ディジタル制御システム

Digital Control System

平成 28 年

辻　峰男

まえがき

本テキストではディジタル制御とコンピュータ（マイコン）制御システムの入門として，まず理解しておかなくてはならないこと及び実際の応用で重要と思われることを簡単な例を用いてできるだけ丁寧に述べている。

第 1 章では，最も簡単な場合を例に取り，ディジタル制御の概要を説明する。ディジタル制御システムではサンプリング周期ごとに制御対象の状態量を検出し，それを基に必要な入力量をマイコンで演算して制御対象に加える。ディジタル制御システムを数学的に表現すると，数列あるいは差分方程式になる。微分方程式で記述される連続系の場合にはラプラス変換を使ってシステムのブロック線図を作るが，数列あるいは差分方程式の場合には z 変換を用いる。第 1 章では z 変換に関しても説明する。

第 2 章では, z 変換により得られる伝達関数を用いてブロック線図を作る方法を説明する。一般に制御対象は連続系であるから，加える操作量は連続量でなくてはならない。このために，マイコンで演算される値をサンプリング周期の間一定に保つ零次ホールドの機能が必要になる。零次ホールドと制御対象の伝達関数(ラプラス変換表示)をまとめて z 変換することでブロック線図を得ることができる。従来のテキストでは，偏差を求めるところまで連続系で考えられているものが少なくないが，偏差はマイコンの中で演算されるのが一般的であろう。また，マイコンで演算した後のサンプラーも不要と思われる。

第 3 章は，ディジタル制御器の話である。多くのディジタル制御法が提案されているが，本稿では実際に良く利用されている PI 制御や補償要素について述べる。PI 制御や補償要素は，連続系として微分方程式や伝達関数（ラプラス変換）で表わされているものが判り易く，良く知られている。これらをディジタル制御に利用するには差分方程式に離散化（ディジタル再設計という）しなくてはならない。離散化すればプログラムして製品に組み込むことができる。離散化の方法は多く存在するが，良く利用される方法を説明する。さらに実際の応用では入力量は無制限に大きくできないので，リミッタが必要となる。本章ではリミッタに関しても詳しく述べている。

第 4 章では，ブロック線図で表されたディジタル制御系の安定判別法を説明する。 z の双一次変換を行うと連続系の安定解析でよく知られたラウスの方法が利用できて便利である。また，ディジタル制御特有の制御法であるデッドビート制御についても例題を通して説明する。

第 5 章では周波数応答の計算法を述べる。連続系のラプラス変換で表した伝達関数の周波数応答は $s = j\omega$ と置くことで得られるが， z 変換で表した伝達関数の周波数応答は $z = e^{j\omega T}$ と置くことで求まる。周波数応答を利用したナイキストの安定判別についても説明する。また， z 変換による周波数応答とフーリエ変換の関係について述べる。

第 6 章では，z 変換を使わないで差分方程式を用いて直接解析する方法を説明する。行列を用いた連立差分方程式(状態方程式)によってディジタル制御システムを記述すると時間応答の計算（解析プログラムの作成）が容易で，系行列の固有値から安定判別ができる。状態方程式は z 変換によるブロック線図に比べ複雑なシステムにも適用しやすく，パソコンによる解析に向いていると言えよう。

第 7 章は，実際に多く用いられているマイコンによるディジタル制御システムの話である。マイコンの歴史を築き現在でも一部に利用されている 8 ビット CPU Z80 を取り上げる。原理を理解するには，構造が簡単な方がわかり易く，Z80 を使ってマイコンの仕組みを説明する。また割り込み周期がサンプリング周期になることが多いので，その理解はディジタル制御に欠かせない。最後に最も簡単な電流制御の場合を例にとりマイコン制御システムのハードウェアとソフトウェアの概要を述べる。

ディジタル制御器の設計法としては，ラプラス変換により連続系として設計し離散化する方法（ディジタル再設計）とディジタル制御系として直接設計する方法がある。ディジタル再設計の場合，ディジタル制御系としての安定解析，ステップ応答，周波数応答などの計算で性能をチェックしておく必要がある。サンプリング周期が十分短かければ，連続系の特性に近づくと考えられるが，サンプリング周期が短くとれない場合や制御ゲインが大きい場合には特に注意を要する。ディジタル制御系として直接設計する場合は，伝達関数の周波数特性を考えることにより，ナイキスト線図やボード線図が描け，連続系に似た設計が可能となる。また，行列を用いた時間領域での設計法もある。実際の設計では，ディジタル再設計の方が広く普及しているようである。これはラプラス変換した伝達関数がこれまでの経験もあって判りやすいこととコンピュータの高速処理によりサンプリング周期が短く選べるケースが多いことが要因であろう。

制御法に関しては連続系と同様，PID 制御や各種補償要素が良く用いられている。最適制御，状態観測器（オブザーバ），適応制御，H_∞制御（ロバスト制御），パラメータ同定なども高性能化を図るために使われているが，本テキストでは述べておらず，これらについては文献(5),(17)などを参照して欲しい。これらにも連続系としての設計法とディジタル制御系としての設計法が考えられるが，状態観測器の一種であるカルマンフィルタや最小 2 乗法によるパラメータ同定は離散時間系として取り扱われている。適用が難しい制御法にチャレンジして，性能の向上を図ることこそ技術者魂であろう。

また，本テキストでは A/D 変換器のビット数で決る量子化誤差（一種の雑音と考えられる）に関しては議論していない。A/D 変換器のビット数が小さいと制御ゲインが大きくできないことが考えられる。ただし，通常は，量子化誤差の影響が出ないようにビット数を選ぶので，問題となることは少ないように思われる。

目次

第１章　ディジタル制御とは？

1.1　数列

ディジタル制御の話を始める前に**数列**(sequence, progression)の復習をしよう。**公比**(common ratio) p の**等比数列**(geometric sequence)

$$x_{k+1} = p\,x_k \qquad (k = 0,1,2,\cdots) \tag{1-1}$$

を考える。このとき，

$$x_k = p^k\,x_0 \tag{1-2}$$

と表せる。$k \to \infty$ のとき，x_k がどうなるか考えると，以下のことが判る。

$|p| < 1$　　のとき　$x_k \to 0$　　　（収束，安定）

$p = 1$　　のとき　$x_k \to x_0$　　（収束，安定限界）
$p = -1$　　のとき　$x_k \to \pm x_0$　　（振動，安定限界）

$|p| > 1$　　のとき　$x_k \to \infty$　　（発散，不安定）

ディジタル制御系は数列として捉(とら)えることができる。すなわち，k 番目の値と $(k+1)$ 番目の値がどのように関係するかが問題となる。この場合，値が無限大になることは避けなければならない。制御ではこれを不安定という。従って，(1-1)の安定条件は，

$$|p| < 1 \qquad (\ -1 < p < 1\ ) \tag{1-3}$$

である。

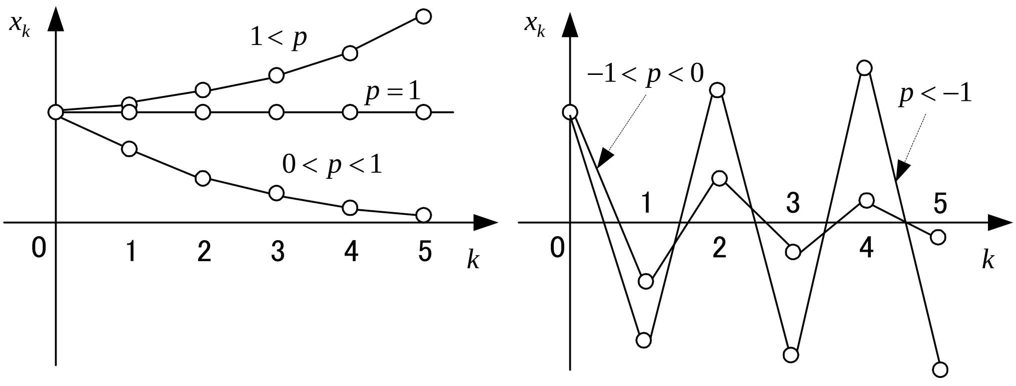


図 1−1　いろいろの公比に対する等比数列の変化

今度は次式で表現される数列（一般には**差分方程式**(difference equation)という）を考える。

$$x_{k+1} = p\,x_k + q \qquad (k = 0,1,2,3\cdots) \tag{1-4}$$

ここで，p, q は一定値とする。$k = 0,1,2,3\cdots$ を代入して，次式が得られる。

$$x_1 = p\,x_0 + q$$

$$x_2 = p\,x_1 + q = p^2 x_0 + pq + q$$

$$x_3 = p\,x_2 + q = p^3 x_0 + p^2 q + pq + q$$

同様にして

$$x_k = p^k x_0 + (p^{k-1} + p^{k-2} + \cdots + p + 1)q = p^k x_0 + \frac{1-p^k}{1-p}q \qquad (p \neq 1) \tag{1-5}$$

(1-5)で，$k \to \infty$ のとき，$|p| < 1$ であれば,

$$x_{\infty} = \frac{q}{1-p} \tag{1-6}$$

となる。これは，(1-4)で，$x_{k+1} = x_k = x_{\infty}$ と置くことでも得られる。従って，(1-4)においても，x_{∞} が一定値に収束する（安定である）ための条件は，p の絶対値が 1 以下であることである。実際のシステムでは，q は入力または指令値に相当する。

○　**等比数列の公式**

$$a + ar + ar^2 + \cdots + ar^n = \frac{a(1-r^{n+1})}{1-r} \qquad r \neq 1 \qquad n+1\text{は項数}$$

$$a + ar + ar^2 + ar^3 + \cdots = \frac{a}{1-r} \qquad |r| < 1 \qquad \text{無限等比級数}$$

1.2　z 変換

いま，ある数列（≡は定義を表す）

$$\{x(k)\} \equiv \{x(0), x(1), x(2), \cdots\} \qquad (k = 0,1,2,\cdots) \qquad 『定義』 \tag{1-7}$$

に対し，複素数 z を用いて，数列 $\{x(k)\}$ の**z変換**(z-transform)を次式で定義する。

$$Z\{x(k)\} \equiv X(z) \equiv x(0) + x(1)z^{-1} + x(2)z^{-2} + x(3)z^{-3} + \cdots \tag{1-8}$$

複素数 z は z 変換 $X(\mathrm{z})$ が収束するように選ぶものとする。定義は覚えるしかない。定義が

判らないとか難しいと言って逃げていたらダメです。数列の最初から順番に $1,\ z^{-1}, z^{-2}, z^{-3}\cdots$ を掛けて加えるだけである。この定義を用いると，数列

$$\{x(k+1)\} \equiv \{x(1), x(2), x(3), \cdots\} \qquad (k=0,1,2,\cdots) \quad 『定義』 \tag{1-9}$$

に対する z 変換は次式で与えられる。（初項が何から始まるかがポイント！）

$$Z\{x(k+1)\} = x(1) + x(2)z^{-1} + x(3)z^{-2} + x(4)z^{-3} + \cdots \tag{1-10}$$

(1-8)を用いると，

$$Z\{\,x(k+1)\} = z\,X(z) - z\,x(0) \tag{1-11}$$

が成り立つ。一方，数列

$$\{x(k-1)\} \equiv \{x(-1), x(0), x(1), x(2)\cdots\} \qquad (k=0,1,2,\cdots)『定義』 \tag{1-12}$$

に対する z 変換は，$x(k)=0\ :\ k<0$ と定義し（ディジタル制御では負の時間の信号は 0 と考えるということ），$\{x(k-1)\} \equiv \{0, x(0), x(1), x(2)\cdots\}$　だから，次式で与えられる。

$$Z\{x(k-1)\} = x(0)z^{-1} + x(1)z^{-2} + x(2)z^{-3} + \cdots = z^{-1}X(z) \tag{1-13}$$

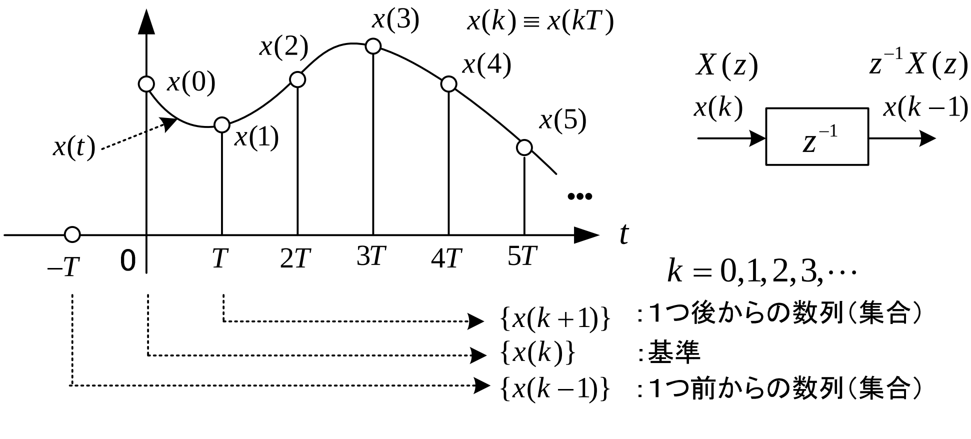


図 1-2　サンプリングした値と数列の関係

制御においては，周期 T ごとにデータが検出されたり出力されたりする。この周期 T を**サンプリング周期**(sampling period)と呼ぶ。連続量 $x(t)$ に対し，$t=kT$ における値を $x(k)$ と書く。この関係を図 1-2 に示す。ディジタル制御系は差分方程式で表され，時間的に

$k=0,1,2,\cdots$ と変化すると考えてよく，　$\{x(k-1)\}$ の k 番目の要素 $x(k-1)$ は $\{x(k)\}$ の k 番目の要素 $x(k)$ より 1 サンプリング周期分データが古く，それが(1-13)になるので，z^{-1} には**1サンプリング周期遅らせる（古いデータにする）**という意味があることが判る。

z 変換は，一つの値に対する変換ではなく，数列全体すなわち全ての値に対する変換であることを忘れないでおこう。だから $x(1)$ の z 変換と言っても意味がない。

[例題 1-1]　値が全て 1 である数列　$\{x(k)\}=\{1,\ 1,\ 1,\ \cdots\}$　の z 変換を求めよ。

(解)　定義より，

$$Z\{x(k)\}=1+z^{-1}+z^{-2}+z^{-3}+\cdots=\frac{1}{1-z^{-1}}=\frac{z}{z-1}$$

但し，収束するように z を選ぶものとする。ラプラス変換の場合と同様，z 変換を用いるときに，z 変換の収束条件を気にすることはまずないだろう。

＊　$\{x(k)\}=\{1,\ 0,\ 0,\ 0,\cdots\}$ の z 変換は 1，　$\{x(k)\}=\{0,\ 1,\ 0,\ 0,\cdots\}$ の z 変換は z^{-1} である。従って，1 の z 変換と言う表現はあいまいである。通常，これは $\{x(k)\}=\{1,1,1,\cdots\}$ の z 変換を意味するのだろうが。

[例題 1-2]　数列

$$\{x(k)\}=\{1,\ p,\ p^2,\ p^3,\cdots\}$$

すなわち，$x(k)=p^k$　$(k=0,1,2,\cdots)$ の z 変換を求めよ。

(解)　定義より，

$$Z\{x(k)\}=1+pz^{-1}+p^2z^{-2}+p^3z^{-3}+\cdots=\frac{1}{1-pz^{-1}}=\frac{z}{z-p}$$

