

制御の勘所

version 0.01 : 2004/04/16

近藤 正示

目次

| | | |
|-----|----------------------------|----|
| 1 | なぜフィードバック制御が使われるのか | 2 |
| 1.1 | 制御の目的とフィードバック | 2 |
| 1.2 | 制御系の感度 | 3 |
| 1.3 | フィードバック制御系の感度 | 5 |
| 1.4 | フィードフォワード制御系の感度 | 6 |
| 1.5 | 外乱および検出ノイズの影響と感度 | 7 |
| 2 | 動的システムの安定性と過渡応答 | 9 |
| 2.1 | 力学系の安定性 | 9 |
| 2.2 | 状態方程式とその解 | 11 |
| 2.3 | 解の挙動と極, そして安定性 | 11 |
| 2.4 | システム行列の固有値と伝達関数の極 | 12 |
| 2.5 | 伝達関数の極と零の性質 | 13 |
| 2.6 | ナイキストの安定判別法への補足 | 15 |
| 3 | 状態フィードバックによる極配置 | 17 |
| 3.1 | この節の位置づけ | 17 |
| 3.2 | 極配置の可能条件 | 18 |
| 3.3 | おまけ: 謎の数列 1, 2, 5, 10, ... | 18 |
| 4 | オブザーバによる状態変数の復元 | 19 |
| 4.1 | オブザーバとは? | 19 |

| | | |
|-----|-------------------------------------|----|
| 4.2 | 状態変数推定の問題設定 | 20 |
| 4.3 | 出力の微分を利用する方法 | 21 |
| 4.4 | 微分を避ける工夫 (シミュレータ) | 22 |
| 4.5 | 微分を避けて収束も速くする (全次元オブザーバ) | 24 |
| 4.6 | 検出できないものだけ推定 (低次元オブザーバ) | 25 |
| 4.7 | オブザーバを用いた状態フィードバック (分離定理) | 26 |
| 5 | P I D 制御 | 28 |
| 5.1 | 制御系設計における PID 制御の位置づけ | 28 |
| 5.2 | PID 制御の基本形と 3 動作の役割 | 29 |
| 5.3 | PID 調節器のパラメータ調整法 | 32 |
| 5.4 | PID 制御構造の変形 | 34 |

1 なぜフィードバック制御が使われるのか

1.1 制御の目的とフィードバック

フィードバックは、制御対象に関する我々の知識の不完全さに対処するただ一つの方策であり、同時に知識の不完全さを補い得る有効な方策である*1。これを以下に説明する。

制御とは、ある目的に適合するように必要な操作を対象に加えることである。制御の目的が目標値への追従ならば、図 1.1 のシステムにおいて制御系の目標値 r から制御量 y への伝達関数を 1 にすることである。しかし、こ

*1 計測自動制御学会：「自動制御ハンドブック」, p.465

れだけならフィードフォワード制御でも可能である（後で図 1.2(b) のところで述べる）。フィードバックの本質とは何だろうか？

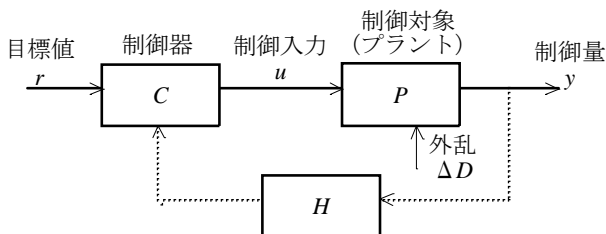


図 1.1 制御系の構成

そこで、「外乱があっても制御系の目標値から制御量への伝達関数を 1 にできるであろうか？」と考える。これは、フィードフォワード制御系では実現できない。なぜならば、外乱がいつ・どのくらいの大きさで入るか事前には分からないから、そのような（未知）外乱のことを制御器は知らず、対策できる筈がない。したがって、外乱の影響がそのまま制御量に出てしまい、伝達関数は 1 にならない。

このような未知外乱へ対応できるようにするのが、フィードバック制御である。すなわち、外乱が入れば制御量に影響が出る。それを見てから制御入力に修正を加えれば、制御量へ現れる外乱の影響を抑制できる。これがフィードバック制御の基本的な考え方である。事前に行うことができない外乱に対しては、フィードバックが唯一の方策である。しかし、制御入力の修正（対策）が後手に回り、安定性に問題を生じることもある。

1.2 制御系の感度

外乱の影響を一般的に考察するために、「感度」という考え方を導入しよう。図 1.1 の制御系において、外乱 ΔD が加えられたとき制御特性が ΔC だけ変動した場合について考える。我々は「感度」という言葉からおよそ次

のような感じをもっている。

- ΔD が小さくても、 ΔC が大きければ、感度は大きい。
- ΔD が大きくても、 ΔC が小さければ、感度は小さい。

このことから、感度を

$$\text{感度} = \Delta C / \Delta D \quad (1.1)$$

と定義してもよさそうに思われる。

しかし、うまくゆかない場合がある。それを明らかにするため、外乱と制御特性の関係を考えてみよう。制御系の特性を $f(a_1, a_2, a_3, \dots)$ とし、パラメータ a_1 が Δa_1 だけ変動した場合を考える。もし、 $a_1 \rightarrow a_1 + \Delta a_1$ となっても a_1 がもともと大きければ、その結果生じる特性の変動 Δf は相対的に小さくなる*2。したがって、上式の感度の定義では、パラメータの大小の影響を受け、特性変動の数量的な比較が難しいという欠点がある。

上記の欠点を除くため、 Δf と Δa_1 をそれぞれ f と a_1 で正規化して比を作る。すなわち、

$$S = \frac{\frac{\Delta f}{f}}{\frac{\Delta a_1}{a_1}} \quad (1.2)$$

を考える。感度 S が 1 のときは外乱の % 変動と制御特性の % 変動が等しく、 S が 1 以下のときは外乱に対して制御特性が変化しにくいことを意味する。上式の感度を、相対感度（または、パラメータ感度）という。さらに、(1.2) で $\Delta a_1 \rightarrow 0$ の場合を考えると、

$$S_0 = \frac{\frac{\partial f}{f}}{\frac{\partial a_1}{a_1}} = \frac{\partial(\ln f)}{\partial(\ln a_1)} \quad (1.3)$$

