5　ＰＩＤ制御

5.1 制御系設計における PID 制御の位置づけ

制御系の実用設計とは，制御対象の事前知識の正確さと制御性能，および，制御装置の複雑さとの間のバランスを取ることである。制御対象の事前知識，すなわち，その数学モデルが完璧なら極配置あるいはLQ最適化に基づく状態フィードバックでよいし，検出ノイズをも考慮するならLQG制御を使えばよく，さらに対象モデルの変動も考慮に入れたければH_∞制御ということになる。これらは，制御対象の数学的モデルを出発点として設計を進める「モデルベース設計」である。この反対に，制御したい対象はそこにあるのだけれど，数学的モデルはよく分かっておらず，それを調べている時間もお金もないという場合がある。こんなときは，とりあえず制御対象と適当な補償装置を使って試行錯誤で制御パラメータを決めていくことになる。即物的かつ現実的な「現物ベース設計」である。その典型的な例が，本章で説明するPID制御である。

PID制御系のブロック図を図 5.1 に示す。このように簡単な構造でありながら，けっこう強力で，プロセス制御などに広く普及し活用されている。実際，実用されている制御法式のなかでPID制御は 84%を占めるという報告もある[1]。

PID制御の歴史は古い。その着想は 1922 年の論文に見られるという[2]。一方，「現代制御」は，宇宙ロケットの制御を目指して 1960 年前後から開発された（古いのに現代とは？）。さらに，「 H_∞最適制御理論」は，「現代制御」が数学モデルにたより過ぎることを反省して開発され，を「ポスト現代制御[3]」という。

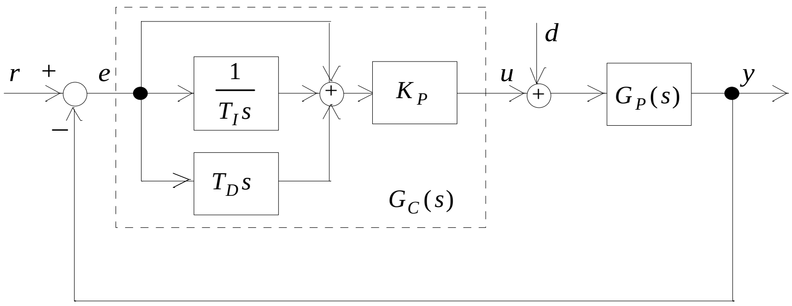

図 5.1　PID 制御の基本形

5.2 PID 制御の基本形と 3 動作の役割

図 5.1 のように構成したものが PID 制御の基本形である。PID 調節器の伝達関数 $G_C(s)$ は，

$$G_C(s) = K_P + K_I \frac{1}{s} + K_D s = K_P\left(1 + \frac{1}{T_I s} + T_D s\right) \quad \cdots\cdots (1.a)$$

$$K_I \equiv K_P / T_I, \quad K_D \equiv K_P T_D \quad \cdots\cdots (1.b)$$

(1.a)の第 2 辺の各項は，順に，比例(Proportional)，積分(Integral)，微分(Differential)の動作を行なっており，PID はこれらの頭文字を取ったものである。この意味で(1.a)の第 2 辺の表現が適しているが，慣習により(1.a)の第 3 辺のように表現されることが多い。

比例・積分・微分のそれぞれの動作の役割は次のようになっている。

- 比例動作(K_P)：偏差eのオフセット（定常偏差）を小さく抑える。
- 積分動作(K_I, K_P/T_I)：偏差eのオフセットを 0 にする。
- 微分動作(K_D, $K_P T_D$)：偏差eの振動（時間変化率）を小さく抑える。

演習問題 上の比例・積分・微分の 3 動作の役割を確認せよ。
（ヒント）図 5.1 から次式が得られる。

---

[1] 古田，富田：”先端制御技術の動向調査”，計測と制御，29 巻 10 号，pp.953-958, (1990)
[2] 須田：「PID制御」，p.7，システム制御情報ライブラリー6，システム制御情報学会，(1992.7)
[3] 「ポスト現代・・・」は，週刊誌の名前みたい。きっと，この次は「新潮制御」では？

$$e \to u:\quad u(s) = K_P(1+\frac{1}{T_I s}+T_D s)e(s) \quad \cdots\cdots(2)$$

$$u \to e:\quad e(s) \equiv r(s) - y(s) = r(s) - G_P(s)u(s) \quad \cdots\cdots(3)$$

(2)と(3)を連動して考えれば, 3 つの動作が分かる。たとえば, 積分動作について考える。仮に $e(s)$に正のオフセットがあれば，(2)の右辺第 2 項の積分器に偏差 $e(s)$が貯まり $u(s)$が増えていき（塵も積もれば山となる），(3)により $e(s)$が減り続ける。この動作は，(2)の積分器の入力 $e(s)$が 0 になるまで続く。

微分動作は次のように考える。仮に $e(s)$が時間とともに正に増加したとすれば，(2)の右辺第 3 項の微分器により $u(s)$が正に大きくなる。しかし，この $u(s)$は，(3)の第 3 辺のように逆符号で作用するため， $e(s)$の時間変化率を小さく抑え込むことになる。