（注）この式を，p で微分すると，$x(k)=k\,p^{k-1}$　$(k=0,1,2,\cdots)$ の z 変換が得られる。

$$Z\{0,1,2p,3p^2,4p^3,\cdots\}=z^{-1}+2p\,z^{-2}+3p^{\,2}z^{-3}+\cdots=\frac{z}{(z-p)^2}$$

[例題 1-3]　数列

$\{x(k+2)\}=\{x(2),\ x(3),\ x(4),\ \cdots\}$　$(k=0,1,2,\cdots)$

の z 変換を $X(z)$ を用いて表せ。

(解)　定義より，

$$Z\{x(k+2)\}=x(2)+x(3)z^{-1}+x(4)z^{-2}+x(5)z^{-3}+\cdots=z^2X(z)-z^2x(0)-zx(1)$$

1.3　z 変換による差分方程式の解法

　次に示すかんたんな**差分方程式**(difference equation)を z 変換で解いてみよう。

$$x(k+1) = p\,x(k) + q \qquad (k = 0,1,2,\cdots) \tag{1-14}$$

なお，　$x_{k+1} = p\,x_k + q$　のように書くこともある。上式は以下の意味がある。

$$x(1) = p\,x(0) + q,\ \ x(2) = p\,x(1) + q,\ \ x(3) = p\,x(2) + q, \cdots \tag{1-15}$$

z 変換の定義より，

$$Z\{x(k+1)\} = x(1) + x(2)z^{-1} + x(3)z^{-2} + x(4)z^{-3} + \cdots$$

$$Z\{x(k)\} \equiv X(z) \equiv x(0) + x(1)z^{-1} + x(2)z^{-2} + x(3)z^{-3} + \cdots$$

$$Z\{q,q,q,\cdots\} = q + q\,z^{-1} + q\,z^{-2} + q\,z^{-3} + \cdots$$

これらの式を用いると，(1-15)より，**数列**（成分ではなく）の z 変換には以下の関係が導ける。

$$Z\{x(k+1)\} = p\,Z\{x(k)\} + Z\{q,q,q,\cdots\} \tag{1-16}$$

(1-11)の公式を用いると，

$$z\,X(z) - z\,x(0) = p\,X(z) + \frac{q\,z}{z-1}$$

従って，この式から $X(z)$ を求め，**逆z変換**(inverse z-transform)すると $x(k)$ が求まる。この解き方は後述の例題 1-4 で詳しく述べる。 $q = 0$ の場合には，次式となる。

$$X(z) = \frac{z}{z-p}\,x(0)$$

$x(0)$ は定数だから，例題 1-2 を逆に考えて，一般解は次式となる。

$$x(k) = p^k\ x(0) \qquad (k = 0,1,2,\cdots)$$

　当然の結果であるが，z 変換で求めたことに意味がある。また，ある成分について書かれた（1-14）の関係式が，数列全体に及ぶ z 変換に対しても同じ関係式で成り立つ（(1-16）式）ことを良く考えて欲しい。$1, z^{-1}, z^{-2}, z^{-3} \cdots$ と順番に掛けて加えたものが z 変換だから，任意の順番のところで成り立つと z 変換したものにも成り立つということを。

[例題 1-4] 次の差分方程式を解け。

$$x(k+2)-x(k+1)+0.24x(k)=1,\quad x(0)=1,\ x(1)=2 \quad (\ k=0,1,2,\cdots) \qquad ①$$

(解)　①を数列で表して，それを z 変換すると，次式を得る。

$$Z\{x(k+2)\}-Z\{x(k+1)\}+0.24Z\{x(k)\}=Z\{1,1,1,\cdots\} \qquad ②$$

故に，
$$z^2X(z)-z^2x(0)-zx(1)-\{zX(z)-zx(0)\}+0.24X(z)$$
$$=(z^2-z+0.24)X(z)-z^2-z$$
$$=\frac{z}{z-1} \qquad ③$$

よって，
$$X(z)=\frac{z^3}{(z-0.6)(z-0.4)(z-1)} \qquad ④$$

$X(z)/z$ を作り（これがコツ），これを**部分分数展開**すると次式となる。

$$\frac{X(z)}{z}=\frac{z^2}{(z-0.6)(z-0.4)(z-1)}=\frac{25}{6}\frac{1}{z-1}-\frac{9}{2}\frac{1}{z-0.6}+\frac{4}{3}\frac{1}{z-0.4} \qquad ⑤$$

両辺に z をかけて，

$$X(z)=\frac{25}{6}\frac{z}{z-1}-\frac{9}{2}\frac{z}{z-0.6}+\frac{4}{3}\frac{z}{z-0.4} \qquad ⑥$$

逆変換して，

$$x(k)=\frac{25}{6}-\frac{9}{2}(0.6)^k+\frac{4}{3}(0.4)^k \quad (k=0,1,2,\cdots) \qquad ⑦$$

＊なお，⑤式の $\dfrac{a}{z-p}$ 係数 a は，重根でなければ，以下の公式が使える。

$$a=(z-p)\left.\frac{X(z)}{z}\right|_{z=p} \qquad ⑧$$

[問題 1-1]　　z 変換を用いて，次の差分方程式を解け。

$x(k+2)+4x(k+1)+3x(k)=1,\quad x(0)=0,\ x(1)=0\quad (k=0,1,2,\cdots)$

（答）
$$Z\{x(k+2)\}+4Z\{x(k+1)\}+3Z\{x(k)\}=Z\{1,1,1,\cdots\}$$
$$z^2X(z)-z^2x(0)-zx(1)+4\{zX(z)-zx(0)\}+3X(z)=\frac{z}{z-1}$$
$$\frac{X(z)}{z}=\frac{1}{8}\frac{1}{z-1}-\frac{1}{4}\frac{1}{z+1}+\frac{1}{8}\frac{1}{z+3}$$
$$x(k)=\frac{1}{8}-\frac{1}{4}(-1)^k+\frac{1}{8}(-3)^k\quad (k=0,1,2,\cdots)$$

[問題 1-2]　z 変換を用いて，次の差分方程式を解け。ただし，$p \neq 1$とする。

$x(k+1) = p\,x(k) + q$,　$x(0) = 1$　$(k = 0,1,2,\cdots)$

（答）　$Z\{x(k+1)\} - p\,Z\{x(k)\} = Z\{q,q,q,\cdots\}$

$$zX(z) - zx(0) - p\,X(z) = q\frac{z}{z-1}$$

$$\frac{X(z)}{z} = \frac{z-1+q}{(z-p)(z-1)} \qquad x(k) = \frac{q}{1-p} + \frac{1-p-q}{1-p}p^k$$

z 変換に関する重要な定理を以下にまとめておく。

1.　**定義**　$Z\{x(k)\} \equiv X(z) \equiv x(0) + x(1)z^{-1} + x(2)z^{-2} + x(3)z^{-3} + \cdots$

2.　$Z\{x(k-1)\} = z^{-1}X(z)$　　$(k = 0,1,2,\cdots)$

$Z\{x(k+1)\} = z\,X(z) - z\,x(0)$　　$(k = 0,1,2,\cdots)$

$Z\{x(k+2)\} = z^2X(z) - z^2x(0) - zx(1)$　　$(k = 0,1,2,\cdots)$

3. **数列の z 変換**

数列	$X(z)$
$\{x(k)\} = \{1, 0, 0, 0, \cdots\}$ $k = 0$ のとき 1	1
$\{x(k)\} = \{0, 1, 0, 0, \cdots\}$ $k = 1$ のとき 1	z^{-1}
$\{x(k)\} = \{0, \cdots, 0, 1, 0, \cdots, 0\}$ $k = j$ のとき 1	z^{-j}
$\{x(k)\} = \{1, 1, 1, \cdots\}$ 一般項　$x(k) = 1$	$\dfrac{z}{z-1}$
$\{x(k)\} = \{1, p, p^2, p^3, \cdots\}$ 一般項　$x(k) = p^k$	$\dfrac{z}{z-p}$

4. **線形性**(linearity)

2 つの数列$\{x(k)\}$, $\{y(k)\}$ $(k = 0,1,2,\cdots)$の線形結合の z 変換

$$Z\{a\,x(k) + b\,y(k)\} = a\,Z\{x(k)\} + b\,Z\{y(k)\} \qquad (1\text{-}17)$$

ただし，a, b は定数。

5. **初期値の定理**(initial value theorem)

$$x(0) = \lim_{z\to\infty} X(z) \qquad (1\text{-}18)$$

6.　**最終値の定理**(final value theorem)

$$x(\infty)=\lim_{z\to 1}(z-1)X(z) \tag{1-19}$$

初期値の定理は定義より明らかである。

<u>線形性の証明</u>

$$\{x(k)\}\equiv\left\{x(0),\ x(1),x(2),x(3),\cdots\right\},\quad \{y(k)\}\equiv\left\{y(0),\ y(1),y(2),y(3),\cdots\right\}$$

及び

$$Z\{x(k)\}=x(0)+x(1)z^{-1}+x(2)z^{-2}+x(3)z^{-3}+\cdots$$
$$Z\{y(k)\}=y(0)+y(1)z^{-1}+y(2)z^{-2}+y(3)z^{-3}+\cdots$$

と定義されている。よって，

$$\{a\,x(k)+b\,y(k)\}=\{a\,x(0)+b\,y(0),a\,x(1)+b\,y(1),a\,x(2)+b\,y(2),\cdots\}$$

であるから，

$$\begin{aligned}Z\{a\,x(k)+b\,y(k)\}&=a\,x(0)+b\,y(0)+(a\,x(0)+b\,y(0))z^{-1}\\&\quad+(a\,x(2)+b\,y(2))z^{-2}+(a\,x(3)+b\,y(3))z^{-3}+\cdots\\&=a\,Z\{x(k)\}+b\,Z\{y(k)\}\end{aligned}$$

<u>最終値の定理の証明</u>

$$Z\{x(k)\}=\sum_{k=0}^{\infty}x(k)\,z^{-k}=X(z) \qquad ①$$

である。また，

$$Z\{x(k+1)\}=\sum_{k=0}^{\infty}x(k+1)\,z^{-k}=z\,X(z)-z\,x(0) \qquad ②$$

②－①　より

$$\sum_{k=0}^{\infty}x(k+1)\,z^{-k}-\sum_{k=0}^{\infty}x(k)\,z^{-k}=(z-1)X(z)-z\,x(0) \qquad ③$$

両辺を $z\to 1$ とすると，③は次式となる。

$$左辺＝\lim_{z\to 1}\left(\sum_{k=0}^{\infty}x(k+1)\,z^{-k}-\sum_{k=0}^{\infty}x(k)\,z^{-k}\right)＝\sum_{k=0}^{\infty}x(k+1)-\sum_{k=0}^{\infty}x(k)$$

$$＝\lim_{k\to\infty}\left(\sum_{k=0}^{k}x(k+1)-\sum_{k=0}^{k}x(k)\right)＝\lim_{k\to\infty}x(k+1)-x(0) \qquad ④$$

$$右辺＝\lim_{z\to 1}((z-1)X(z)-z\,x(0))＝\lim_{z\to 1}(z-1)X(z)-x(0) \qquad ⑤$$

④，⑤より，　$\lim_{k\to\infty}x(k+1)=x(\infty)=\lim_{z\to 1}(z-1)X(z)$

（注意）　$X(z)=\dfrac{A}{(z-1)B}$　となっているかもしれないので，$\lim_{z\to 1}(z-1)X(z)$ で z に 1 を代入して 0 になるとしてはいけない。例題 1-4 の⑥，⑦式で定理を確認せよ。

1.4　かんたんなディジタル制御系

RL 回路の電流をコンピュータで制御する**ディジタル制御系**(digital control system)を考えることにより，**ディジタル制御**(digital control)とはどんなものか体験していただこう。

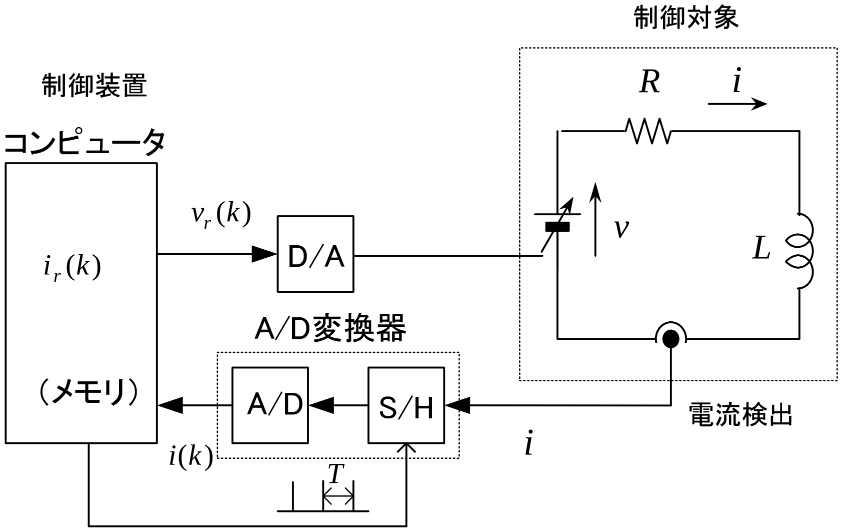


図 1-3　コンピュータ制御系(ディジタル制御系)の構成

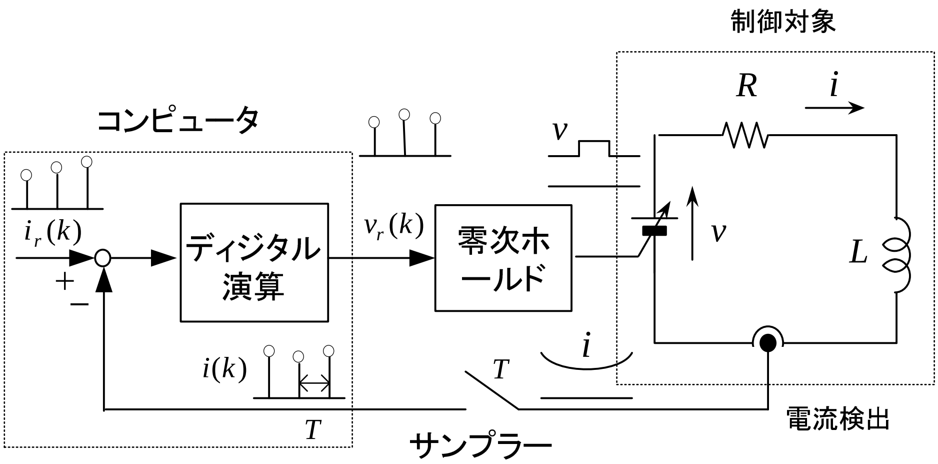


図 1-4 コンピュータ制御系(ディジタル制御系)のモデル

RL 回路の電流をホール素子で検出し，それを**A/D変換器**(A/D converter)によりコンピュータへ取り込む。取り込んだ値はコンピュータのメモリへ保存される。A/D 変換器の中には**サンプルホールド回路**(S/H)(sample hold circuit)があり，ある瞬間の値が保持され（ちょうど写真のように）それがディジタル量に変換される。この動作は**サンプリング周期** T ごとに行われ，**サンプラー**(sampler)として表わしている。なお，ディジタル量の精度は A/D 変換器のビット数で決まる。例えば 8 ビットデータに変換するなら四捨五入の様にして 256 通りの

