\lambda K}$	T_1+T	$\frac{T_1 T_2}{T_1+T_2}$
$\frac{K}{\tau^2 s^2+2\zeta\tau s+1}$	$\frac{2\zeta\tau}{\lambda K}$	$2\zeta\tau$	$\frac{\tau}{2\zeta}$
$\frac{K}{s}$	$\frac{1}{\lambda K}$	—	—

モデル誤差を考慮した調整

- プロセスモデルが既知である場合には，計算機上で制御パラメータを変化させた制御シミュレーションを行い，最適な制御パラメータを求めることができる．
- モデル誤差の影響を考慮することを忘れてはならない．
- モデル誤差を考慮しないノミナルモデルに対して徹底的に調整された制御パラメータは，実プロセスの制御へ適用するには強すぎることが多く，制御系を不安定にしてしまう恐れもある．

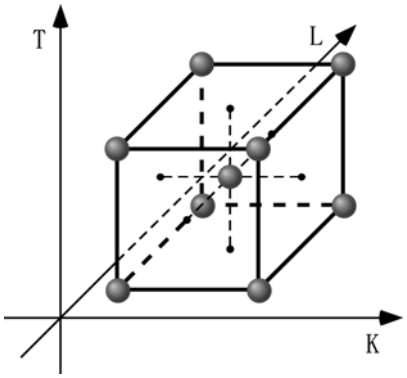
ロバスト性

- ロバスト安定性
  モデル誤差がある場合の制御系の安定性
- ロバスト性能
  モデル誤差がある場合の制御性能

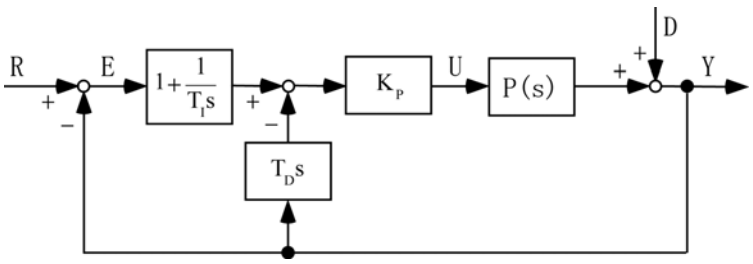
モデル誤差を考慮した調整法

- ノミナルモデル中の各パラメータの誤差範囲を見積り，各パラメータの最小値と最大値を決める．ノミナルモデルと合わせて，最大モデル誤差を考慮した複数個のモデルを用意する．
- 構築した複数のモデルを制御対象として制御シミュレーションを行い，制御性能が最も悪くなるモデルを用いた場合でも，許容できる範囲内の制御性能が実現できるように制御パラメータを調整する．

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

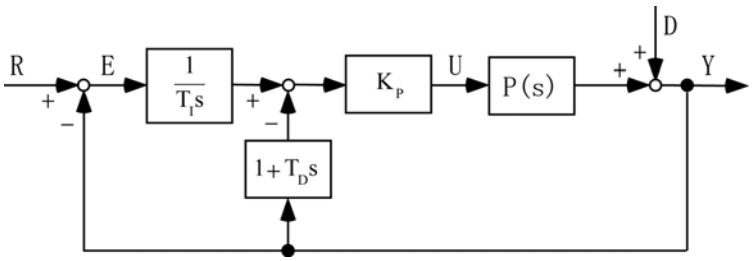


微分先行型PID制御（PI-D制御）



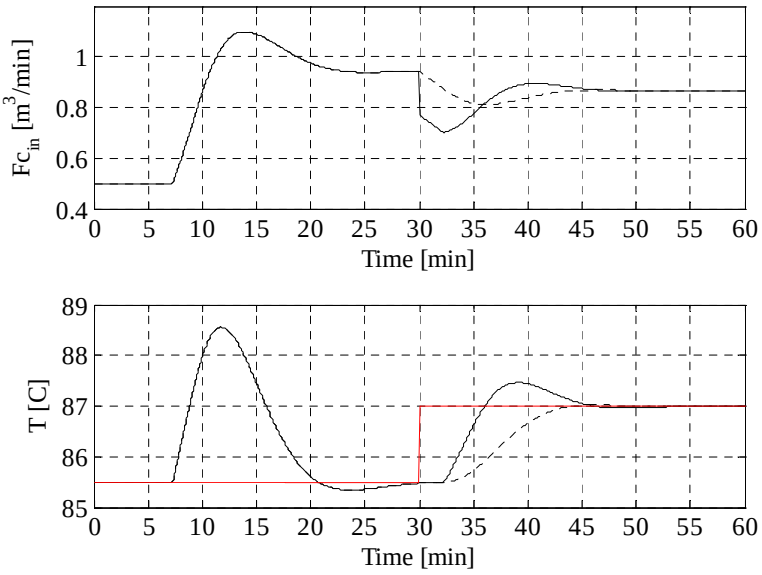
- PID制御を用いてステップ状の設定値変更を行うと，微分動作のために，操作変数はインパルス関数状に変化してしまう．
- このような急激な変化を避けるために，設定値を直接微分せず，制御変数のみに微分動作が働くようにする方法が考えられる．