となる。これを微分感度という。

*2 たとえば、太平洋に塩 1kg（外乱）を投入しても、海水の塩辛さ（特性）はほとんど変わらない。

1.3 フィードバック制御系の感度

ここでは、図 1.2(a) のフィードバック制御系の相対感度および微分感度を求める。

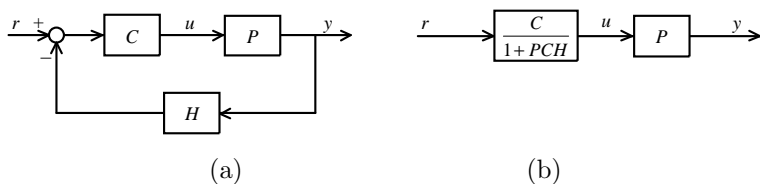


図 1.2

ただし、 $r \rightarrow y$ の伝達関数 T

$$T \equiv \frac{y}{r} = \frac{PC}{1 + PCH} \quad (1.4)$$

を制御系の特性とし、 $P \rightarrow P + \Delta P$ となったときの特性 T の変動を ΔT とする。

演習 図 1.2(a) に対して次の各式が成り立つことを示せ。

$$\Delta T = \frac{\Delta P}{1 + (P + \Delta P)CH} \cdot \frac{T}{P} \quad (1.5)$$

$$S \equiv \frac{\Delta T}{T} \bigg/ \frac{\Delta P}{P} = \frac{1}{1 + (P + \Delta P)CH} \quad (1.6)$$

$$S_0 = \frac{1}{1 + PCH} \quad (1.7)$$

*3

*3 演習のおわりを示す。ちゃんとやったらチェックしよう！

上の (1.7) から、微分感度 S_0 を小さくするにはループゲイン PCH を大きくすればよいことが分かる。また、(1.6) を変形して、

$$\frac{\Delta T}{T} = S \frac{\Delta P}{P} \quad (1.8)$$

であるから、相対感度 S の絶対値が 1 より小さければ、パラメータの % 変動より特性の % 変動を小さくできることが分かる。

1.4 フィードフォワード制御系の感度

ここでは、図 1.2(a) と同じ伝達関数をもつ図 1.2(b) のフィードフォワード制御系の感度を計算し、フィードバック制御系と比較する。

演習 以下の各項を確認せよ。

1. 図 1.2 の (a) と (b) では、 $r \rightarrow y$ の伝達関数 T が等しい。
2. 図 1.2(a) のフィードバック制御系で $P \rightarrow P + \Delta P$ となったとき、出力の変動 Δy_B は、次のようになる。

$$\Delta y_B = \frac{\Delta PC}{\{1 + (P + \Delta P)CH\}(1 + PCH)} r \quad (1.9)$$

3. 図 1.2(b) のフィードフォワード制御系では、プラントが $P \rightarrow P + \Delta P$ となってもコントローラの中の P は変化しないことに注意すれば、出力の変動 Δy_F が次式となる。

$$\Delta y_F = \frac{\Delta PC}{1 + PCH} r \quad (1.10)$$

上記の (1.9) と (1.10) を比較して、(1.6) を使えば、

$$\Delta y_B = S \Delta y_F \quad (1.11)$$

となる。この式から、閉ループ系の出力変動は、開ループ系の出力変動に相対感度 S を乗じたものである。したがって、 $|S| < 1$ とできれば、開ループ

系に比べ閉ループ系の出力変動を小さくできることが分かる。普通、フィードバック制御系でループゲイン PCH を大きくするのは、(1.7) から分かるように $|S| < 1$ としたいからである。

1.5 外乱および検出ノイズの影響と感度

制御系には、図 1.3 に示すように、外乱 d だけでなく検出ノイズ n がつきものである。

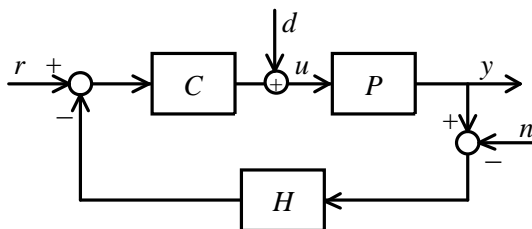


図 1.3

この時、外乱抑圧と検出ノイズ抑圧の間に (1.17) の関係があり、ジレンマに陥ることを以下に示そう。

演習 図 1.3 においてループゲイン $L = PCH$ とすれば、次式が成り立つことを示せ。

$$T_{ry} \equiv \frac{y}{r} = \frac{PC}{1+L} \quad (1.12)$$

$$T_{dy} \equiv \frac{y}{d} = \frac{P}{1+L} \quad (1.13)$$

$$T_{ny} \equiv \frac{y}{n} = \frac{L}{1+L} \quad (1.14)$$

$$\text{微分感度 } S_0 \text{ は (1.7) に注意すれば, } S_0 = \frac{1}{1+L} \quad (1.15)$$

$$T_{dy} = S_0 P \quad (1.16)$$

$$S_0 + T_{ny} = 1 \quad (1.17)$$

実は、上の最後の結果 (1.17) がジレンマを示しているのである。外乱の影響を小さくするため (1.13) の T_{dy} を小さくすると (1.16) から S_0 も小さくなる。すると、(1.17) からノイズの影響 T_{ny} が大きくなってしまふ。逆に、ノイズの影響 T_{ny} を小さくすれば、外乱の影響 T_{dy} が大きくなってしまふ。

このジレンマは、図 1.3 の系でフィードバック経路の H にローパス特性を持たせても逃れられない。確かに、 H にローパス特性を持たせれば高周波数領域では L が小さくなり、(1.14) からノイズの影響を小さくできる。ところが、それには代償がある。すなわち、 L が小さくなると (1.15) から S_0 が大きくなり、その結果 (1.16) から外乱の影響が大きくなってしまふのである。そこで、すべての周波数領域で外乱抑圧とノイズ抑圧の両方を同時に達成することをあきらめる。普通は低周波領域では外乱抑圧を重視し、高周波領域ではノイズ抑圧を重視するというようにする。

ここで、もっと進んだ制御理論すなわち H_∞ 制御と (1.17) との関係を示しただけ述べておこう。次のことを知れば難解な H_∞ 制御も少しは身近になるだろう(ならないか?)。 H_∞ 制御では (1.17) の S_0 を感度関数といい T_{ny} を相補感度関数という。後者に「相補」という接頭辞がつく理由は、(1.17)

のように二つの感度関数の和がいつも1であり、一方が大きくなれば他方が小さくなる、すなわち、相補的に変化するからである。上の例で制御対象の変動を考えるには、その影響を外乱に含めればよい。外から入る外乱、および、制御対象の変動による影響（制御システムの内乱？）の二つをひっくり返して外乱と考えることができる。したがって、感度関数 S_0 を小さくできれば、外乱が抑圧される同時に、制御対象変動の影響も抑圧される（ロバストになる）。しかし、センサノイズが増幅されるという代償はある。