数値のいずれかに割り当てられる。これを**量子化**(quantization)という。本テキストでは，十分なビット数があり量子化は考えず，時間的な動作のみを検討する。コンピュータでは，検出した$i(k)$とその**指令値**(reference value)$i_r(k)$をもとに制御演算を行い，その結果求まる入力指令$v_r(k)$をD/A変換器へ送る。D/A変換器へは，サンプリング周期Tごとにしか値は送られてこないが，制御対象へは連続的に電圧を加える必要があり，次の指令値が来るまでTの間その値を保持する。これを**零次ホールド**(zero-order holding)という（図1-5参照）。指令値に従って制御対象の入力$v(t)$が作られる。D/A変換器やvの電源部分については，実際にはいろいろの構成法があり，パワートランジスタを用いた電力変換器も必要であるが，最も単純な動作を仮定し，図1-4，図1-5のように考えよう(詳細は7章で述べる)。

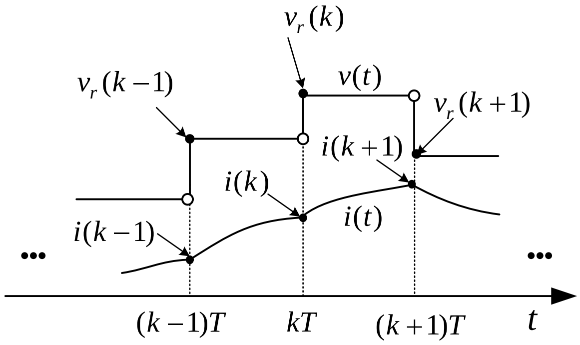


図1-5 サンプリングと零次ホールドの動作

図1-4のディジタル制御系のブロック線図を求め，安定解析を行う。

まず，図1-4の制御対象で，$v_r(k)$と$i(k+1)$の関係を求めてみよう。期間$kT \le t < (k+1)T$で成立する**微分方程式**は，この期間$v(t) = v_r(k)$であるから

$$v_r(k) = R\,i + L\frac{di}{dt} \tag{1-20}$$

である。この解は，この期間では$v_r(k)$が一定であるから直流回路と同じように考えて，

$$i(t) = \frac{v_r(k)}{R} + Ae^{-R\,t/L} \qquad ただし， \qquad kT \le t < (k+1)T \tag{1-21}$$

となる。(1-21)に，$t = kT$を代入し，$i(kT) \equiv i(k)$としてAを求めると，

$$A = \{i(k) - \frac{v_r(k)}{R}\}e^{R\,k\,T/L} \tag{1-22}$$

コイルがあるので電流は連続だから，(1-21)で$t = (k+1)T$とおき，$i((k+1)T) \equiv i(k+1)$と定義すると，次式の**差分方程式**が得られる（すぐ導けるのでヤレ）。

$$i(k+1) = e^{-RT/L}\,i(k) + \frac{1}{R}(1 - e^{-RT/L})\,v_r(k) \tag{1-23}$$

kは任意であるから，(1-23)は$k = 0, 1, 2, \cdots$と考えてよい。よって (1-23)で表される数列をz変換すると((1-14)参照)，次式が得られる。

$$z\,I(z) - z\,i(0) = e^{-RT/L}\,I(z) + \frac{1}{R}(1 - e^{-RT/L})V_r(z) \tag{1-24}$$

ここで，$I(z) = Z\{i(k)\}$，$V_r(z) = Z\{v_r(k)\}$

連続系の場合と同様，初期値 $i(0) = 0$ として，次式の伝達関数が得られる。

$$G_p(z) = \frac{I(z)}{V_r(z)} = \frac{1}{R}\frac{1 - e^{-RT/L}}{z - e^{-RT/L}} \tag{1-25}$$

z 変換を用いたディジタル制御の伝達関数は**パルス伝達関数**(pulse transfer function)と呼ばれる。なお，この伝達関数には，零次ホールドと制御対象それにサンプラーを含めた，$v_r(k)$ と $i(k)$ に関するものであることに注意してほしい。

次に，マイコンによる制御演算を考えよう。ここでは，最もかんたんな**P（比例）制御**(proportional control)が行われているとしよう。すなわち，

$$v_r(k) = K(i_r(k) - i(k)) \qquad (k = 0,1,2,\cdots) \tag{1-26}$$

とする。なお，$i(k)$ は k 時点での検出値で，$v_r(k)$ は k 時点で D/A 変換器に送られるから，(1-26)ではコンピュータの演算時間を無視している。これを z 変換すると，次式となる。

$$V_r(z) = K(I_r(z) - I(z)) \tag{1-27}$$

(1-25)，(1-27)から，図 1-6 のブロック線図が得られる。ブロック線図の定義や演算はラプラス変換の場合と同じである。

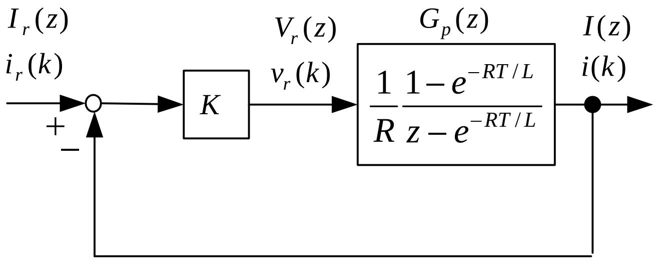


図 1-6　z 変換を用いた図 1-4 のブロック線図

図より，**閉ループパルス伝達関数**(closed loop pulse transfer function)は次式で与えられる。

$$G(z) = \frac{I(z)}{I_r(z)} = \frac{KG_p}{1 + KG_p}$$

$$= \frac{(1 - e^{-RT/L})K}{zR - Re^{-RT/L} + (1 - e^{-RT/L})K} \tag{1-28}$$

連続系同様，**閉ループ伝達関数の分母を** 0 とおいた式は**特性方程式**(characteristic equation)と呼ばれる。その根 z の**絶対値がすべて** 1 **以下であれば系は安定**である。

$$z = e^{-RT/L} - \frac{K}{R}(1 - e^{-RT/L}) \tag{1-29}$$

であり，安定条件は

$$|\,z\,| = \left| e^{-RT/L} - \frac{K}{R}(1 - e^{-RT/L}) \right| < 1$$

$$\therefore \quad -1 < e^{-RT/L} - \frac{K}{R}(1 - e^{-RT/L}) < 1$$

$e^{-RT/L} < 1$ に注意して，整理すると，安定となるゲイン K の範囲は，

$$-R < K < \frac{1 + e^{-RT/L}}{1 - e^{-RT/L}} R \tag{1-30}$$

サンプリング周期 T が非常に短いとき（これは連続系に近いと考えられよう），比例ゲイン K を大きく選んでも安定であることが判る。

なお，z 変換しないで，差分方程式のままでも安定判別は可能である。(1-26)を(1-23)に代入し， $v_r(k)$ を消去すると次式を得る

$$i(k+1) = [e^{-RT/L} - \frac{K}{R}(1 - e^{-RT/L})]i(k) + \frac{K}{R}(1 - e^{-RT/L})i_r(k) \tag{1-31}$$

この式を一般化したものは行列を使って下記の様に書ける（6 章で詳しく述べる）。

$\boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{r}(k)$　　　$\boldsymbol{r}(k)$ ：指令値

安定条件は(1-4)について考えた場合と同じで，(1-30)に一致する。なお，指令値 $i_r(k)$ は $i(k)$ に無関係であれば安定性に影響せず，一定である必要はない。

）微分方程式の公式（

微分方程式(differential equation)

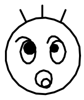
$$a\frac{dx}{dt} + bx = c$$

ただし，a, b, c は**一定の定数**（$c = 0$ でもよい）

解は，　$x = \dfrac{c}{b} + Ae^{-\frac{b}{a}t}$　　A は定数

電気回路は
文献(18)参照

(1-20)から(1-21)を導くとき利用する。



覚え方 1)微分の項を 0 とおいて， $x=\dfrac{c}{b}$ を得る。これが，第 1 項目

2) x に関係ない項 $c=0$ とし， $\dfrac{d}{dt}=p$ とおく。 $a\,p+b=0$ 　より， $p=-\dfrac{b}{a}$

これが，第 2 項目の t の係数。

（$a\,p+b=0$ 　を特性方程式という。）

1.5　ディジタル制御系の解析法まとめ

ディジタル制御系の解析では大きく 2 つの方法があることが判った。これを図 1-7 に示す。実際のシステムは一番上の段の様に，差分方程式で表されるディジタル制御器Ⓐ，入力を階段状に変化させる零次ホールドⒷ及び微分方程式で表される制御対象Ⓒからなる。これらから直接安定解析はできないので，ⒷとⒸからサンプリング時点に着目して差分方程式Ⓓを導く。ⒶとⒹよりシステム全体の差分方程式を求めると安定解析が可能となる（(1-31)に相当）。これが第 1 の方法である。次に，ⒶとⒹをともに z 変換して，システム全体のブロック線図を作り特性方程式を求めて安定判別するのが第 2 の方法である (図 1-6 のブロック線図)。第 2 の方法にはパルス伝達関数より周波数特性がすぐに得られる利点がある（第 5 章で述べる)。また，第 2 の方法では，①のルートで求める場合と，零次ホールドⒷと制御対象Ⓒのラプラス変換をまとめて z 変換することで求めることもできる（ルート②)。ルート②のことは第 2 章で詳しく述べる。

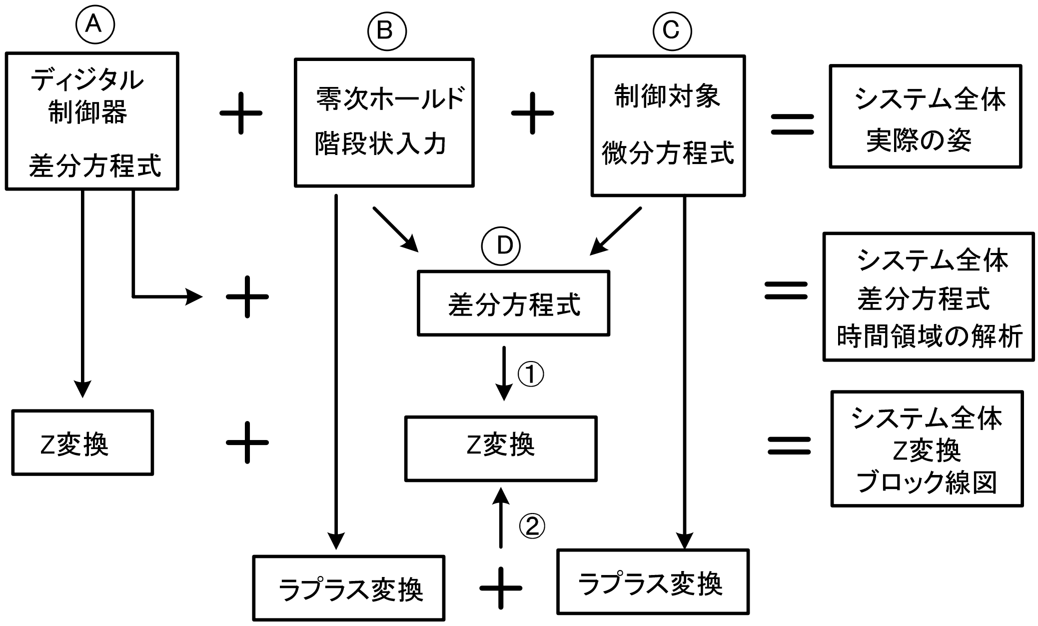
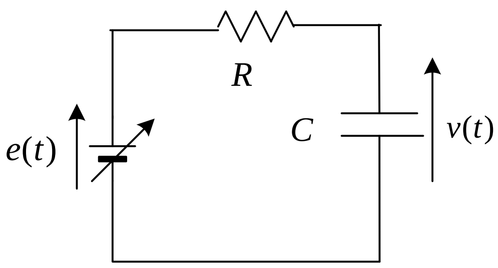


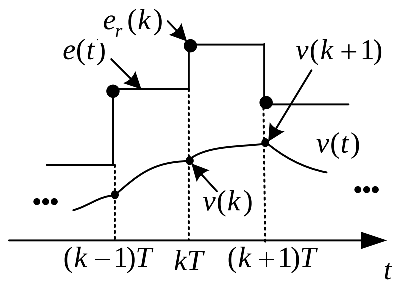
図 1-7 ディジタル制御系の解析法まとめ

［問題 1-3］ 図の回路（制御対象）で，サンプリング周期をT とし，零次ホールドにより

$$e(t) = e_r(k) \qquad kT \le t < (k+1)T$$

と階段状に制御する。電圧指令$e_r(k)$ に対するコンデンサ電圧 $v(k) \equiv v(kT)$ のパルス伝達関数$V(z)/E_r(z)$ を求めよ。

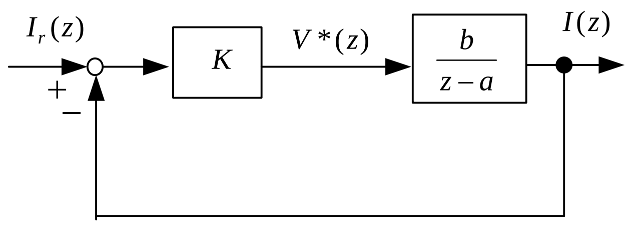




（答） $e_r(k) = RC\dfrac{dv}{dt} + v \qquad kT \le t < (k+1)T$

$$\frac{V(z)}{E_r(z)} = \frac{1-e^{-T/(RC)}}{z-e^{-T/(RC)}}$$

［問題 1-4］ 図の制御系で，安定となるゲインK の条件を求めよ。また，安定の場合に電流の指令値が$\{i_r(k)\} = \{1,1,1,\cdots\}$ のときの応答$i(k)$ および**定常偏差**(steady state error)を求めよ。ただし， a,b は正の定数とする。



（答） 安定条件 $|a-Kb|<1 \quad \therefore\ \dfrac{a-1}{b} < K < \dfrac{a+1}{b}$,

$$I(z) = \frac{Kb}{z-a+Kb}\,\frac{z}{z-1} \qquad \frac{I(z)}{z} = \frac{Kb}{z-a+Kb}\,\frac{1}{z-1} = \frac{Kb}{1-a+Kb}\left(\frac{1}{z-1} - \frac{1}{z-a+Kb}\right)$$

$$i(k) = \frac{Kb}{1-a+Kb}\left(1-(a-Kb)^k\right)$$

定常偏差 $i_r(\infty) - i(\infty) = \dfrac{1-a}{1-a+bK}$

（注）定常偏差だけなら最終値の定理 $i(\infty) = \lim_{z\to1}(z-1)I(z)$ が便利

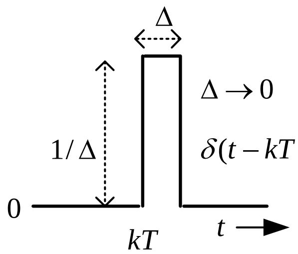
第２章　ブロック線図

2.1　インパルス列を用いた z 変換

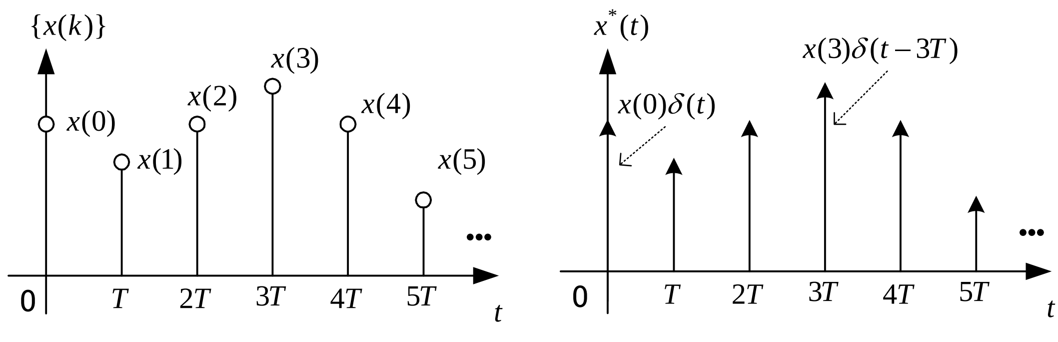
前節では，数列を用いて z 変換を定義したが，**インパルス列**(string of impulses)を用いても z 変換を求めることができる。数列で考えるより複雑そうに見えるが，これにより，z 変換したシステム全体のブロック線図が容易に求められるなどの利点がある。