◇

実際の応用では，PIDの 3 動作を全部使わなくても，1 動作ないし 2 動作の調節器（たとえばP動作，PI動作など）でもあまり効果が変わらないことがある。このような場合にPIDの 3 動作のうちどれを使えばよいかについての選択基準[1]を図 5.2 のグラフに示す。ただし，制御対象の伝達関数 $G_P(s)$ が，おおよそ，

$$G_P(s) = \varepsilon^{-Ds}\frac{1}{(1+T_1 s)(1+T_2 s)} \quad \cdots\cdots（4）$$

で近似できるものとする。

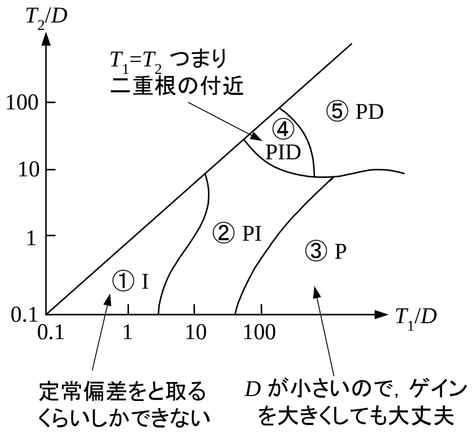

図 5.2　PID 調節器の動作の選択基準

5.3 PID 調節器のパラメータ調整法

実際の制御対象の特性（伝達関数）を正確に知らなくても，「現物ベース」でPID調節器のパラメータを調整する方法が，いろいろ知られている[2]。ここでは，ZieglerとNicholsによって提案された限界感度法と過渡応答法の要点のみを述べる。（詳しくは脚注の文献を参照）

まず，限界感度法を説明する。この方法では，調節器をP動作だけとし（T_Iを最大，T_Dを 0 とすればよい），比例ゲインK_Pを次第に大きくしていき，出力が一定振幅で振動を持続する（安定限界）ところでK_Pの増加を止める。このときの比例ゲインをK_C，振動周期をT_Cとすれば，調節器の各パラメータを表 5.1 のように決めればよい。

しかし，限界感度法には次の問題点がある。制御系を一旦安定限界までもっていくため試験中は通常の動作を行なえなくなる。また，操作量に飽和があると比例ゲインK_Pを大きくしても正確に安定限界を求めることができない。

[1] 茅：「自動制御工学」, p.149, 共立出版 (1976), または，堀，大西：「制御工学の基礎」, p.131, 丸善 (1997)

[2] 制御パラメータに決めかたのいろいろな表については, 須田:「PID制御」, 第2章, システム制御情報学会 (1992), 北森・真鍋の方法については，堀，大西：「制御工学の基礎」， pp.135-137, 丸善 (1997)。

一方，過渡応答法は，制御対象単体にステップ状の入力を加えたときのステップ応答から次のように制御パラメータを決める。多くの制御対象では，ステップ応答が図 5.3 のような曲線（これをシグモイド型曲線という）となる。この曲線の勾配が最も急なところに接線を引き，その勾配を R とし，この接線が横軸（時間軸）と交わる時刻を L とする。このとき，調節器の各パラメータを表 5.2 のように決めればよい。

なお，ここに示した制御パラメータの決めかたをすると，限界感度法および過渡応答法のどちらも，できあがった制御系は目標値のスッテプ変化に対する応答に 25%程度のオーバーシュートが生じる。これを変えるには，試行錯誤で制御パラメータを微調整することになる。微調整の方法は制御対象ごとに異なり，一般的なルールはないようである。

表 5.1　限界感度法による制御パラメータの決めかた

	比例ゲイン K_P	積分時間 T_I	微分時間 T_D
P 調節器	$0.5K_C$	―	―
PI 調節器	$0.45K_C$	$0.83T_C$	―
PID 調節器	$0.6K_C$	$0.5T_C$	$0.125T_C$

表 5.2　過渡応答法による制御パラメータの決めかた

	比例ゲイン K_P	積分時間 T_I	微分時間 T_D
P 調節器	$1/RL$	―	―
PI 調節器	$0.9/RL$	$3.33L$	―
PID 調節器	$1.2/RL$	$2L$	$0.5L$

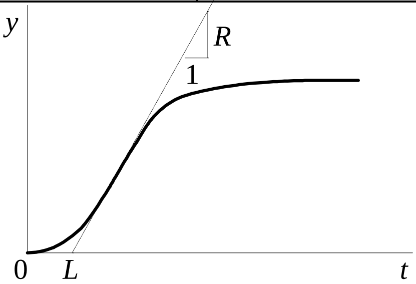

図 5.3　ステップ応答波形

5.4 PID 制御構造の変形

図 5.1 の PID 制御の基本形では問題を起こすことがある。これは，目標値がステップ状に変化した場合の急峻な変化が，D 動作で増幅されたり，P 動作をすどうしになったりして，制御対象に加えられるからである。対策は，D 動作，あるいは，PD 動作をフィードバックに移せばよい。D 動作をフィードバックに移したものは，微分先行型 PID 制御または PI-D 制御などと呼ばれる。PD 動作をフィードバックに移したものは，比例・微分先行型 PID 制御または I-PD 制御などと呼ばれる。

さらには，このように D 動作あるいは PD 動作をそっくりフィードバックに移すのではなく，一部を直列補償に残すことも考えられる。これは 2 自由度 PID 制御につながる考え方である。

これらについては，「須田：PID 制御，システム制御情報学会」の第 3・4 章に詳しく書いてあり，ここでは省略する。なお同書の第 5・6 章には，PID の自動調整法についても書いてある。