当然のことであるが、良いセンサを使ってノイズを小さくするのが最も効果がある。センサのところで信号とノイズが混ざってしまうと、「外乱抑圧のことは少しあきらめて下さい」という請求書が後で届く。情報処理の技術に長けた人ならば、信号処理で信号とノイズを分離できる筈だから制御対象のモデルに基づく最適フィルタ（カルマン・フィルタ）を使えばよい、と言うかもしれない。しかし、実際の制御対象とそのモデルがずれた場合に問題が残る。このモデルミスマッチは、制御系のロバスト性の低下という代償を払わされることになる。センサノイズには、いつも代償を払えという請求書がついている考えてよい。

ならば、センサをなくしてしまえばよいではないか、という主張にも一理ある。センサ無しの制御、すなわち、フィードフォワード制御への道である。我々の知識が完全ならばそれもよい。ただし、そのために使う制御対象の性質・特性・事前知識には、環境変化および経年変化などの影響が少ない確固不変なものを慎重に選ぶ必要がある。

2 動的システムの安定性と過渡応答

2.1 力学系の安定性

直感が働きやすいという理由で、図 2.1 の曲面上のボール（力学系）の挙動を考えて、安定・不安定の意味を考えよう。

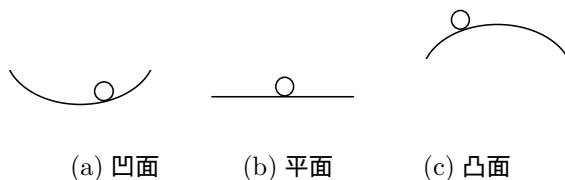


図 2.1 図 2.1 曲面上のボールの挙動

図 2.1 で、ボールと曲面の間に摩擦があるとすれば、次のことが分かる。

- 図 (a) ボールは底で止まる。=> この系は安定。
- 図 (b) ボールは動かない。=> 安定と覚えてよさそう。
- 図 (c) ボールは面から落ちる。=> この系は不安定。

これを一般化して安定・不安定を定義する。力学系が安定であるとは、最初に定められた領域 A にあったものが十分時間がたっても A 内に留まることを指す。不安定とは、十分時間がたつと A の外へ出て行き放しになることを指す。

さて、図 (a) の場合に最終的にボールは凹面の底で止まる理由はなにか。摩擦のためボールの力学的エネルギー（運動エネルギー + 位置エネルギー）が時間とともに減少し、ついには 0 になるからである*4。

実は、この場合の力学的エネルギーが、いわゆるリアプノフ関数に相当する。力学的エネルギー（リアプノフ関数）が時間とともに減少すれば、そのシステムは安定であると言って差し支えない。ていねいにいうと、リアプノフ関数とは、あるシステムで負にならずに、かつ、時間とともに発散しないものことである。したがって、リアプノフ関数が見つければそのシステムは安定である。これをリアプノフの安定理論*5という。なお、上の図 (b) に比

*4 ボールにエンジンは内蔵されていないとする。

*5 適当な制御の教科書を参照。たとえば、小郷, 美多:「システム制御理論入門」, p.79, 実

べて、図 (a) は安定の度合いが大きく必ず凹面の底でボールが止まるから漸近安定と呼ばれる。

2.2 状態方程式とその解

ここでは、状態微分方程式 (2.1) の解が (2.2) となることを示す。次節で時間的な挙動を調べるための準備である。

線形時不変なシステムの状態方程式は、

$$\left. \begin{aligned} \frac{d}{dt} \mathbf{x}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) \end{aligned} \right\} \quad (2.1)$$

であり、初期値を $\mathbf{x}(0)$ とすれば、その解 $\mathbf{x}(t)$ は

$$\mathbf{x}(t) = \varepsilon^{\mathbf{A}t} \mathbf{x}(0) + \int_0^t \varepsilon^{\mathbf{A}(t-\tau)} \mathbf{B}\mathbf{u}(\tau) d\tau \quad (2.2)$$

である*6。実際に (2.2) を時間で微分すれば (2.1) を満たすから、正しいことが分かる。なお、 $\varepsilon^{\mathbf{A}t}$ は状態推移行列と呼ばれいろいろな計算法がある。普通の指数関数と同じ計算ができると思ってよい。

2.3 解の挙動と極、そして安定性

(2.1) で制御入力 $\mathbf{u}(t) = 0$ の場合を自由システムといい、すべての初期値 $\mathbf{x}(0)$ に対して

$$\lim_{t \rightarrow \infty} \mathbf{x}(t) = 0 \quad (2.3)$$

となるとき漸近安定という。

教出版 (1981)。

*6 たとえば、小郷、美多：「システム制御理論入門」、p.63、実教出版 (1981)。

自由システムが漸近安定であるための必要十分条件は、 A のすべての固有値の実部が負：

$$\operatorname{Re}\{\lambda_i(A)\} < 0 \quad (\forall i) \quad (2.4)$$

である。なぜならば、 $u(t) = 0$ のときの解は (2.2) から求まり、さらに次のように変形できる*7。

$$\mathbf{x}(t) = e^{At}\mathbf{x}(0) = \mathbf{T} \begin{bmatrix} e^{\lambda_1 t} & & & \\ & e^{\lambda_2 t} & & \\ & & \ddots & \\ & & & e^{\lambda_n t} \end{bmatrix} \mathbf{T}^{-1}\mathbf{x}(0) \quad (2.5)$$

ここで、行列 T は A を (対角線上に各固有値が並ぶ) 対角行列に変換する一定な行列である。よって、次の必要十分関係が成り立つ。

漸近安定の定義 (2.3) が成り立つ

⇔ 上式第 3 辺が任意の初期値 $x(0)$ に対しても $t \rightarrow \infty$ のとき 0 となる

⇔ 第 3 辺の大きな行列の各項が $t \rightarrow \infty$ のときそれぞれ 0 となること

⇔ (2.4) が成り立つ。■

2.4 システム行列の固有値と伝達関数の極

ここでは、状態方程式から伝達関数を導き、伝達関数の極とシステム行列 A の固有値が同じであることを示す。

まず、状態方程式から伝達関数を導こう。(2.1) の第 1 式をラプラス変換すれば次式となる。

$$s\mathbf{X}(s) - \mathbf{x}(0) = \mathbf{A}\mathbf{X}(s) + \mathbf{B}U(s)$$

*7 たとえば、中野，美多：「制御基礎理論」，p.126，昭晃堂（1992.1），または，小郷，美多：「システム制御理論入門」，p.63，実教出版（1981）

これを $X(s)$ について解き，伝達関数を求めるため初期値 $x(0) = 0$ とし，(2.1) の第 2 式を使えば

$$Y(s) = C(sI - A)^{-1}BU(s)$$

となる。上式から， $U(s) \rightarrow Y(s)$ の伝達関数行列 $G(s)$ は次式となる。

$$G(s) = C(sI - A)^{-1}B = \frac{C \operatorname{adj}(sI - A)B}{\det(sI - A)} \quad (2.6)$$