数列$\{x(k)\}$に対応して，インパルス列$x^*(t)$を

$$
\begin{aligned}
x^*(t) &\equiv x(0)\delta(t) + x(1)\delta(t-T) + x(2)\delta(t-2T) + x(3)\delta(t-3T) + \cdots \\
&= \sum_{k=0}^{\infty} x(k)\delta(t-kT) \qquad (2\text{-}1)
\end{aligned}
$$



と定義する。**デルタ関数**(delta function)$\delta(t-kT)$は，$t = kT$のときのみ∞の値を持つが，幅は無限小で，その積分は 1 である(面積 1 の長方形の高さを無限に伸ばした形と考えると良い)。数列$\{x(k)\}$の例としては，図 2-1(a)に示す様に連続時間信号$x(t)$を周期Tごとに**サンプリング(標本化)**(sampling)した**サンプル値信号**(sampled-data signal)がある。



(a) サンプル値信号　　(b) インパルス列

図 2-1 サンプル値信号とインパルス列

インパルス列の各値は∞であり表示しにくいので，矢印をつけて表し，高さを数列の値に合わせて書く。両者の違いは，数列を積分しても値はないが，インパルス列は積分できる点にある。すなわちインパルス列は一般の関数と同じように**ラプラス変換**(Laplace transform)

ができる。そこで，$\delta(t-kT)$ のラプラス変換 $L(\ )$を求めてみよう。$\delta(t-kT)$ は $t=kT$ 付近以外は 0 なので，e^{-st} は一定値として積分の外へ出せる。

$$L(\delta(t-kT))=\int_0^\infty \delta(t-kT)e^{-st}dt$$

$$=e^{-skT}\int_0^\infty \delta(t-kT)\,dt$$

$$=e^{-skT} \tag{2-2}$$

従って，(2-1)をラプラス変換すると次式が得られる。

$$L(x^*(t))\equiv X^*(s)=x(0)+x(1)e^{-sT}+x(2)e^{-2sT}+x(3)e^{-3sT}+\cdots$$

$$=\sum_{k=0}^{\infty}x(k)e^{-skT} \tag{2-3}$$

いま，

$$e^{sT}\equiv z \tag{2-4}$$

とおくと，

$$X^*(s)=\sum_{k=0}^{\infty}x(k)\,z^{-k}$$

$$=X(z) \tag{2-5}$$

が成立する。このようにインパルス列をラプラス変換して z 変換を求めることができる。

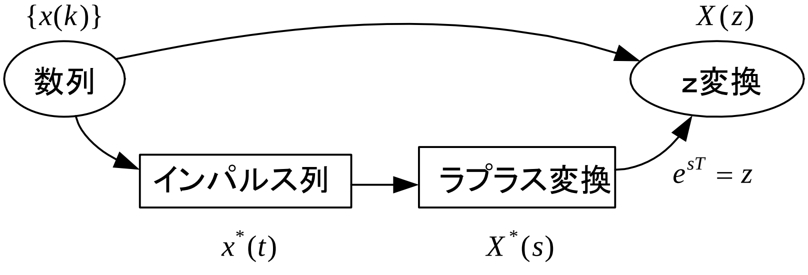


図 2-2　z 変換に至る 2 つの方法

2.2　z 変換によるディジタル制御系のブロック線図

(1)零次ホールドの伝達関数

　まず，重要な零次ホールドの伝達関数を求めよう。ここで考える零次ホールドの働きを図 2-3 に示す。 数列のラプラス変換はできないから，零次ホールドの入力としてインパルス列 $x^*(t)$ を考える。

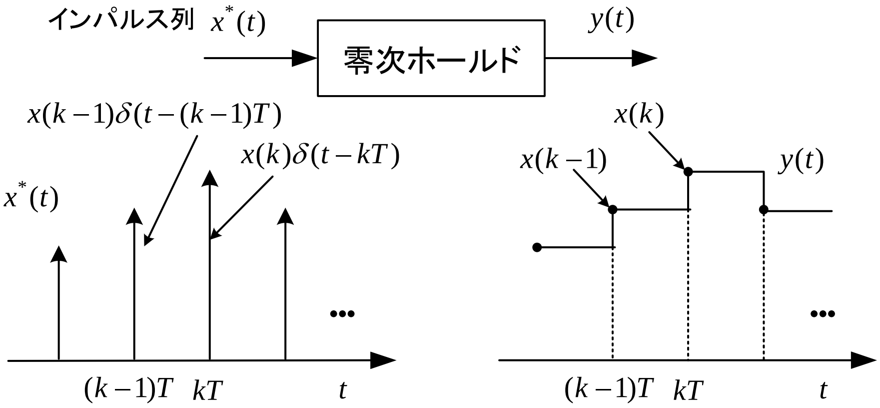


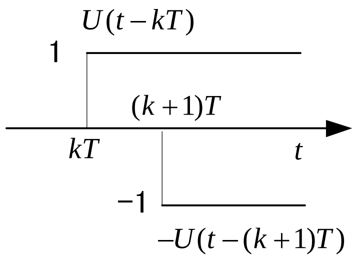
図 2-3 零次ホールド

単位ステップ関数 (unit step function)

$$U(t-kT)=\begin{cases}1 \ : \ t \geq kT \\ 0 \ : \ t < kT\end{cases} \tag{2-6}$$

を用いると，零次ホールドの出力は次式で表現できる。

$$y(t)=\sum_{k=0}^{\infty} x(k)[U(t-kT)-U(t-(k+1)T)] \tag{2-7}$$



(2-7)をラプラス変換すると以下のようになる。

$$\begin{aligned}Y(s)&=\sum_{k=0}^{\infty} x(k)\ \left(\frac{1}{s}e^{-skT}-\frac{1}{s}e^{-s(k+1)T}\right)\\ &=\frac{1}{s}\ (1-e^{-sT})\sum_{k=0}^{\infty} x(k)\,e^{-skT}\\ &=\frac{1}{s}\ (1-e^{-sT})\ X^*(s)\end{aligned} \tag{2-8}$$

従って，**零次ホールドの伝達関数** $G_{oh}(s)$ は,

$$G_{oh}(s)=\frac{1}{s}\ (1-e^{-sT}) \tag{2-9}$$

で与えられる。このゲインと位相特性は付録にある。

零次ホールドを伝達関数で表すためにインパルス列という考え方が役立った。そしてこの伝達関数は零次ホールド＋制御対象のパルス伝達関数（z 変換）を求めるのに役立つ。

（2）連続時間信号及びラプラス変換した値のz変換

零次ホールドの伝達関数を利用することを考える。連続時間信号 $x(t)$ をサンプリングして得られる数列$\{x(k)\}$（図 2-1(a)を見よ）を用いて z 変換が求まるが，$x(t)$ にはラプラス変換 $X(s)$ も存在する。そこで $X(s)$ **のz変換が** $X(s)$ **を逆ラプラス変換した** $x(t)$ **から得られる数列のz変換と約束して**以下のように書く。

$$Z\{x(k)\} \equiv Z\{x(kT)\} \equiv Z(x(t)) \equiv Z(X(s)) \tag{2-10}$$

これから，以下の公式を得る。

公式 1　$Z(a\,f(t)+b\,g(t)) = aZ(f(t))+bZ(g(t))$　(2-11)

$Z(a\,F(s)+b\,G(s)) = a\,Z(F(s))+b\,Z(G(s))$　(2-12)

公式 2　$Z(U(t)) = Z(\frac{1}{s}) = \frac{z}{z-1}$　(2-13)

公式 3　$Z(t) = Z(\frac{1}{s^2}) = \frac{T\,z}{(z-1)^2}$　(2-14)

公式 4　$Z(e^{-a\,t}) = Z(\frac{1}{s+a}) = \frac{z}{z-e^{-aT}}$　(2-15)

公式 5　$Z(f(t-nT)) = Z(e^{-s\,n\,T}F(s)) = z^{-n}Z(F(s))$　(2-16)

公式 6　$Z(\frac{1-e^{-sT}}{s}\frac{1}{s+a}) = (1-z^{-1})Z(\frac{1}{s(s+a)}) = \frac{1}{a}\frac{1-e^{-a\,T}}{z-e^{-a\,T}}$　(2-17)

ただし，n は整数，a，b は定数。$F(s)=L(f(t)), G(s)=L(g(t))$ ：ラプラス変換

公式 1 の証明：(1-17)に(2-10)の定義を用いると得られる。

公式 2 の証明：単位ステップ関数$U(t)$ をサンプリングした数列は，$\{1,1,1,\cdots\}$ であるから，(2-10)の定義を用いると得られる。

公式 3 の証明：数列$\{0,T,2T,3T,\cdots\}$ を z 変換すればよく，例題 1-2（注）で $p=1$ とおき，全体を T 倍すればよい。

公式 4 の証明：

$$Z\{1,e^{-aT},e^{-2aT},e^{-3aT},\cdots\} = 1+e^{-aT}z^{-1}+e^{-2aT}z^{-2}+e^{-3aT}z^{-3}+\cdots$$

$$= \frac{1}{1-e^{-aT}z^{-1}} = \frac{z}{z-e^{-aT}}$$

公式 5 の証明：

$$Z(f(t)) \equiv Z(F(s)) = Z\{f(0), f(T), f(2T), f(3T), \cdots\}$$

$$= f(0) + f(T)z^{-1} + f(2T)z^{-2} + f(3T)z^{-3} + \cdots$$

$$Z(f(t-T)) = Z\{0, f(0), f(T), \cdots\}$$

$$= f(0)z^{-1} + f(T)z^{-2} + f(2T)z^{-3} + \cdots$$

$$= z^{-1} Z(F(s))$$

$$Z(f(t-nT)) = Z\{f(-nT), f(-(n-1)T), \cdots, f(0), f(T), \cdots\}$$

$$= Z\{0, 0, \cdots, 0, f(0), f(T), \cdots\}$$

$$= f(0)z^{-n} + f(T)z^{-(n+1)} + f(2T)z^{-(n+2)} + \cdots$$

$$= z^{-n} Z(F(s))$$

公式 6 の証明：

$$Z(\frac{1-e^{-sT}}{s}\frac{1}{s+a}) = (1-z^{-1})Z(\frac{1}{s(s+a)}) \qquad ((2\text{-}11),(2\text{-}16)\text{より})$$

$$= (1-z^{-1})Z(\frac{1}{a}(\frac{1}{s}-\frac{1}{s+a})) = \frac{1}{a}(1-z^{-1})(\frac{z}{z-1}-\frac{z}{z-e^{-aT}}) = \frac{1}{a}\frac{1-e^{-aT}}{z-e^{-aT}}$$

(3) z変換によるシステムのブロック線図

零次ホールドの伝達関数を用いたディジタル制御系のチャンポンブロック線図（この言葉は著者の造語で一般にはない(champon; noodles with seafood, vegetables, etc.)）を図 2-4 に示す。$u^*(t)$ は入力のインパルス列である。すなわち,

$$u^*(t) = \sum_{k=0}^{\infty} u(k)\delta(t-kT)$$

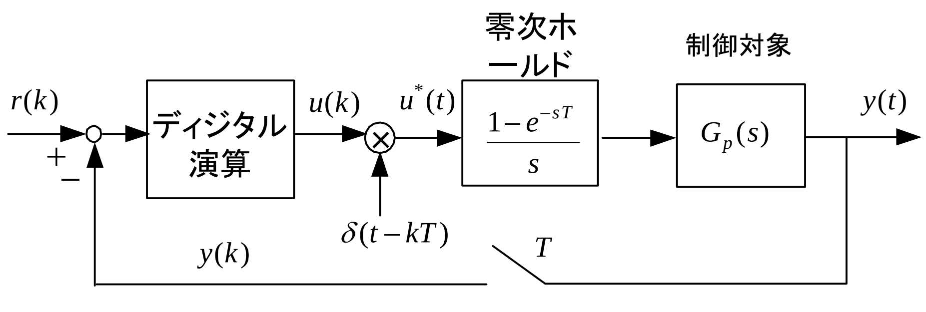


図 2-4　ディジタル制御系のチャンポンブロック線図

入力 $u(k)$ の z 変換とそのインパルス列 $u^*(t)$ から求まる z 変換は同じであるから（図 2-2），簡単のため図 2-4 を図 2-5 のように書くこともある。図 2-5 で $y(k), r(k), u(k)$ をその z 変換 $Y(z), R(z), U(z)$ で置き換えたものは厳密に正しい。ブロック線図は演算の仕方さえ誤らなければ，そこから得られるイメージがピンとくるものが望ましく，ディジタル制御器は差分方程式（または z 変換したもの），制御対象は連続系の伝達関数で表現されているのが物理的に判りやすいと思われる。なぜなら，差分方程式の方が制御器のソフトウェアとの結びつきが良く，制御対象はラプラス変換されたものが判りやすいからである。

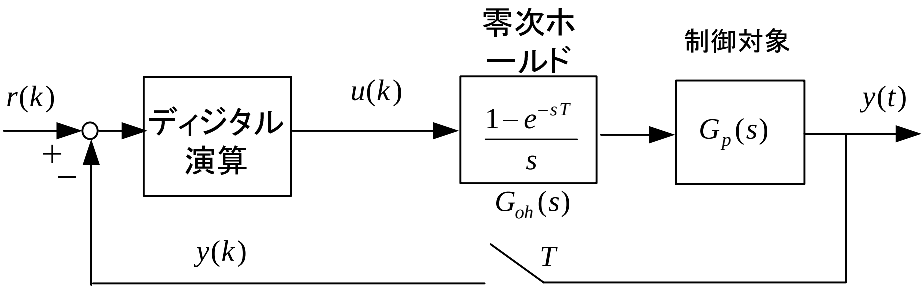
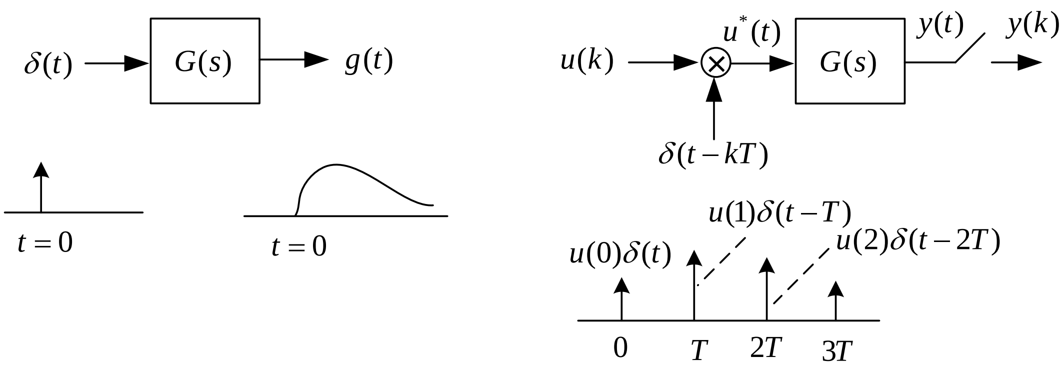