I-PD制御



- 微分先行型PID制御は，ステップ状設定値変更時に操作変数の急激な変化を防ぐのに有効である．しかし，設定値のステップ状変化に対して操作変数がステップ状に変化することは避けられない．
- この操作変数のステップ状変化を避けるために，微分動作だけでなく比例動作も制御変数のみに働くようにする方法が考えられる．

PID制御とI-PD制御



PI-D制御とI-PD制御

27

＜微分先行型PID制御およびI-PD制御の特徴＞

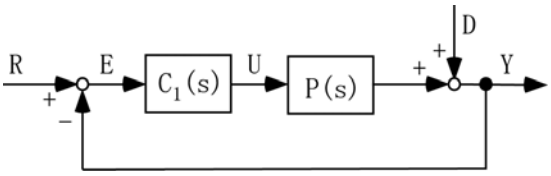
- 設定値変更に対する制御応答はPID制御と異なる.
- 外乱に対する制御応答はPID制御と全く同じである.
- 設定値追従性能と外乱抑制性能を独立に調整できる.

1自由度制御から2自由度制御へ

2自由度制御

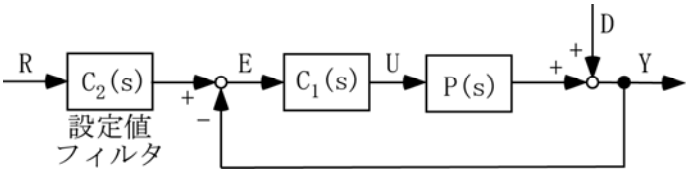
28

1自由度制御



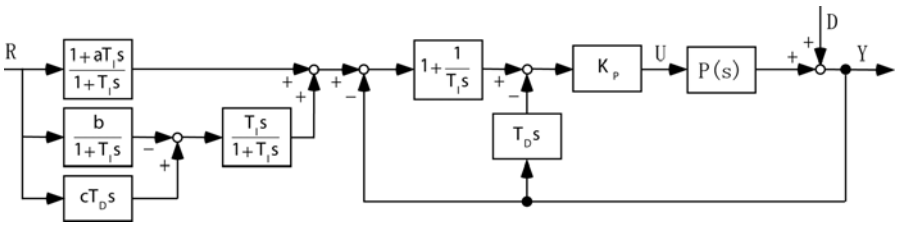
$$Y = \frac{PC_1}{1+PC_1}R \qquad Y = \frac{1}{1+PC_1}D$$

2自由度制御



$$Y = \frac{PC_1C_2}{1+PC_1}R \qquad Y = \frac{1}{1+PC_1}D$$

2自由度PID制御



不完全微分

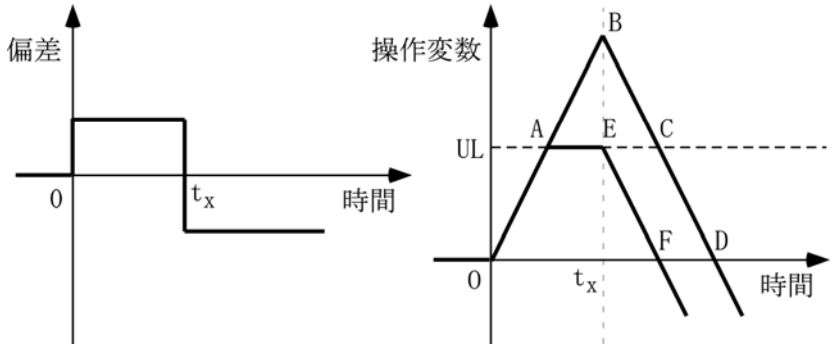
- 微分制御は偏差の傾きに応じて操作量を決定するため，測定ノイズが存在する場合には，微分制御が制御性能を低下させる原因ともなる．
- 偏差を直接微分するのではなく，1次遅れフィルタを用いることにより，測定ノイズの影響を軽減し，制御性能を改善する方法がある．

$$\frac{T_D s}{1+T_D s/\gamma}$$

- γは微分ゲインと呼ばれ，10前後の値に設定される．

リセットワインドアップ

31



- 操作変数が上下限制約にかかった場合，積分動作をオフにしなければ，制御が遅れ，応答は振動的になる.

おわり

32

- 宿題？