次に，伝達関数の極はシステム行列 A の固有値と同じであることを示そう。伝達関数の極は，伝達関数の分母=0 の根であるから，(2.6) の第 3 辺の分母=0 とすれば，

$$\det(sI - A) = 0 \quad (2.7)$$

を満たす s が極である。しかるに，上式は A の固有値が満たす固有方程式である。したがって，極と固有値は同じものである。

演習「伝達関数が安定である条件はすべての極がラプラス平面の左半平面にあること」を示せ。

(ヒント) 伝達関数の極と A の固有値は同じもので，安定条件は (2.4) である。

2.5 伝達関数の極と零の性質

せっかく伝達関数 $G(s)$ の極が出てきたから，ついでに極と零の性質を説明する。(以下のことは安定性と関係はないが，知っていれば得である。) 1 入力 1 出力システムの伝達関数を

$$G(s) = \frac{N(s)}{D(s)} = \frac{\beta_n s^n + \beta_{n-1} s^{n-1} + \cdots + \beta_1 s^1 + \beta_0}{s^n + \alpha_{n-1} s^{n-1} + \cdots + \alpha_1 s^1 + \alpha_0}$$

とするとき，上式第 3 辺の分母または分子を 0 とした方程式の根が，それぞれ極 λ_i と零 z_i である。次の性質を示すことができる。

- 極 λ_i : $G(s)$ にインパルスを入力すると、出力に $\varepsilon^{\lambda_i t}$ という成分が出てくる。すなわち、極に対応する成分は、伝達関数をちょっとひっぱれば出てくる。
- 零 z_i : $\varepsilon^{z_i t}$ という信号を入力に加えても、それに対応する成分は出力に出ない。すなわち、零に対応する成分はブロックされて伝達関数を通らない。

さらに、フィードバックでは極は動かせるが、零は動かせない。したがって、零を動かすためには、新しいブロックを追加するしかない。

演習 上の極と零の性質を示せ。

(ヒント) 極の性質：簡単のため極はすべて実数で重複はないものとする
と、伝達関数は、次のように部分分数に分解できる。

$$Y(s) = G(s)U(s) = \left\{ \frac{K_1}{s - \lambda_1} + \frac{K_2}{s - \lambda_2} + \dots + \frac{K_n}{s - \lambda_n} \right\} U(s)$$

入力がインパルスなら $U(s) = 1$ として、上式を逆ラプラス変換すれば

$$y(t) = K_1 \varepsilon^{\lambda_1 t} + K_2 \varepsilon^{\lambda_2 t} + \dots + K_n \varepsilon^{\lambda_n t}$$

となって、上の極の性質が分かる。次に、零の性質：入力信号 $u(t) = \varepsilon^{z_i t}$ のラプラス変換は $U(s) = \frac{1}{s - z_i}$ だから、それを $G(s)$ に加えると出力は、

$$Y(s) = \frac{N(s)}{D(s)} \frac{1}{s - z_i} = \frac{(s - z_i) \tilde{N}(s)}{D(s)} \frac{1}{s - z_i} = \frac{\tilde{N}(s)}{D(s)}$$

となる。上式最後の分子分母に共通因子がなければ、 $\varepsilon^{z_i t}$ に対応する極がない。・・・(ヒントおわり)

演習フィードバックでは極は動かせるが、零は動かせない、ということを示せ。

(ヒント) $G(s) = N(s) / D(s)$ の出力を $-K$ でフィードバックした系の伝達関数は次式である。

$$\frac{Y(s)}{U(s)} = \frac{N(s)}{D(s) + KN(s)}$$

(ヒントおわり)

2.6 ナイキストの安定判別法への補足

まず、ナイキストの安定判別法を復習をして問題点を明らかにしておこう。

ナイキストの安定判別法とは、「図 2.2(a) の負帰還制御系は、 $\omega : 0 \rightarrow \infty$ としたときの開ループ伝達関数 $G(j\omega)H(j\omega)$ の軌跡が、図 2.2(b) の実線のように点 $(-1+0j)$ を左に見れば安定であり、破線のように右に見れば不安定である」というものである。

この証明は次の方針で行なう。次の関係が同値であることを複素積分の知識を使って示す*8。

「閉ループ伝達関数の分母式 $1 + G(s)H(s) = 0$ の根が複素左半平面にある (安定条件)」 \Leftrightarrow 「 $\omega : 0 \rightarrow \infty$ としたときナイキスト軌跡が点 $(-1+0j)$ を左に見る。」

上の方針は安定極の観点からは理解しやすいが、次の点がなんとなく分かりにくい。

- なぜ、閉ループでなく開ループの伝達関数の評価でよいのか？
- なぜ、点 $(-1 + 0j)$ が特別視されるのか？

演習 先に進む前に、次の詐欺師の例について考えよ。

ある詐欺師は、お金を貸すとそれを k (実数) 倍して返してくれる。この詐欺師とお金のやり取りしたとき損得の境目の条件は何か？何度も繰り返すと

*8 古典制御の教科書にはたいてい書いてある。たとえば、堀、大西：「制御工学の基礎」, pp.68-73, 丸善 (1997.9)

どうなるか？

上に示した分かりにくい点を解明するために，図 2.2(a) の負帰還閉ループ系の安定性を以下のようにして調べる。

目標入力 0 で，A 点に単位インパルスを入力したときの挙動を考えよう。単位インパルスは，すべての角周波数成分を含む正弦波の和（フーリエ級数）に展開できる。A 点に入力されたいろいろな角周波数成分の正弦波は，A 点 \rightarrow B 点の周波数応答特性によって，振幅がゲイン倍され，かつ，位相がずれて B 点に出力される。B 点の正弦波信号のうち，位相遅れが 180° となる角周波数成分に注目しよう。図 2.2(b) でいえば，ナイキスト軌跡が負の実軸（位相遅れが 180° ）を通るときの角周波数 ω_1, ω_2 である。位相が 180° ずれていると，正弦波信号の正負が反転する。図 2.2(a) では，B 点 \rightarrow A 点に戻すときにもう 1 回正負反転する。結局，A 点に入力された単位インパルスのうち上の角周波数の信号は，2 回正負反転されて（つまり正負は変わらずに）再び A 点に戻ってくる。このように，ループを一巡りだけしたときの様子を調べるには，閉ループではなく開ループ伝達関数を評価しておけばよい。