図 2-5 ディジタル制御系のチャンポンブロック線図（図 2-4 の簡易表示）

図 2-4 や図 2-5 では差分方程式のブロックとラプラス変換した連続系のブロックがチャンポンになっており，解析を行うためにはシステム全体を z 変換のブロック線図で表示することが望まれる。

そこで， $u(k)$ と $y(k)$ の間の伝達関数を求めよう。零次ホールド＋制御対象の伝達関数を $G(s) = G_{0h}(s)G_p(s)$ として，図 2-6 に示すブロック線図を考える。



(a)　インパルス応答 $g(t)$　　　(b)　インパルス列に対する応答 $y(t)$

図 2-6　零次ホールド＋制御対象の応答

(a)図に示すように， $G(s)$ のインパルス応答を $g(t)$ とすると（L^{-1}：逆ラプラス変換），

$$g(t)=L^{-1}\left(G(s)\right) \tag{2-18}$$

である（何故なら$\delta(t)$のラプラス変換は1だから）。(b)図で$k=0$として，入力が$u(0)\delta(t)$のときの出力が(a)図より$u(0)g(t)$だから，それをサンプリングした信号は

$$\left\{y_0(k)\right\}=\left\{u(0)g(0),u(0)g(T),u(0)g(2T),u(0)g(3T),\cdots\right\} \tag{2-19}$$

と表せる。$k=1$として，入力$u(1)\delta(t-T)$に対する出力は$u(1)g(t-T)$だから，それをサンプリングした信号は時間T 以降に生じ

$$\left\{y_1(k)\right\}=\left\{0,u(1)g(0),u(1)g(T),u(1)g(2T),\cdots\right\} \tag{2-20}$$

である。以下同様に求めることができ，それらの和として

$$\begin{aligned}\left\{y(k)\right\}=\{&u(0)g(0)\\ &,u(0)g(T)+u(1)g(0)\\ &,u(0)g(2T)+u(1)g(T)+u(2)g(0)\\ &,u(0)g(3T)+u(1)g(2T)+u(2)g(T)+u(3)g(0)\\ &,\cdots\ \}\end{aligned} \tag{2-21}$$

Σを用いて成分を書くと， $y(k)=\sum_{i=0}^{k}u(i)g((k-i)T) \qquad k=0,1,2,3,\cdots$

である。$y(k)$の右辺は数列uとgの**畳み込み和**(convolution sum)と呼ばれている。 (2-21)式をz変換すると定義より

$$\begin{aligned}Y(z)&=Z\{y(k)\}=u(0)G(z)+u(1)G(z)z^{-1}+u(2)G(z)z^{-2}+u(3)G(z)z^{-3}+\cdots\\ &=(u(0)+u(1)z^{-1}+u(2)z^{-2}+u(3)z^{-3}+\cdots)G(z)\\ &=G(z)U(z)\end{aligned} \tag{2-22}$$

ここで，$G(z)$は$g(t)$をサンプリングした信号のz変換で次式を用いた。

$$G(z)=g(0)+g(T)z^{-1}+g(2T)z^{-2}+g(3T)z^{-3}+\cdots \tag{2-23}$$

(2-22)は任意の$G(s)$について成り立つ。つまり，$G(s)$の中に零次ホールドが含まれなくてよい。(2-22)より，図2-4について，数列$\{u(k)\}$と数列$\{y(k)\}$に関する伝達関数は

$$G(z)\equiv Z(g(t))\equiv Z(G(s))=\frac{Y(z)}{U(z)}=Z(\frac{1-e^{-sT}}{s}G_p(s))=(1-z^{-1})Z(\frac{G_p(s)}{s}) \tag{2-24}$$

により計算できることが判った。すなわち<u>零次ホールド＋制御対象の部分をまとめてz変換するだけで良い</u>。$G(z)$を用いると，図2-4は図2-7で表わせる。なお，ディジタル制御

器の部分は差分方程式で表されるので，z 変換は容易で$C(z)$はすぐに得られる。(2)節の公式 1～6 は，制御対象の伝達関数$G_p(s)$が得られている場合，(2-24)より零次ホールド＋制御対象の部分の伝達関数$G(z)$を求めるのに便利である。

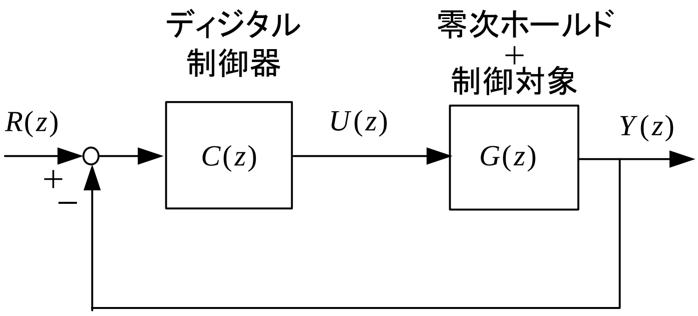
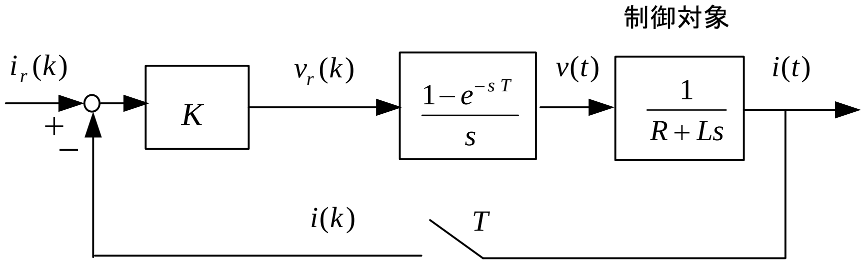


図 2-7　z 変換によるシステムのブロック線図

[例題 2-1]　RL回路のディジタル制御系で，電流の比例制御を考えると，以下のブロック線図が得られる。これから z 変換したブロック線図を導け。



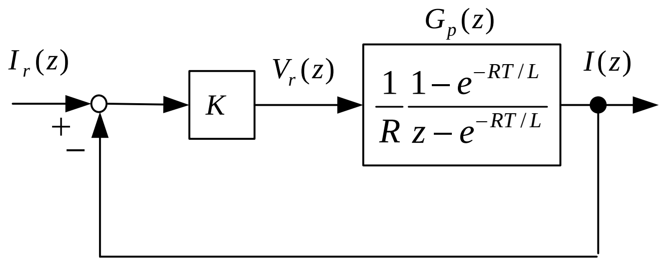
（解）図で，零次ホールドと制御対象をまとめて z 変換したパルス伝達関数$G(z)$は，公式 6 より，以下のように求まる。（図の$v_r(k)$と$i(k)$の間）

$$\frac{I(z)}{V_r(z)} = G(z) = Z\left(\frac{1-e^{-sT}}{s}\frac{1}{R+Ls}\right) = \frac{1}{L}Z\left(\frac{1-e^{-Ts}}{s}\frac{1}{s+(R/L)}\right) = \frac{1}{R}\frac{1-e^{-RT/L}}{z-e^{-RT/L}}$$

比例制御の部分については，次式のように z 変換できる。

$$V_r(z) = K(I_r(z) - I(z))$$

以上により， z 変換した量についてのブロック線図が以下の様に得られる。



なお，このブロック線図は，既に図 1-6 に示していたものと同じである。しかし，制御対象が比較的簡単な場合には，例題 2-1 の方法が容易にブロック線図を求めることができ，便利である。

[例題 2-2]　制御対象が次式で与えられるとき，零次ホールドと制御対象をまとめたパルス伝達関数を求めよ。

$$G_p(s)=\frac{1}{s(1+sT_p)}$$

(解)　$G(z)=(1-z^{-1})Z(\frac{G_p(s)}{s})=(1-z^{-1})\ Z(\frac{1}{s^2(1+sT_p)})$

$$=(1-z^{-1})\,Z(\frac{a}{s}+\frac{b}{s^2}+\frac{c}{1+sT_p})$$

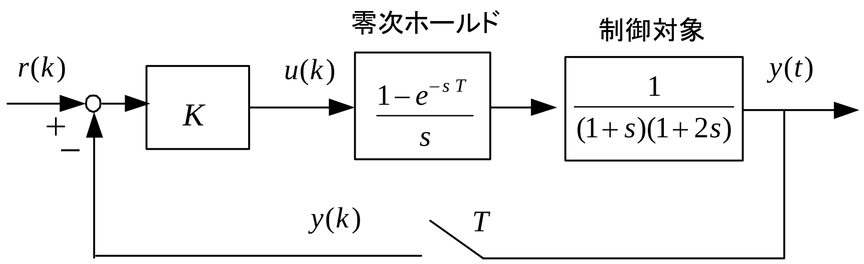
恒等式を解いて，$a=-T_p$，$b=1$，$c=T_p^2$

$$G(z)=(1-z^{-1})Z(-\frac{T_p}{s}+\frac{1}{s^2}+\frac{T_p^2}{1+sT_p})$$

$$=\frac{z-1}{z}(-\frac{T_p\,z}{z-1}+\frac{T\,z}{(z-1)^2}+\frac{T_p\,z}{z-e^{-T/T_p}})$$

$$=\frac{T}{z-1}-\frac{T_p(1-e^{-T/T_p})}{z-e^{-T/T_p}}$$

[例題 2-3]　零次ホールドと制御対象をまとめたパルス伝達関数$Y(z)/U(z)$を求めよ。



(解) $$\frac{Y(z)}{U(z)}=(1-z^{-1})Z(\frac{1}{s(1+s)(1+2s)})=(1-z^{-1})Z(\frac{1}{s}+\frac{1}{1+s}-\frac{4}{1+2s})$$

$$=1+\frac{z-1}{z-a}-\frac{2(z-1)}{z-b}$$

$$=\frac{(a+1-2b)z+ab+b-2a}{(z-a)(z-b)}$$

ただし，$a=e^{-T},b=e^{-T/2}$

〔例題 2-4〕 制御対象が次式で与えられるとき，零次ホールドと制御対象をまとめたパルス伝達関数を求めよ。

$$G_p(s)=\frac{1}{s^2+2\xi s+1} \quad , \quad 0<\xi<1$$

（解） $$G(z)=(1-z^{-1})Z\left(\frac{G_p(s)}{s}\right)=(1-z^{-1})Z\left(\frac{a}{s}+\frac{bs+c}{s^2+2\xi s+1}\right)$$

係数を比べて，$(a+b)s^2+(2\xi a+c)s+a=1 \quad \therefore a=1 \quad , \quad b=-1 \quad , \quad c=-2\xi$

$$G(z)=\frac{z-1}{z}Z\left(\frac{1}{s}-\frac{s+2\xi}{s^2+2\xi s+1}\right)$$

付録の公式を利用する。

$$Z\left(\frac{1}{s}\right)=\frac{z}{z-1}$$

$$Z\left(\frac{s+2\xi}{s^2+2\xi s+1}\right)=Z\left(\frac{s+\xi}{(s+\xi)^2+1-\xi^2}+\frac{\xi}{(s+\xi)^2+1-\xi^2}\right)$$

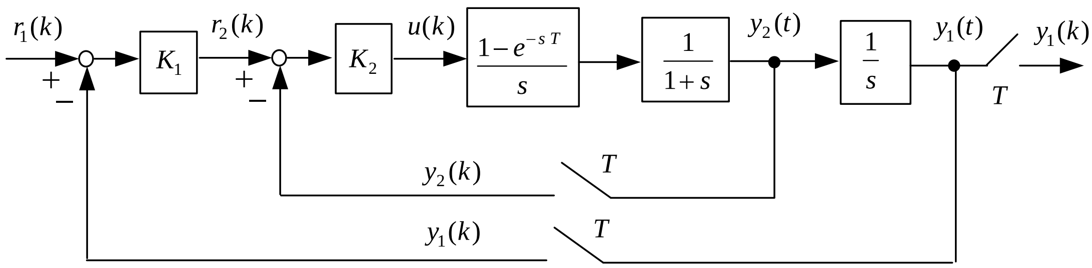
$$=\frac{z\left(z-e^{-\xi T}\cos\beta T\right)}{z^2-2e^{-\xi T}\left(\cos\beta T\right)z+e^{-2\xi T}}+\frac{\xi}{\beta}\cdot\frac{ze^{-\xi T}\sin\beta T}{z^2-2e^{-\xi T}\left(\cos\beta T\right)z+e^{-2\xi T}}$$

ただし，$\beta=\sqrt{1-\xi^2}$

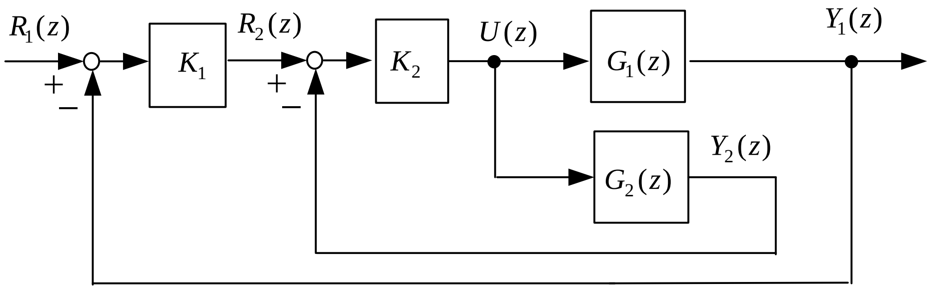
整理して，

$$G(z)=1-\frac{(z-1)\left\{z+(e^{-\xi T}/\beta)(\xi\sin\beta T-\beta\cos\beta T)\right\}}{z^2-2e^{-\xi T}(\cos\beta T)z+e^{-2\xi T}}$$

〔例題 2-5〕　マイナーループ（$y_2(k)$ のフィードバック）を有する図のディジタル制御システムを z 変換による伝達関数だけを用いたブロック線図に直し，$Y_1(z)/R_1(z)$ を求めよ。



（解）求めるブロック線図は図のように表せる。



ここで，

$$G_1(z)=(1-z^{-1})Z(\frac{1}{s^2(1+s)})=\frac{T}{z-1}-\frac{1-e^{-T}}{z-e^{-T}}$$ （例題 2-2 より）

$$G_2(z)=(1-z^{-1})Z(\frac{1}{s(1+s)})=\frac{1-e^{-T}}{z-e^{-T}}$$ （例題 2-1 より）

一般のブロック線図と同様に計算して，

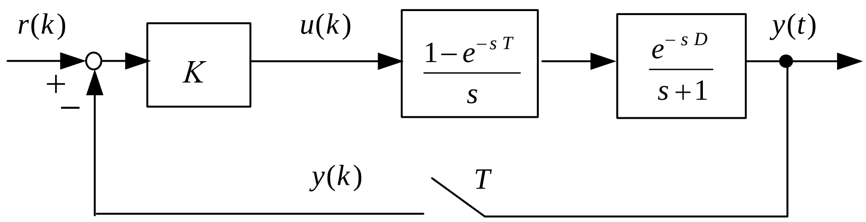
$$\frac{U(z)}{R_2(z)}=\frac{K_2}{1+K_2G_2}$$

だから，

$$\frac{Y_1(z)}{R_1(z)}=\frac{K_1G_1\frac{K_2}{1+K_2G_2}}{1+K_1G_1\frac{K_2}{1+K_2G_2}}=\frac{K_1K_2G_1}{1+K_2G_2+K_1K_2G_1}$$

K_1, K_2 は z の関数すなわち $K_1(z)$, $K_2(z)$ でも成立つ。

〔例題 2-6〕　図の**むだ時間要素**を含む制御対象のディジタル制御システムで，$Y(z)/U(z)$ を求めよ。ただし，$T=1\mathrm{s}$, $D=1.2\mathrm{s}$ とする。



（解）$\dfrac{Y(z)}{U(z)}=(1-z^{-1})Z(\dfrac{e^{-1.2s}}{s(1+s)})$　　①

ラプラス逆変換して

$$L^{-1}(\frac{1}{s(1+s)})=L^{-1}(\frac{1}{s}-\frac{1}{1+s})=1-e^{-t}$$

故に　$L^{-1}(\dfrac{e^{-1.2s}}{s(1+s)})=\begin{cases}0 & 0\le t\le 1.2\\ 1-e^{-(t-1.2)} & 1.2<t\end{cases}$

$$\begin{aligned}Z(\frac{e^{-1.2s}}{s(1+s)})&=Z\left\{\underset{t\Rightarrow 0}{0},\ \underset{T}{0},\ \underset{2T}{1-e^{-(2-1.2)}},\underset{3T}{1-e^{-(3-1.2)}},\underset{4T}{1-e^{-(4-1.2)}},\cdots\right\}\\&=z^{-2}+z^{-3}+z^{-4}+\cdots-e^{1.2}(e^{-2}z^{-2}+e^{-3}z^{-3}+e^{-4}z^{-4}+\cdots)\\&=\frac{z^{-2}}{1-z^{-1}}-e^{1.2}\frac{e^{-2}z^{-2}}{1-e^{-1}z^{-1}}\qquad ②\end{aligned}$$

①，②より

$$\begin{aligned}\frac{Y(z)}{U(z)}&=(1-z^{-1})(\frac{z^{-2}}{1-z^{-1}}-e^{1.2}\frac{e^{-2}z^{-2}}{1-e^{-1}z^{-1}})\\&=\frac{1}{z^2}\frac{(e-e^{0.2})z+e^{0.2}-1}{ez-1}\end{aligned}$$

第３章　ディジタル PI 制御と離散化

ディジタル制御器の設計を行う場合，まずよく知られた連続系で設計し，得られた制御則を差分方程式に直して（**離散化** discretization という），解析や実験を行ったうえで最終的な制御ゲインや補償器を決定する場合が多い。これを**ディジタル再設計**(digital redesign)という。離散化は，制御用のコンピュータに組み込むプログラムを作る場合になくてはならないものである。制御法は多く存在し，ディジタル制御特有のものもある。本章では，良く用いられている**PI制御（比例積分制御）**やフィルタなどを例にとり，離散化の方法を述べる。また，実際には入力値に上限や下限があるので**リミッタ**が設けられる。

3.1　ディジタルＰＩ制御

ＰＩ（比例＋積分）制御は簡単に実現でき，ステップ応答の定常偏差を 0 にすることから，ディジタル制御においても広く用いられている。実用上大変重要である。

まず，次式で表せる積分制御について考えよう。

$$u(t) = \int_0^t e(t)dt \qquad \text{(3-1)}$$

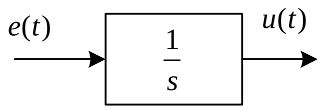


図 3-1　積分器

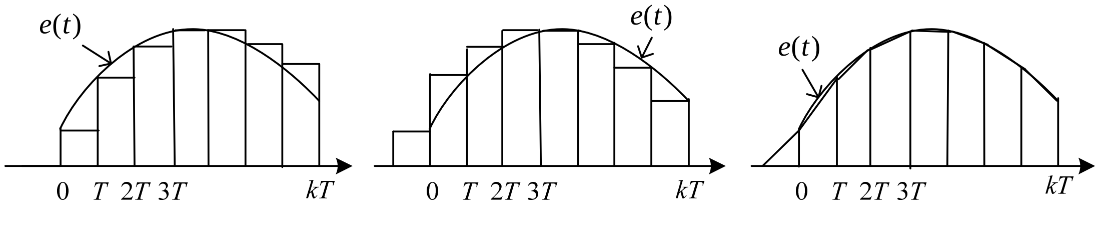
この積分を数値的に計算する場合，まず基本的な 3 つの方法を考える。

① $u(k) = \sum_{m=0}^{k} e(m-1)T$　　（**前進矩形近似**）　　(3-2)

② $u(k) = \sum_{m=0}^{k} e(m)T$　　（**後退矩形近似**）　　(3-3)

③ $u(k) = \frac{1}{2}\sum_{m=0}^{k} (e(m)+e(m-1))T$　　（**台形近似**）　　(3-4)

ただし， $u(k) \equiv u(kT), e(k) \equiv e(kT)$ と書いている。また， $e(-1)=0$ とする。これらを図形的に書くと図 3-2 となる。 $t = kT$ までの長方形または台形の和が $u(k)$ である。



①　前進矩形近似　　②後退矩形近似　　③台形近似

図 3-2　積分の近似法

(3-2)から(3-4)を，差分方程式の形に書き換えてみると，次式が得られる。$u(k-1)$ は，(3-2) の $u(k)$ で k を $k-1$ とおいた値である。

① $u(k)=u(k-1)+e(k-1)T$ (3-5)

② $u(k)=u(k-1)+e(k)T$ (3-6)

③ $u(k)=u(k-1)+\frac{1}{2}(e(k)+e(k-1))T$ (3-7)

これらは，それぞれ微分方程式を①前進オイラー法(単にオイラー法とも呼ばれる)，②後退オイラー法，③台形公式で差分近似したものに一致している。微分方程式の数値解析ではこれらの名称が使われる。いろいろの呼び方がある。

① **前進オイラー法（前進差分，前進矩形近似）**

(forward Euler’s method, forward difference, forward rectangular rule)

$$\frac{du(t)}{dt}=e(t)\rightarrow\frac{u(k)-u(k-1)}{T}=e(k-1)$$

② **後退オイラー法（後退差分，後退矩形近似）**

(backward Euler’s method, backward difference, backward rectangular rule)

$$\frac{du(t)}{dt}=e(t)\rightarrow\frac{u(k)-u(k-1)}{T}=e(k)$$

③ **台形公式（双一次変換，タスティン変換，台形近似）**

(trapezoidal rule, bilinear transform, Tustin’s rule)

$$\frac{du(t)}{dt}=e(t)\rightarrow\frac{u(k)-u(k-1)}{T}=\frac{1}{2}\left\{e(k-1)+e(k)\right\}$$

台形近似は少し複雑となるが，精度はこの中では最も良い。サンプリング周期 T を短く選ぶと精度は向上するが，短いほど良いというわけではない。**情報落ち**という問題がある。(3-5)～(3-7)で T が短かすぎると $e(k-1)T$ などの増分は小さな値となり，$u(k-1)$ と加えたときに，増分が全て切り捨てられてしまう場合が起こるのである。これは使用する制御用マイコンの有効桁数（固定小数点か浮動小数点か）とも関係する。積分器の近似を，上記の 3 つの方法で比較した結果，台形公式が位相角に関して最も正確であると述べられている（文献(3)）。前進矩形近似では積分値に最新のデータが入っていないのは望ましくない。

(3-5)から(3-7)の差分方程式を z 変換すると，それぞれ以下のようになる。

① $U(z)=z^{-1}U(z)+z^{-1}E(z)T$

② $U(z)=z^{-1}U(z)+E(z)T$

③ $U(z)=z^{-1}U(z)+\frac{T}{2}(1+z^{-1})E(z)$

従って，パルス伝達関数は以下のようになる（積分要素$1/s$に対応）。

① $$\frac{U(z)}{E(z)}=\frac{Tz^{-1}}{1-z^{-1}}=\frac{T}{z-1}$$ （前進矩形近似） (3-8)

② $$\frac{U(z)}{E(z)}=\frac{T}{1-z^{-1}}=\frac{Tz}{z-1}$$ （後退矩形近似） (3-9)

③ $$\frac{U(z)}{E(z)}=\frac{T}{2}\left(\frac{1+z^{-1}}{1-z^{-1}}\right)=\frac{T}{2}\left(\frac{z+1}{z-1}\right)$$ （台形近似） (3-10)

図 3-3 の伝達関数は(3-9)となり，これが(3-6)に対応している。図 3-3 を見ると前の値に増分を加えて積分値になる様子が良くわかる。

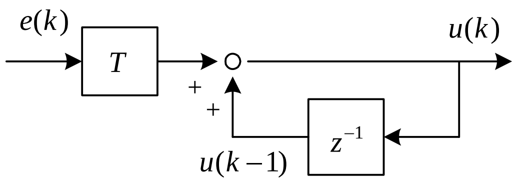


図 3-3 積分器のパルス伝達関数のブロック線図（後退矩形近似）

さて，いよいよＰＩ制御の話をしよう。アナログ制御系(連続系）の場合，ＰＩ制御は次式で表現される。

$$u(t)=K_P e(t)+K_I\int_0^t e(t)dt \tag{3-11}$$

ここで，偏差：$e(t)=r(t)-y(t)$

これを数値的に計算する場合，積分の近似の違いで基本的に３つの方法が考えられる。

① $$u(k)=K_P e(k)+K_I\sum_{m=0}^{k}e(m-1)T$$ （前進矩形近似） (3-12)

② $$u(k)=K_P e(k)+K_I\sum_{m=0}^{k}e(m)T$$ （後退矩形近似） (3-13)

③ $$u(k)=K_P e(k)+\frac{K_I}{2}\sum_{m=0}^{k}(e(m)+e(m-1))T$$ （台形近似） (3-14)

ここで，$e(k)=r(k)-y(k)$

これらは，**位置アルゴリズム**(position algorithm)と呼ばれている。

(3-12)から(3-14)を書き換えて，差分方程式の形に書き換えてみると，以下の式が得られる。$u(k-1)$は，$u(k)$でkを$k-1$とおいた値である。

① $u(k)=u(k-1)+K_P(e(k)-e(k-1))+K_I Te(k-1)$ (3-15)

② $u(k)=u(k-1)+K_P(e(k)-e(k-1))+K_I Te(k)$ (3-16)

③ $u(k)=u(k-1)+K_P(e(k)-e(k-1))+\frac{K_I T}{2}(e(k)+e(k-1))$ (3-17)

これらは，**速度アルゴリズム**(velocity algorithm)と呼ばれている。(3-15)から(3-17)を z 変換して，P I 制御器のパルス伝達関数は以下のようになる。

① $C(z)=\frac{U(z)}{E(z)}=K_P+\frac{K_I T}{z-1}$ (3-18)

② $C(z)=\frac{U(z)}{E(z)}=K_P+\frac{K_I T\,z}{z-1}$ (3-19)

③ $C(z)=\frac{U(z)}{E(z)}=K_P+\frac{K_I T}{2}(\frac{z+1}{z-1})$ (3-20)

実際の P I 制御としては，後退矩形近似か台形近似が用いられているようである。

次に，連続系の **P I D 制御**(proportional-integral-derivative control)は，次式で与えられる。

$$u(t)=K_P e(t)+K_I\int_0^t e(t)dt+K_D\frac{d\,e(t)}{d\,t} \quad (3\text{-}21)$$

これを離散化する場合，積分制御と微分制御に後退矩形近似を用いると次式の**位置アルゴリズム**が得られる（文献(6)）。

$$u(k)=K_P e(k)+K_I\sum_{m=0}^{k}e(m)T+\frac{K_D}{T}(e(k)-e(k-1)) \quad (3\text{-}22)$$

これより**速度アルゴリズム**は次式となる。

$$u(k)=u(k-1)+K_P(e(k)-e(k-1))+K_I Te(k)+\frac{K_D}{T}(e(k)-2e(k-1)+e(k-2)) \quad (3\text{-}23)$$

後述するが速度アルゴリズムにはワインドアップ防止の効果や目標値の大幅な変更に対して応答を良くする効果がある。なお $e(k)$ がノイズを含む場合には，誤差が大きくなって問題である。その場合には，微分要素として後述の**擬似微分**(pseudo differential)を用いる。

3.2　フィルタおよび補償要素

ディジタル制御器には，P I 制御の他に，一次遅れ要素や位相要素が用いられることがある。この場合の差分方程式を導いておく。連続系として表わした制御器の伝達関数を $C_c(s)=U(s)/E(s)$ とすると（$U(s),E(s)$ はそれぞれ入力，偏差とは限らない），

P I 制御　：$C_c(s)=K_P+\frac{K_I}{s}$, $U(s)=(K_P+\frac{K_I}{s})E(s)$ (3-24)

一次遅れ要素：　$C_c(s)=\frac{\omega_0 K}{\omega_0+s}$ ，$U(s)=\frac{\omega_0 K}{\omega_0+s}E(s)$ (3-25)

位相要素： $C_c(s) = K\dfrac{s+\omega_1}{s+\omega_2}$， $U(s) = K\dfrac{s+\omega_1}{s+\omega_2}E(s)$ (3-26)

と表わされる。伝達関数$C_c(s)$が与えられる場合，ディジタル補償要素$C(z)$の求め方として，次式が用いられることがある。

前進矩形近似： $$C(z) = C_c(s)\Big|_{s=\frac{z-1}{T}} \tag{3-27}$$

後退矩形近似 ： $$C(z) = C_c(s)\Big|_{s=\frac{z-1}{Tz}} \tag{3-28}$$

台形近似 ： $$C(z) = C_c(s)\Big|_{s=\frac{2(z-1)}{T(z+1)}} \tag{3-29}$$

(3-8)～(3-10)が積分要素$1/s$の z 変換であるから，これらの関係式が得られる。但し，純粋微分sには台形近似は適用せず，近似微分するかその部分だけに矩形近似を用いる。これはPID 制御の D 制御で現れる。PI 制御の$C(z)$は既に(3-18)～(3-20)で求め, 差分方程式も(3-15)～(3-17)にある。

（１） 一次遅れ要素

・ **前進矩形近似**

$$C(z) = \frac{\omega_0 K}{\omega_0 + \dfrac{z-1}{T}} = \frac{T\omega_0 K}{\omega_0 T + z - 1} = \frac{z^{-1}T\omega_0 K}{(\omega_0 T - 1)z^{-1} + 1}$$

$U(z) = C(z)E(z)$ より， $((\omega_0 T - 1)z^{-1} + 1)U(z) = z^{-1}T\omega_0 K\,E(z)$

$$\therefore\ U(z) = (1-\omega_0 T)z^{-1}U(z) + z^{-1}\omega_0 T\,KE(z)$$

差分方程式で表わすと，

$$u(k) = (1-\omega_0 T)u(k-1) + \omega_0 T\,Ke(k-1) \tag{3-30}$$

・ **後退矩形近似**

$$C(z) = \frac{\omega_0 K}{\omega_0 + \dfrac{z-1}{Tz}} = \frac{zT\omega_0 K}{(\omega_0 T + 1)z - 1} = \frac{T\omega_0 K}{\omega_0 T + 1 - z^{-1}}$$

$U(z) = C(z)E(z)$ より， $(\omega_0 T + 1 - z^{-1})U(z) = T\omega_0 K\,E(z)$

$$\therefore\ U(z) = \frac{1}{\omega_0 T + 1}z^{-1}U(z) + \frac{\omega_0 T K}{\omega_0 T + 1}E(z)$$

差分方程式で表わすと，

$$u(k)=\frac{1}{\omega_0 T+1}u(k-1)+\frac{\omega_0 T\,K}{\omega_0 T+1}e(k) \tag{3-31}$$

・　台形近似

$$C(z)=\frac{\omega_0 K}{\omega_0+\dfrac{2(z-1)}{T(z+1)}}=\frac{\omega_0(T\,z+T)K}{(\omega_0 T+2)z+\omega_0 T-2}=\frac{\omega_0(T+Tz^{-1})K}{\omega_0 T+2+(\omega_0 T-2)z^{-1}}$$