次に，ループを閉じたときのことを考える。（信号がぐるぐる巡るからといって，目を回さないように！）ここでも，位相遅れが 180° となる角周波数成分に注目する。この信号は A 点を出発して，振幅が開ループのゲイン倍されて戻ってくる。（ただし，上のことから正負は同じ。）この信号をさらにもう 1 回ループを巡らせ，さらにもう 1 回・・・と繰り返す。つまり，位相遅れが 180° となる角周波数の正弦波がぐるぐる巡る。（すなわち，上の演習の詐欺師とのやり取りと同じことをする。）このとき，開ループゲインが 1 より小さいと振幅が減少し閉ループ系は漸近安定になり，ゲインが 1 より大きいと振幅が次第に増大し閉ループ系は発散不安定になる。もし，ナイキスト軌跡が点 $(-1+0j)$ を通れば，位相遅れが 180° のとなる角周波数の開ループゲインが 1 である。よって，その角周波数の信号は増減せずに持続振

動することになる。すなわち，点 $(-1+0j)$ は，閉ループ系の安定と不安定の境目になっている。

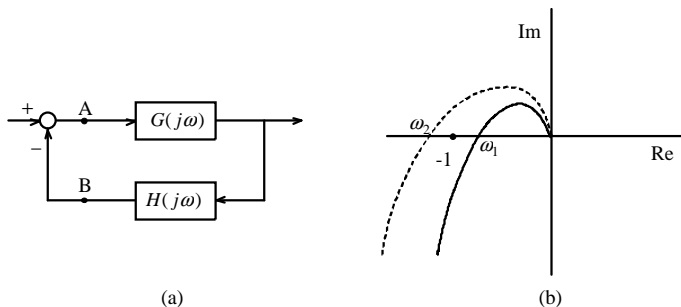


図 2.2

図 2.2 ナイキストの安定判別法

3 状態フィードバックによる極配置

3.1 この節の位置づけ

この節は，次節でオブザーバを導入することを合理化するために設けた。すなわち，前節で，極（固有値）がシステムの挙動を主に支配していることを示した。その極を自由自在に移動できる条件をこの節で示す。ただし，それを行なうには全状態変数を検出しなければならない。しかし，センサからは一部の状態変数しか検出できないことがある。したがって，そのような場合に，全状態変数を復元するオブザーバが必要となるのである。オブザーバについては，次節で述べる。

3.2 極配置の可能条件

次の線形時不変なシステム

$$\left. \begin{aligned} \frac{d}{dt} \mathbf{x}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) \end{aligned} \right\} \quad (3.1)$$

において，全状態 $\mathbf{x}(t)$ が検出できるとする。このとき，制御入力 $\mathbf{u}(t)$ を

$$\mathbf{u}(t) = -\mathbf{F}\mathbf{x}(t) \quad (3.2)$$

と定める。これを状態フィードバックという。状態フィードバックを行った閉ループ系の方程式は，(3.1) の第 1 式に (3.2) を代入して，

$$\frac{d}{dt} \mathbf{x}(t) = (\mathbf{A} - \mathbf{B}\mathbf{F})\mathbf{x}(t) \quad (3.3)$$

となる。このとき，閉ループ系の極，すなわち， $(\mathbf{A} - \mathbf{B}\mathbf{F})$ の固有値を任意の位置に移動できるための必要十分条件は，システム行列 \mathbf{A} が可制御であることである*⁹。したがって，状態フィードバックで極の移動を自由にするために全状態の検出が必要となる。

3.3 おまけ： 謎の数列 1, 2, 5, 10, ...

上の数列を，どこかでみたような気がしないだろうか？ 必ず出会っている。そう，1, 2, 5, 10, ... は，オシロスコープなどの測定器のレンジ設定の系列である。

問 測定器のレンジ設定の系列が，1, 2, 5, 10, ... となっている理由は？
(ヒント) それぞれの比を求めてみよ。

*⁹ たとえば，小郷，美多：「システム制御理論入門」，p.116-118，実教出版（1981）。

2倍ずつなら 1, 2, 4, 8, 16, 32, ... となるが、この系列は 10 進法に慣れた我々の日常生活にはなじまない。10 進 1 桁上がる毎に繰り返さないからである。この系列は、コンピュータと麻雀の御用達のような。(いわゆる「倍半分の法則^{*10}」はこの系列である。) 例は多くないが、1, 3, 10, ... という系列も使われている。たとえば、ものさしの長さである。この調子でゆくと次の系列は 1, 10, 100, ... で、10 進法ではこれでおわりである。

ところで、システムの設計でとりあえず数値を一つ決めないといけないことがある。たとえば、デジタル制御系のサンプリング周期などである。できるだけ短くすればよいというだけで、特別に明確な設計指針がない。こういときに役に立つのが、「10 倍則 (あるいは 1/10 則)」である。とりあえずサンプリング周期を欲しい応答時間のおよそ 1/10 にするのである。10 倍も離れていればおそらく大丈夫というのが根拠である。10bit ($2^{10} = 1024$) の A/D コンバータを使うのは、ほぼ 0.1% の精度だからである。8bit の A/D, D/A を使うのは安価だからである。明確な理由もないのに、12bit (=0.025%) はやりすぎと思うのは、貧乏人のひがみだろうか？

4 オブザーバによる状態変数の復元

4.1 オブザーバとは？

制御対象が可観測であれば状態フィードバック制御により制御系の極 (固有値) を任意に配置できる (ことが知られている)。ただし、この状態フィードバックを行うにはすべての状態変数を必要とする。しかし、現実には一部の状態変数が検出できないことがある。

「オブザーバ (観測器)」は、検出できるものを利用して状態変数の値を推定値として復元するものである。「オブザーバ」は、1964 年に D. G. Luenberger により提案された。「オブザーバ」とよく似ているものに「カル

*10 とりあえずパラメータを「エイヤー！」と倍・半分にしてみても、3 点で傾向をみることに。

マンフィルタ」がある。「カルマンフィルタ」は、1960年に R. E. Kalman によって提案された。両者の違いは、「オブザーバ」がノイズのない確定システムを対象とするのに対して、「カルマンフィルタ」は外乱および観測ノイズがある確率システムの出力からノイズを濾して状態を推定するものである。

本節では、検出できない状態変数を復元する方法に関して、順に、問題設定、出力微分による方法、シミュレータ、全次元オブザーバ、さらに、低次元オブザーバを概説し、最後に、制御と観測の分離定理を示す。