$U(z)=C(z)E(z)$　より　$\left\{\omega_0 T+2+(\omega_0 T-2)z^{-1}\right\}U(z)=\omega_0(T+Tz^{-1})K\,E(z)$

$$\therefore\ U(z)=\frac{2-\omega_0 T}{2+\omega_0 T}z^{-1}U(z)+\frac{\omega_0 T(1+z^{-1})}{2+\omega_0 T}K\,E(z)$$

差分方程式で表わすと，入力 $e(k)$，出力 $u(k)$ に対し

$$u(k)=\frac{2-\omega_0 T}{2+\omega_0 T}u(k-1)+\frac{\omega_0 T\,K}{2+\omega_0 T}\left\{e(k)+e(k-1)\right\} \tag{3-32}$$

（２） 位相要素

・　前進矩形近似

$$C(z)=K\frac{\dfrac{z-1}{T}+\omega_1}{\dfrac{z-1}{T}+\omega_2}=K\frac{\omega_1 T+z-1}{\omega_2 T+z-1}=K\frac{(\omega_1 T-1)z^{-1}+1}{(\omega_2 T-1)z^{-1}+1}$$

$U(z)=C(z)E(z)$　より $((\omega_2 T-1)z^{-1}+1)U(z)=K((\omega_1 T-1)z^{-1}+1)E(z)$

$$\therefore\ U(z)=(1-\omega_2 T)z^{-1}U(z)+K\,E(z)+K(\omega_1 T-1)z^{-1}E(z)$$

差分方程式で表わすと，

$$u(k)=(1-\omega_2 T)u(k-1)+K\,e(k)+K(\omega_1 T-1)e(k-1) \tag{3-33}$$

・　後退矩形近似

$$C(z)=K\frac{\dfrac{z-1}{T\,z}+\omega_1}{\dfrac{z-1}{T\,z}+\omega_2}=K\frac{\omega_1 T\,z+z-1}{\omega_2 T\,z+z-1}=K\frac{\omega_1 T+1-z^{-1}}{\omega_2 T+1-z^{-1}}$$

$U(z)=C(z)E(z)$　より　$(\omega_2 T+1-z^{-1})U(z)=K(\omega_1 T+1-z^{-1})E(z)$

$$\therefore\ U(z)=\frac{1}{\omega_2 T+1}z^{-1}U(z)+\frac{K(\omega_1 T+1)}{\omega_2 T+1}E(z)-\frac{K}{\omega_2 T+1}z^{-1}E(z)$$

差分方程式で表わすと，

$$u(k)=\frac{1}{\omega_2 T+1}u(k-1)+\frac{K(\omega_1 T+1)}{\omega_2 T+1}e(k)-\frac{K}{\omega_2 T+1}e(k-1) \qquad (3\text{-}34)$$

・　**台形近似**

$$C(z)=K\frac{\dfrac{2(z-1)}{T(z+1)}+\omega_1}{\dfrac{2(z-1)}{T(z+1)}+\omega_2}=K\frac{2(z-1)+\omega_1 T(z+1)}{2(z-1)+\omega_2 T(z+1)}=K\frac{2(1-z^{-1})+\omega_1 T(1+z^{-1})}{2(1-z^{-1})+\omega_2 T(1+z^{-1})}$$

$U(z)=C(z)E(z)$　より

$$\left\{\omega_2 T(1+z^{-1})+2(1-z^{-1})\right\}U(z)=K\left\{\omega_1 T(1+z^{-1})+2(1-z^{-1})\right\}E(z)$$

$$\therefore\ U(z)=\frac{2-\omega_2 T}{2+\omega_2 T}z^{-1}U(z)+\frac{K(2+\omega_1 T_1)}{2+\omega_2 T_2}E(z)-\frac{K(2-\omega_1 T)}{2+\omega_2 T_2}z^{-1}E(z)$$

差分方程式で表わすと，

$$u(k)=\frac{2-\omega_2 T}{2+\omega_2 T}u(k-1)+\frac{K(2+\omega_1 T)}{2+\omega_2 T}e(k)-\frac{K(2-\omega_1 T)}{2+\omega_2 T}e(k-1) \qquad (3\text{-}35)$$

[問題 3-1]　制御器の伝達関数を $C_c(s)=\dfrac{U(s)}{E(s)}=\dfrac{s\omega_0 K}{\omega_0+s}$　（**擬似微分**）について，台形近似を用いて，差分方程式を導け。

（答）$u(k)=\dfrac{2-\omega_0 T}{2+\omega_0 T}u(k-1)+\dfrac{2\omega_0 K}{2+\omega_0 T}\left\{e(k)-e(k-1)\right\}$

3.3　FIR フィルタおよび IIR フィルタ

ディジタル信号処理の分野では，出力信号をフィードバックしない **FIR**(Finite Impulse Response:有限長インパルス応答)フィルタと出力信号をフィードバックする **IIR**(Infinite Impulse Response:無限長インパルス応答)フィルタに分類される。これらのフィルタも当然ノイズ除去などに利用できる。

2 次の **FIR** フィルタは次式で与えられ，そのブロック線図を図 3-4 に示す。

$$u(k)=a_0 e(k)+a_1 e(k-1)+a_2 e(k-2) \qquad (3\text{-}36)$$

伝達関数は次式で与えられる。

$$C(z)=\frac{U(z)}{E(z)}=a_0+a_1 z^{-1}+a_2 z^{-2} \qquad (3\text{-}37)$$

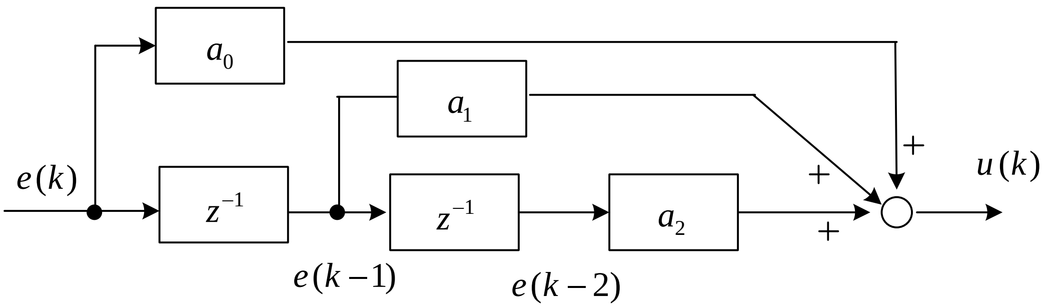


図 3-4　2 次の FIR フィルタ

$a_0 = a_1 = a_2 = 1/3$　の場合，連続する 3 サンプル値の平均をとることを意味し，**移動平均フィルタ**(moving average filter)の一例である。平均をとることでノイズを低減できる。FIR フィルタの次数を決定する方法として**窓関数法**がある(文献(13))。

1 次の IIR フィルタは次式で与えられる。

$$u(k) = b_1 u(k-1) + a_0 e(k) + a_1 e(k-1) \tag{3-38}$$

(3-32)は，この一例である。(3-38)をブロック線図に示すと図のように表わせる。

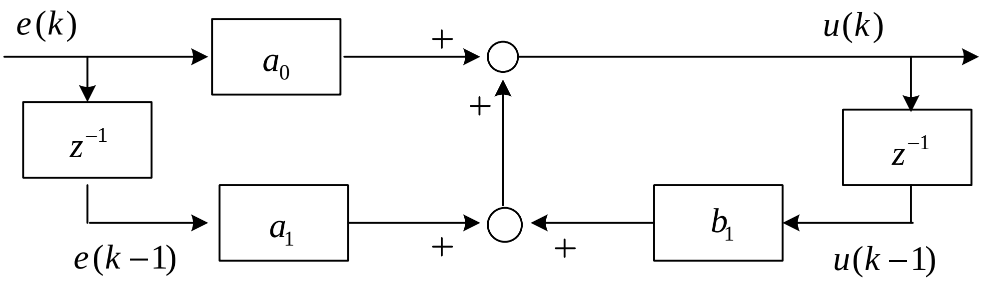


図 3-5　1 次の IIR フィルタ

伝達関数は，次式で与えられる。

$$C(z) = \frac{U(z)}{E(z)} = \frac{a_0 + a_1 z^{-1}}{1 - b_1 z^{-1}} \tag{3-39}$$

急峻なカットオフ特性を有するフィルタを作る場合，IIR フィルタは FIR フィルタより格段に少ない次数で実現できるという特徴がある。

2次の IIR フィルタとして，バタワース低域フィルタ(Butterworth low pass filter)を紹介す

る。ディジタルフィルタの設計の基礎となるアナログフィルタ（プロトタイプフィルタとよばれる）は，次式で与えられる。ω_0 は遮断角周波数である。

$$C_c(s) = \frac{\omega_0^2}{s^2 + \sqrt{2}\,\omega_0\, s + \omega_0^2} \tag{3-40}$$

減衰係数は　$\zeta = \dfrac{1}{\sqrt{2}} = 0.707$　に選ばれる。この値は振幅特性 $|G_c(j\omega)|$ に極値を持たない減衰係数の最小値である。双一次変換を用いると

$$C(z) = \frac{\omega_0^2}{(\frac{2}{T})^2 \frac{(z-1)^2}{(z+1)^2} + \sqrt{2}\,\omega_0 \frac{2}{T}\frac{z-1}{z+1} + \omega_0^2} = \frac{a_0(1+2z^{-1}+z^{-2})}{1-b_1 z^{-1} - b_2 z^{-2}}$$

ここで，$a_0 = \dfrac{(\omega_0 T)^2}{4 + 2\sqrt{2}\,\omega_0 T + (\omega_0 T)^2}$

$$b_1 = \frac{8 - 2(\omega_0 T)^2}{4 + 2\sqrt{2}\,\omega_0 T + (\omega_0 T)^2},\ b_2 = \frac{2\sqrt{2}\,\omega_0 T - 4 - (\omega_0 T)^2}{4 + 2\sqrt{2}\,\omega_0 T + (\omega_0 T)^2}$$

差分方程式で表わすと，

$$u(k) = b_1 u(k-1) + b_2 u(k-1) + a_0(e(k) + 2e(k-1) + e(k-2)) \tag{3-41}$$

2 次のローパスフィルタは，1 次のローパスフィルタに比べて急峻なカットオフ特性が得られるが，位相の遅れには注意が必要だろう。

3.4　微分方程式で与えられた制御則の離散化

　制御器の式が最初から微分方程式で与えられている場合には，直接，前進オイラー法，後退オイラー法または台形公式を使って差分近似する方が簡単である。台形公式が最も精度が良いので，一般的に利用されているであろう。非線形微分方程式（これは状態オブザーバを利用した制御などで現れる）を差分近似する場合，後退オイラー法または台形公式を使うと，非線形方程式を解く必要があるので離散化しにくいことがある。この場合には精度は悪くても前進オイラー法が使われているだろう。精度を良くしたい場合には，前進オイラー法と台形公式を組み合わせた 2 次の Runge-Kutta 法（修正オイラー法）がある。サンプリング周期が非常に短い場合には，これらの方法に大きな差はなくなるだろうが，先に述べた情報落ちの問題がある。

　一般の非線形微分方程式を前進オイラー法, 後退オイラー法, 台形公式, 2 次の Runge-Kutta

法で差分近似する。

$$\frac{d\boldsymbol{x}(t)}{dt} = \boldsymbol{f}(\,\boldsymbol{x}(t)\,, \boldsymbol{u}(t),\, t) \tag{3-42}$$

例：
$$\frac{dx_1(t)}{dt} = f_1 = -2x_1^2(t) + u_1(t) + \sin 2t$$

$$\frac{dx_2(t)}{dt} = f_2 = -2x_1(t)x_2(t) + u_1(t)u_2(t)$$

① **前進オイラー法**

$$\frac{\boldsymbol{x}(k) - \boldsymbol{x}(k-1)}{T} = \boldsymbol{f}(\boldsymbol{x}(k-1), \boldsymbol{u}(k-1), (k-1)T)$$

$$\therefore \quad \boldsymbol{x}(k) = \boldsymbol{x}(k-1) + T\,\boldsymbol{f}(\boldsymbol{x}(k-1), \boldsymbol{u}(k-1), (k-1)T) \tag{3-43}$$

常に容易に計算できる。

② **後退オイラー法**

$$\frac{\boldsymbol{x}(k) - \boldsymbol{x}(k-1)}{T} = \boldsymbol{f}(\,\boldsymbol{x}(k), \boldsymbol{u}(k), kT) \tag{3-44}$$

右辺にも $\boldsymbol{x}(k)$ が含まれるので一般に方程式を解かないといけない。

③ **台形公式**

$$\frac{\boldsymbol{x}(k) - \boldsymbol{x}(k-1)}{T} = \frac{1}{2}\left\{\boldsymbol{f}(\,\boldsymbol{x}(k), \boldsymbol{u}(k), kT) + \boldsymbol{f}(\,\boldsymbol{x}(k-1), \boldsymbol{u}(k-1), (k-1)T)\right\} \tag{3-45}$$

右辺にも $\boldsymbol{x}(k)$ が含まれるので一般に方程式を解かないといけない。

④ **2 次の Runge-Kutta(ルンゲクッタ)法（修正オイラー法，ホイン法）**

$$\hat{\boldsymbol{x}}(k) = \boldsymbol{x}(k-1) + T\,\boldsymbol{f}(\boldsymbol{x}(k-1), \boldsymbol{u}(k-1), (k-1)T) \tag{3-46}$$

$$\boldsymbol{x}(k) = \boldsymbol{x}(k-1) + \frac{T}{2}(\,\boldsymbol{f}(\,\hat{\boldsymbol{x}}(k), \boldsymbol{u}(k), kT) + \boldsymbol{f}(\,\boldsymbol{x}(k-1), \boldsymbol{u}(k-1), (k-1)T)) \tag{3-47}$$

台形公式で微係数 $\boldsymbol{f}$ を求めるときに， $\boldsymbol{x}(k)$ の代わりにオイラー法で求めた $\hat{\boldsymbol{x}}(k)$ を使用する。方程式を解く必要はない。

[例題 3-1]　次の微分方程式を前進オイラー法, 後退オイラー法, 台形公式で差分近似せよ。

$$\frac{dx(t)}{dt} = -a\,x(t) + b\,u(t)$$

(解)

① **前進オイラー法**

$$\frac{x(k) - x(k-1)}{T} = -a\,x(k-1) + b\,u(k-1) \qquad \therefore \quad x(k) = (1 - aT)\,x(k-1) + bT\,u(k-1)$$

② **後退オイラー法**

$$\frac{x(k)-x(k-1)}{T}=-a\,x(k)+b\,u(k)$$

$$\therefore \quad x(k)=\frac{1}{1+aT}x(k-1)+\frac{bT}{1+aT}u(k)$$

② **台形公式**

$$\frac{x(k)-x(k-1)}{T}=\frac{1}{2}\left\{-a\,x(k)+b\,u(k)-a\,x(k-1)+b\,u(k-1)\right\}$$

$$\therefore \quad x(k)=\frac{2-aT}{2+aT}x(k-1)+\frac{bT}{2+aT}u(k)+\frac{bT}{2+aT}u(k-1)$$

［例題 3-2］ 制御器の伝達関数を $\dfrac{U(s)}{E(s)}=K\dfrac{s+\omega_1}{s+\omega_2}$ とする。微分方程式に戻して，台形公式を適用し差分方程式を導け。

（解）この微分方程式については，次式が成立する。

$$\frac{d}{dt}\left(u(t)-K\,e(t)\right)=K\,\omega_1 e(t)-\omega_2 u(t)$$

これに，台形公式を適用して，次式が得られる。

$$\frac{u(k)-K\,e(k)-(u(k-1)-K\,e(k-1))}{T}$$
$$=\frac{1}{2}\left\{K\,\omega_1 e(k)-\omega_2 u(k)+K\,\omega_1 e(k-1)-\omega_2 u(k-1)\right\}$$