4.2 状態変数推定の問題設定

状態変数推定の問題とは、次のものである。「次式の線形なシステムが与えられたとき、

$$\frac{d}{dt} \mathbf{x}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) : \mathbf{A} (n \times n), \mathbf{B}(n \times m) \quad (4.1)$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) : \mathbf{C}(\ell \times n) \quad (4.2)$$

状態変数 $\mathbf{x}(t)$ を復元せよ。ただし、パラメータ \mathbf{A} , \mathbf{B} , \mathbf{C} と入出力 $\mathbf{u}(t)$, $\mathbf{y}(t)$ は既知であるとする。」

状態方程式 (4.1) の解は、初期値を

$$\mathbf{x}(0) = \mathbf{x}_0 \text{ (initial values)} \quad (4.3)$$

とすれば、次式で与えられる。

$$\mathbf{x}(t) = \varepsilon^{\mathbf{A}t} \mathbf{x}_0 + \int_0^t \varepsilon^{\mathbf{A}(t-\tau)} \mathbf{B}\mathbf{u}(\tau) d\tau \quad (4.4)$$

上式により、初期値 \mathbf{x}_0 が分かれば状態変数 $\mathbf{x}(t)$ を計算できる。ゆえに、状態変数推定の問題では、初期値 \mathbf{x}_0 をどのように推定するかがポイントとなる。

4.3 出力の微分を利用する方法

ここでは、出力を微分して状態変数を復元する方法を示す。出力 $y(t)$ を何回でも微分できるとすれば、状態変数 $x(t)$ を次のようにして復元できる。この方法は、理論的には大切である。なぜなら、可観測であるための必要十分条件が出てくるからである。しかし、微分を何回も行うとノイズを増幅するから実用的ではない。

まず、(4.2) を t で次々に微分して、(4.1) を利用すれば次のようになる。

$$\begin{aligned}y(t) &= Cx(t) \\ \dot{y}(t) &= C\dot{x}(t) = CAx(t) + CBu(t) \\ \ddot{y}(t) &= CA\dot{x}(t) + CB\dot{u}(t) = CA^2x(t) + CABu(t) + CB\dot{u}(t) \\ &\vdots \\ y^{(n-1)}(t) &= CA^{n-1}x(t) + CA^{n-2}x(t) + \cdots + CBu^{n-2}(t)\end{aligned}$$

これらをまとめて行列で表せば、

$$Y(t) = O x(t) + T U(t) \quad (4.5)$$

ただし、

$$Y(t) = \begin{bmatrix} y(t) \\ \dot{y}(t) \\ \ddot{y}(t) \\ \vdots \\ y^{(n-1)}(t) \end{bmatrix} \quad (4.6)$$

$$O = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix} \quad (4.7)$$

$$U(t) = \begin{bmatrix} \mathbf{u}(t) \\ \dot{\mathbf{u}}(t) \\ \ddot{\mathbf{u}}(t) \\ \vdots \\ \mathbf{u}^{(n-1)}(t) \end{bmatrix} \quad (4.8)$$

$$T = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ CB & 0 & \cdots & 0 & 0 \\ CAB & CB & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ CA^{n-2}B & CA^{n-3}B & \cdots & CB & 0 \end{bmatrix} \quad (4.9)$$

$x(t)$ を求めるため、(4.5) の両辺に左から O^T を乗じ、それを $x(t)$ について解けば、

$$\mathbf{x}(t) = (O^T O)^{-1} O^T (Y(t) - TU(t)) \quad (4.10)$$

となる。ただし、上式のようにできるのは、逆行列 $(O^T O)^{-1}$ が存在するときだけである。その逆行列が存在するための必要十分条件は、(4.7) の O は $n \times \ell$ 行列だから、

$$\text{rank} O = n \quad (4.11)$$

である。この条件式が成り立てば、(4.10) から $x(t)$ が求められ、その初期値 x_0 も求まる。したがって、(4.11) は、(4.1) と (4.2) からなるシステムが可観測であるための必要十分条件である。ここで、可観測とは、 $0 \sim t$ の間の入力 $u(t)$ と出力 $y(t)$ とから初期値 x_0 を一意に定められることをいう。

ただし、(4.6) のように出力の微分を何回も行うとノイズを増幅するから、上に示した方法は実用的ではない。

4.4 微分を避ける工夫 (シミュレータ)

前節のアプローチは、(4.1) のシステムの入力側から逆にたどって状態変数を求めようとしたから、微分操作が必要になった。それでは、(4.1) のシ

ステムの入力側から順にたどって状態変数を求めれば微分操作を避けられるのではないかと考えられる。これが以下に示すシミュレータの発想である。(図 4.1 を参照のこと。)

まず，入力 $u(t)$ から状態変数 $x(t)$ を復元するため，(4.1) を参考にして次のモデル

$$\frac{d}{dt}\hat{x}(t) = \hat{A}\hat{x}(t) + \hat{B}u(t) \quad (4.12)$$

をつくる。ただし，上付きハットは推定値を示す。初期値は分からないから次のように

$$\hat{x}(0) = \hat{x}_0 \quad (4.13)$$

適当におく。もし， $\hat{A} = A$ ， $\hat{B} = B$ ならば，(4.1) と (4.12) は全く同じシステムであり，入力も同じだから， $\hat{x}(t)$ と $x(t)$ とは同じになりそうである。このようにして状態変数を推定するものをシミュレータという。

しかし，シミュレータでは，(4.13) のように初期値を適当においたから，初期値の誤差が状態変数推定値の誤差にどのように影響するかを調べておく必要がある。それを調べるために， $\hat{A} = A$ ， $\hat{B} = B$ として (4.12) から (4.1) を引けば，

$$\frac{d}{dt}(\hat{x}(t) - x(t)) = A(\hat{x}(t) - x(t)) \quad (4.14)$$

となる。ここで，推定誤差 $e(t)$ を

$$e(t) \equiv \hat{x}(t) - x(t) \quad (4.15)$$

で定義すれば，(4.14) は

$$\frac{d}{dt}e(t) = Ae(t) \quad (4.16)$$

のように書ける。(4.16) の解は，

$$e(t) = \varepsilon^{At}e(0) \quad (4.17)$$

である。よって、状態変数の真値と推定値の関係は、(4.17) と (4.15) から次のようになる。

$$x(t) = \hat{x}(t) - \varepsilon^{At}(\hat{x}(0) - x(0)) \quad (4.18)$$

上式において、 A が漸近安定なら t が大きくなれば $\varepsilon^{At} \rightarrow 0$ となり、上式右辺第 2 項の初期誤差が消えて $\hat{x}(t) \rightarrow x(t)$ となることが分かる。ただし、この収束の速さは A の固有値に依存し、シミュレータではその収束の速さを設計者が自由に決めることができない。なお、シミュレータのブロック図は、後で示す図 4.1 において $K = 0$ としたものである。

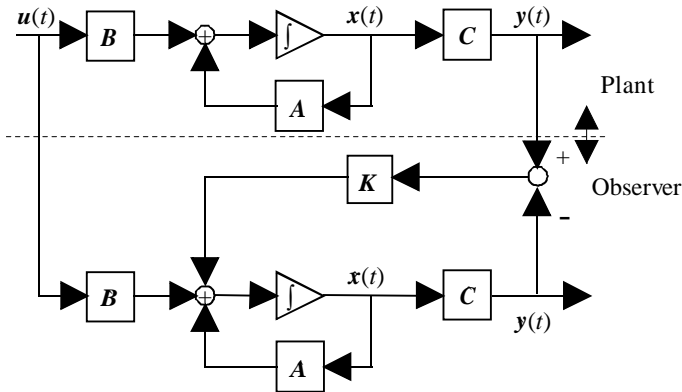


図 4.1 全状態オブザーバの構成

図 4.1 全次元オブザーバの構成

4.5 微分を避けて収束も速くする (全次元オブザーバ)

前節では入力 $u(t)$ だけを使ったが、まだ検出値 $y(t)$ は使っていない。この検出値 $y(t)$ を使って修正をすると状態変数の推定誤差の収束の速さを変えることができる。これが、オブザーバの発想である。

すなわち, (4.12) に検出値 $y(t)$ と推定値 $\hat{y}(t) = C\hat{x}(t)$ の差による修正項を加えて

$$\frac{d}{dt}\hat{x}(t) = \hat{A}\hat{x}(t) + \hat{B}\hat{u}(t) + K(y(t) - C\hat{x}(t)) \quad (4.19)$$

とする。これを全状態オブザーバといい, その構造は図 4.1 のようになっている。

推定誤差の挙動を調べるため, $\hat{A} = A, \hat{B} = B$ として, (4.19) から (4.1) を引いて, (4.2) および (4.15) を使えば,

$$\frac{d}{dt}e(t) = (A - KC)e(t) \quad (4.20)$$

となる。上式は, (4.16) と違い, 修正ゲイン K によって誤差の収束を調整できることが分かる。

ここで, K によって $A - KC$ の固有値をどのくらい変えられるかについては, 制御理論により次のことが知られている。すなわち, (C, A) が可観測であれば, $A - KC$ の固有値を任意の位置に配置でき, 逆も成り立つ。

以上により, (C, A) が可観測であれば, オブザーバにより微分操作を避け, かつ, 初期値誤差の収束を任意に指定できることが分かった。

4.6 検出できないものだけ推定 (低次元オブザーバ)

n 個の状態変数のうち m 個が出力 $y(t)$ として検出できるなら, $y(t)$ はそのまま使用し, 検出できない残りの $n - m$ 個の状態変数だけを復元してもよいと考えられる。これが, 低次元オブザーバの発想である。

まず, 状態変数 $x(t)$ を次のように分割して考える。

$$\hat{x}(t) = \begin{bmatrix} y(t) \\ \hat{z}(t) \end{bmatrix} \quad \begin{array}{l} (m)\text{-th order} \\ (n - m)\text{-th order} \end{array} \quad (4.21)$$

上の分割を考慮して, (4.1) を次のように分割する。

$$\frac{d}{dt} \begin{bmatrix} y(t) \\ \hat{z}(t) \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} y(t) \\ \hat{z}(t) \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u(t) \quad (4.22)$$

上式の 2 行目で $z(t)$ の予測をして, 1 行目で誤差を修正するようにする。すなわち,

$$\begin{aligned} \frac{d}{dt} \hat{z}(t) = & A_{22} \hat{z}(t) + A_{21} y(t) + B_2 u(t) \quad \text{estimation} \\ & + K (\dot{y}(t) - A_{11} y(t) - A_{12} \hat{z}(t) - B_1 u(t)) \quad \text{correction} \end{aligned} \quad (4.23)$$

とする。上式右辺の $\dot{y}(t)$ は, 実際には微分しなくてもよい。それを示すため, 上式を次のように変形する。

$$\frac{d}{dt} \hat{z}(t) = (A_{22} - K A_{12}) \hat{z}(t) + (A_{21} - K A_{11}) y(t) + (B_2 - K B_1) u(t) + K \dot{y}(t) \quad (4.24)$$

上式に従って図 4.2 のようなブロック図を描いてみれば, 右辺最終項の $\dot{y}(t)$ は, 実際に微分しなくてもよいことが分かる。

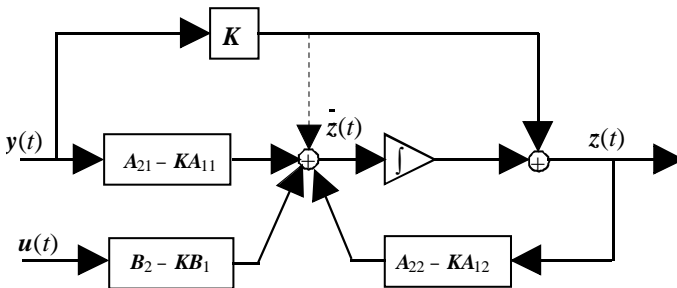


図 4.2 低次元オブザーバの構成

4.7 オブザーバを用いた状態フィードバック (分離定理)

オブザーバで復元した状態変数を使えば, 状態フィードバック系を構成できる。このとき, 制御器の挙動とオブザーバの挙動とが, 相互に絡み合って

複雑になるようなことはないだろうか？このことについては、オブザーバの収束と状態フィードバックの収束を、それぞれ独立に設計できることが、以下のようにして分かる。

まず、実際のシステムのパラメータ (C, A, B) とオブザーバに使うパラメータ $(\hat{C}, \hat{A}, \hat{B})$ は、一致しているものと仮定でき、さらに、(4.1) で与えられるシステムは可制御、可観測であると仮定する。このとき、次の状態フィードバックを考える。

$$u(t) = F\hat{x}(t) + Gv(t) \quad (4.25)$$

この式を、(4.1) と (4.19) に代入すれば、それぞれ

$$\frac{d}{dt}x(t) = Ax(t) + BF\hat{x}(t) + BGv(t) \quad (4.26)$$

$$\frac{d}{dt}\hat{x}(t) = KCx(t) + (A - KC)\hat{x}(t) + BF\hat{x}(t) + BGv(t) \quad (4.27)$$

となる。ここで、(4.26) は (4.15) を使って $\hat{x}(t)$ を消去すれば、

$$\frac{d}{dt}x(t) = (A + BF)x(t) + BF e(t) + BGv(t) \quad (4.28)$$

となる。一方、(4.27) から (4.26) を引いて、(4.15) を使って整理すれば、

$$\frac{d}{dt}e(t) = (A - KC)e(t) \quad (4.29)$$

となる。(4.28) と (4.29) をまとめて書くと、

$$\frac{d}{dt} \begin{bmatrix} x(t) \\ e(t) \end{bmatrix} = \begin{bmatrix} A + BF & BF \\ 0 & A - KC \end{bmatrix} \begin{bmatrix} x(t) \\ e(t) \end{bmatrix} + \begin{bmatrix} BG \\ 0 \end{bmatrix} v(t) \quad (4.30)$$

という拡大系の状態方程式が得られる。上式の右辺第一項の係数行列は上三角行列である。上三角行列の固有値は対角要素のブロック行列の固有値に等しいことが、線形代数で知られている。したがって、拡大系の固有値は、 $A + BF$ および $A - KC$ の固有値に等しい。

ここで、 $A + BF$ の固有値は、(4.28) から状態フィードバック制御系の収束性を決める固有値であるが、状態フィードバックゲイン F で設定することができる。一方、 $A - KC$ の固有値は、(4.29) からオブザーバの収束性を決める固有値であるが、オブザーバの修正ゲイン K で設定することができる。したがって、状態フィードバック制御系の収束性、および、オブザーバの収束性を、それぞれ独立に設定することができることが分かる。このことを、制御と観測の分離定理という。

制御と観測の分離定理が成り立つと、制御器の設計とオブザーバの設計を別々に行ってよくなる。すなわち、制御系全体の設計という大きな問題を、制御器の設計とオブザーバの設計という二つの小さな問題に分割できる。一つの大きな問題より、二つの小さな問題のほうが取り扱いが易しい。

ただし、制御と観測の分離定理が成り立つのは、

- 制御対象のシステムが可制御・可観測であり、かつ、
- 実際のシステムのパラメータとオブザーバのパラメータが一致する

場合である。パラメータが不一致の場合には、フィードバック制御系の挙動とオブザーバの挙動が相互に干渉し、全体の挙動は大変複雑になる。

演習パラメータ (C, A, B) と $(\hat{C}, \hat{A}, \hat{B})$ が等しくないとき、(4.30) はどうなるか？

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、達成すべき制御性能、および、制御装置の複雑さなどの間のバランスを取ることである。制御対象の事前知識（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。検出ノイズをも考慮するならLQG制御

を使えばよい。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、制御したい対象は目の前にあるのだけれど、数学的モデルが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に適切な補償装置を取り付け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、ここで説明する PID 制御である。

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^{*11}。一方、「現代制御^{*12}」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。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た。これを「ポスト現代制御^{*13}」ともいう。

PID 制御系は、図 5.1 に示すように簡単な構造でありながら強力で、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方式のなかで PID 制御は 84% を占めるといふ報告もある^{*14}。

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、

$$G_C(s) = K_P + K_I \frac{1}{s} + K_D s = K_P \left(1 + \frac{1}{T_I s} + T_D s \right) \quad (5.1a)$$

$$K_I \equiv K_P / T_I, \quad K_D \equiv K_P T_D \quad (5.1b)$$

^{*11} 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)

^{*12} 古いのに現代とは？

^{*13} 「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？

^{*14} 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)

(5.1a) の第 2 辺の各項は，順に，比例 (Proportional)，積分 (Integral)，微分 (Differential) の動作を行なっている。PID はこれらの頭文字を取ったものである。この意味で (5.1a) の第 2 辺の表現が適しているが，慣習により (5.1a) の第 3 辺のように表現されることが多い。

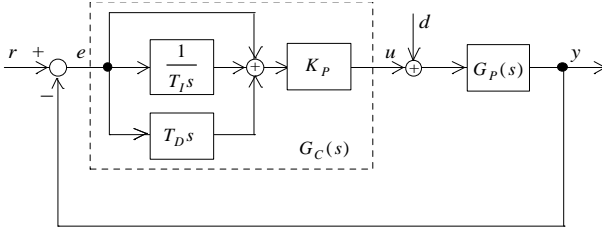


図 5.1 PID 制御の基本形

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。

$$e \rightarrow u : u(s) = K_P \left(1 + \frac{1}{T_I s} + T_D s \right) e(s) \quad (5.2)$$

$$u \rightarrow e : e(s) \equiv r(s) - y(s) = r(s) - G_P(s)u(s) \quad (5.3)$$

(5.2) と (5.3) 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，(5.2) 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)，(5.3) により $e(s)$ が減り続ける。この動作は，(5.2) 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、(5.2)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、(5.3)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。(ヒントおわり)

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^{*15}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、

$$G_P(s) = \varepsilon^{-Ds} \frac{1}{(1 + T_1s)(1 + T_2s)} \quad (5.4)$$

で近似できるものとする。

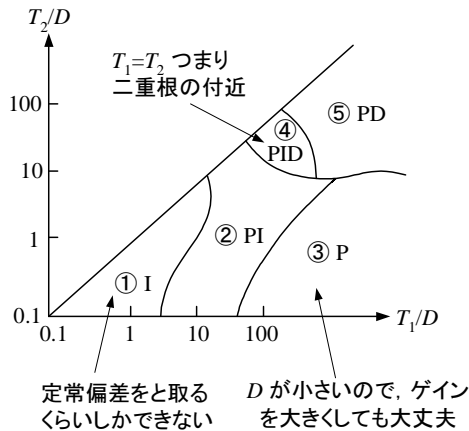


図 5.2 PID 調節器の動作の選択基準

^{*15} 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)

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*16。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。

表 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 | $0.6K_C$ | $0.5T_C$ | $0.125T_C$ |

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。

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

*16 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）。

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。

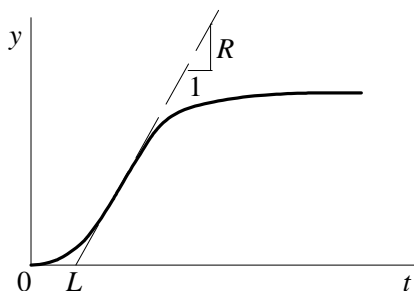


図 5.3 ステップ応答波形

表 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 | $1.2/RL$ | $2L$ | $0.5L$ |

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テッ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。

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。