故に $$u(k)=\frac{2-\omega_2 T}{2+\omega_2 T}u(k-1)+\frac{K(2+\omega_1 T)}{2+\omega_2 T}e(k)-\frac{K(2-\omega_1 T)}{2+\omega_2 T}e(k-1)$$

［問題 3-2］ 次の非線形微分方程式を 2 次のルンゲクッタ法で離散化せよ。

$$\frac{dx}{dt}=x\sin(2t)+1$$

（解） $$\hat{x}(k)=x(k-1)+T\{x(k-1)\sin(2(k-1)T)+1\}$$

$$x(k)=x(k-1)+\frac{T}{2}\{\,\hat{x}(k)\sin(2kT)+1+x(k-1)\sin(2(k-1)T)+1\}$$

3.5 リミッタ

制御対象への入力 $u(k)$ は装置の容量によって，その大きさが制限される。そのため，次式の**リミッタ**（limiter）が設けられる。

$$|u(k)| \le u_{\max} \tag{3-48}$$

ところが，このリミッタをどのようにプログラムするかで，応答に大きな違いが生じる。PI 制御の場合を例にとり，以下に詳しく述べる。

$$u(k) = K_P e(k) + K_I \sum_{m=0}^{k} e(m)T \qquad \text{（後退矩形近似）} \tag{3-13}$$

を用いてプログラムする場合，比例制御と積分制御を分けて，以下の様にプログラムしたとしよう。なお，<u>このプログラムはサンプリング周期 T ごとに繰り返し実行される</u>。この様子を図 3-6 に示す。積分の計算はメモリに値を蓄えることで容易に行える。

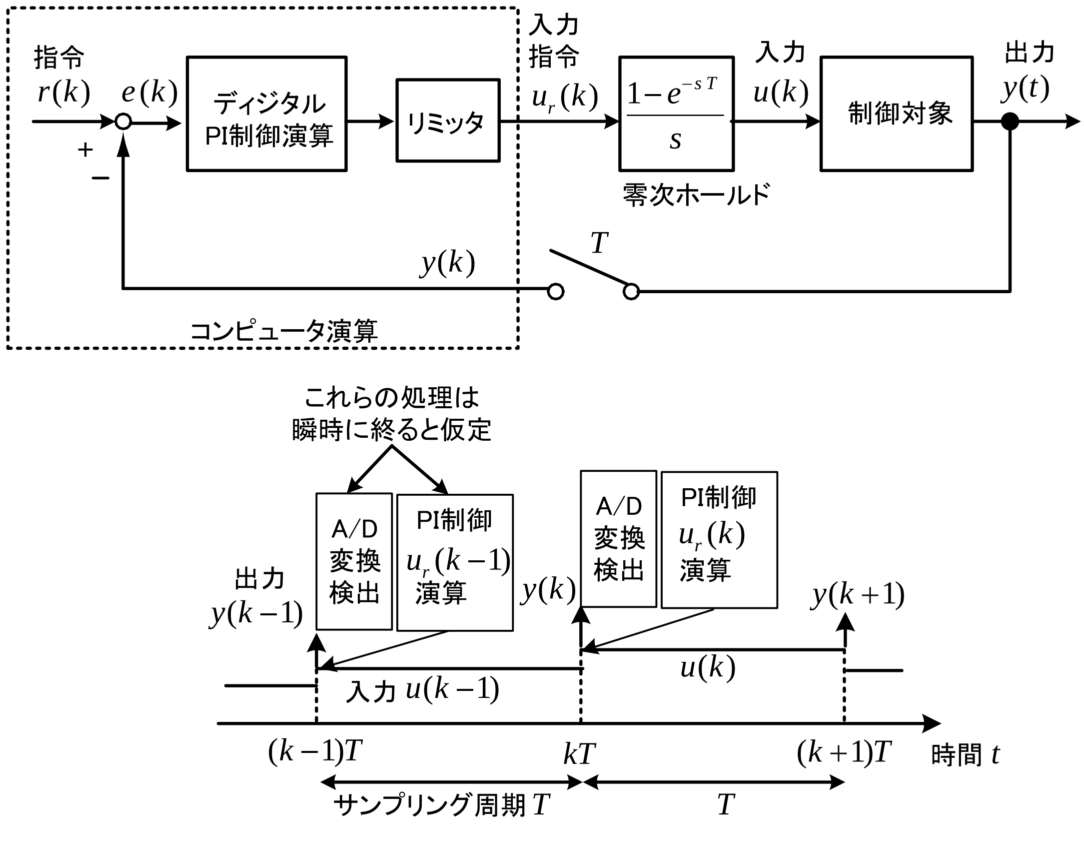


図 3-6　ディジタル PI 制御の実行過程

<u>リミッタを含む PI 制御プログラム（その 1）</u>

$e = r - y$;	偏差 e の計算	①
$u_p = K_p * e$;	比例量計算	②
$u_{inew} = u_{iold} + K_I * T * e$;	積分量計算	③
$u = u_p + u_{inew}$;	入力指令＝比例量＋積分量	④

$if\ (u > u_{\max})\quad u = u_{\max}\quad ;$ リミッタ上限 ⑤

$if\ (u < (-u_{\max}))\quad u = -u_{\max}\quad ;$ リミッタ下限 ⑥

$u_{iold} = u_{inew}\quad ;$ 積分量更新（次回使用） ⑦

（注意）一番最初だけ$u_{iold} = 0$として変数を初期化する。

素直なプログラムであるが，これには大きな問題がある。指令値のステップ変化に対する応答についてこの問題を考える。単純にuのみにリミッタを設けているため，uがリミッタで制限されているときでも積分量は制限されることなくどんどん値が蓄積される。このため，uがリミッタを抜けるまでに時間がかかり，出力yに大きなオーバーシュートが生じる。これは**ワインドアップ現象**（windup）と呼ばれている。図 3-7 はモータの PI 速度制御の例である。出力$y[\mathrm{min}^{-1}]$は回転速度，入力$u\,[\mathrm{A}]$はトルク電流で，速度指令rを 1000min^{-1}から 1500min^{-1}さらに 1000min^{-1}に変化させている。入力$u\,[\mathrm{A}]$の最大値は 10A に設定している。1500min^{-1}に上昇させる場合，偏差eが負になる点でu_{inew}は減少し始めるが，それまでの積分でu_{inew}が大きくuはその後もリミッタの上限値となってワインドアップ現象が生じている。その後，1000min^{-1}に減少させる場合には，積分量u_{inew}は−10A に達しておらず，入力uがリミッタにかかる時間は短い。速度のアンダーシュートは，モータの負荷トルク（電流換算でuの 5A 相当）とモータが出す負のトルクとが同じ向きになるためと考えられる（加速時の加速トルク 10-5=5A 相当：減速時の減速トルク-10-5=-15A 相当）。

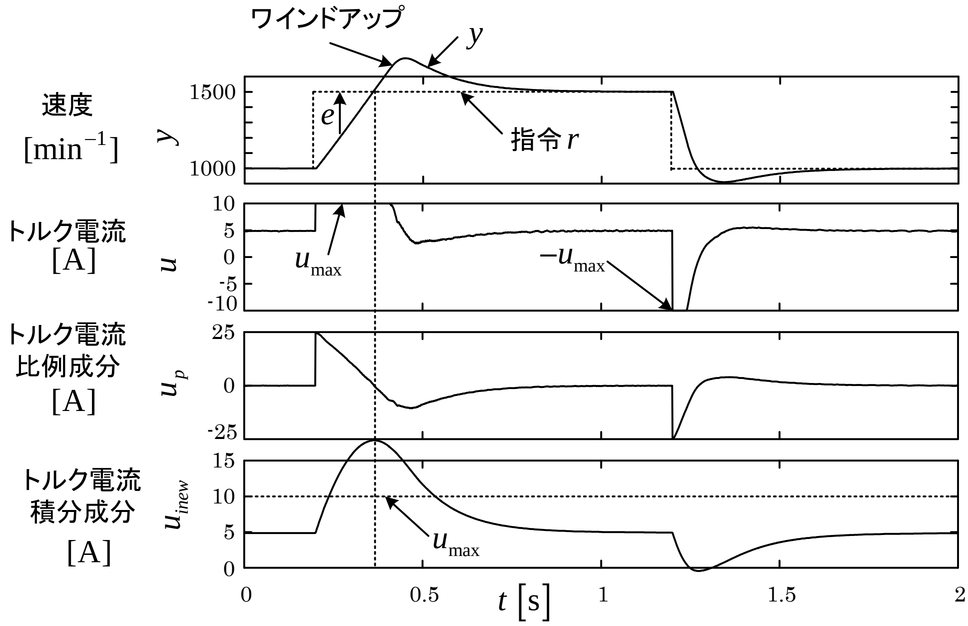


図 3-7 プログラムその 1 の場合

ワインドアップを防ぐためには, u がリミッタにかかったときに積分器に値をためないようする必要がある。つまり u_{inew} にも何らかのリミッタをかければよい。

(3-16)の差分方程式を用いる場合，以下のプログラムで $u(k)$ にリミッタをかけた場合には，ワインドアップ現象は生じない。

リミッタを含む PI 制御プログラム（その 2）

$e = r - y$;	偏差 e の計算	①
$u = u_{old} + K_P * (e - e_{old}) + K_I * T * e$;	入力指令計算	②
$if\ (u > u_{\max})\quad u = u_{\max}$;	リミッタ上限	③
$if\ (u < (-u_{\max}))\quad u = -u_{\max}$;	リミッタ下限	④
$u_{old} = u$;	入力指令更新（次回使用）	⑤
$e_{old} = e$;	偏差更新（次回使用）	⑥

（注意）一番最初だけ $u_{old} = 0,\ e_{old} = 0$ として変数を初期化すること。

ワインドアップが生じないのは，$u(k-1)$ に偏差の項を加えて, 新しく $u(k)$ が求まるため, $u(k)$ がリミッタにかかり変化しなければ, その間は値をためる量が存在しないことによる。ただし，⑤を②の後に書くと，u_{old} に値がたまりワインドアップ現象は起きる。(3-16)の注意点としては，指令値が大きくステップ変化するとき，出力 y を速く立ち上げるためには入力 u は $u_{\max}$ となることが望ましいが，$e(k)$ と $e(k-1)$ の差を K_p 倍することになるため, u が $u_{\max}$ とならないことが考えられる。ただし，穏やかな変化を望むならばそれはメリットにもなり得る。図 3-8 はモータの PI 速度制御の例である。図のように比例成分 $u_p = K_P\, e$ が $u_{\max}$ を超えても，u が $u_{\max}$ となっていない。

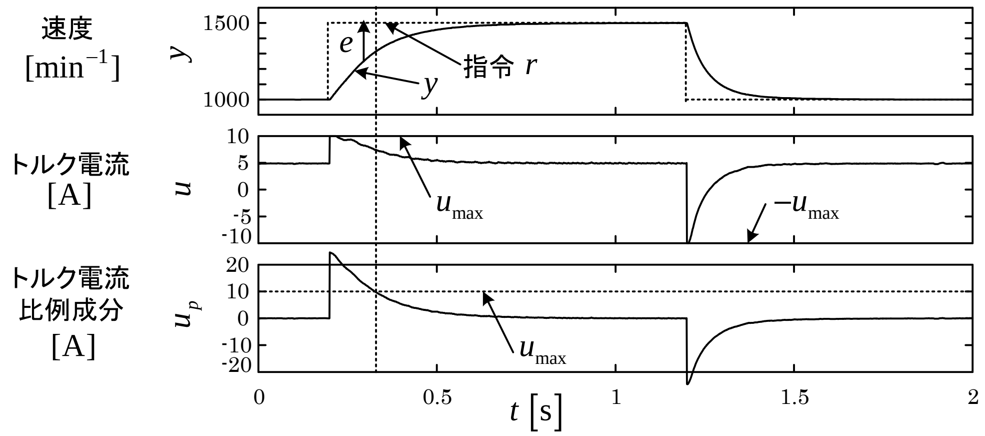


図 3-8 プログラムその２の場合

応答を速くするために，u だけでなく比例成分 $u_p = K_P\,e$ が制限値を超えたら，最大値または最小値を出力することが考えられる。以下に，そのプログラムを示す。

リミッタを含む PI 制御プログラム（その 3）

$e = r - y$;	偏差 e の計算	①
$u = u_{old} + K_P * (e - e_{old}) + K_I * T * e$;	入力指令計算	②
$if\ (u > u_{\max} \,\|\|\, K_P * e > u_{\max})\quad u = u_{\max}$;	リミッタ上限	③
$if\ (u < (-u_{\max}) \,\|\|\, K_P * e < (-u_{\max}))\quad u = -u_{\max}$;	リミッタ下限	④
$u_{old} = u$;	入力指令更新（次回使用）	⑤
$e_{old} = e$;	偏差更新（次回使用）	⑥

（注意）一番最初だけ $u_{old} = 0,\ e_{old} = 0$ として変数を初期化すること。

　　　$||$ は OR の演算子である。

図 3-9 はモータの PI 速度制御の例である。図のように比例成分 $u_p = K_P\,e$ が $u_{\max}$ を超えると，u が $u_{\max}$ となって図 3-8 に比べて速度の応答が速くなっている。ただ，減速の場合には速度のアンダーシュートが生じている。プログラムその 1 は明らかに良くないが，その２とその３は目的によって選択すべきであろう。

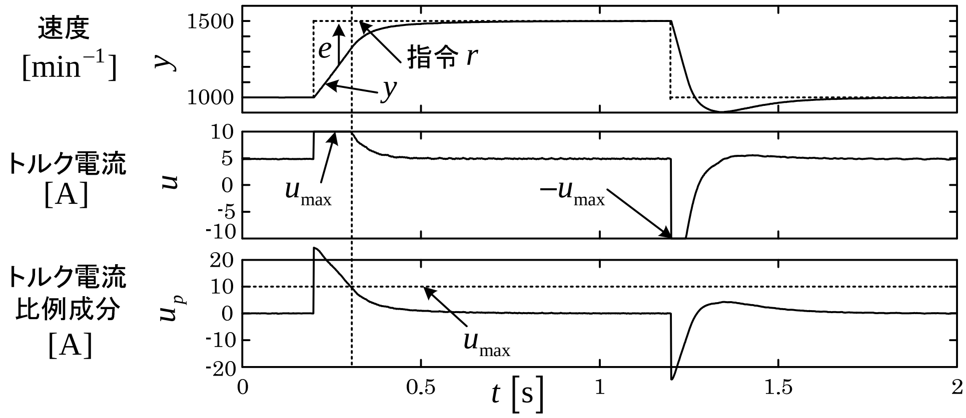


図 3−9 プログラムその３の場合

(3-13)と(3-16)は，リミッタにかからなければ等価であるが，リミッタにかかると，その処理の仕方によって応答が大きく異なるので注意して欲しい。

［問題 3−3］　(3-13)のプログラムその１ではワインドアップが生じるのに，数学的に等価な(3-16) のプログラムその２ではワインドアップが生じない。何故か。

第４章　安定解析

ブロック線図で表されたディジタル制御系の安定判別法を説明する。ｚの双一次変換を行うと連続系の安定解析でよく知られたラウスの方法が利用でき便利である。また，ディジタル制御特有の制御法であるデッドビート制御についても例題を通して説明する。

4.1　安定条件

ｚ変換を用いた伝達関数により表示した図 4-1 のディジタル制御系について考える。

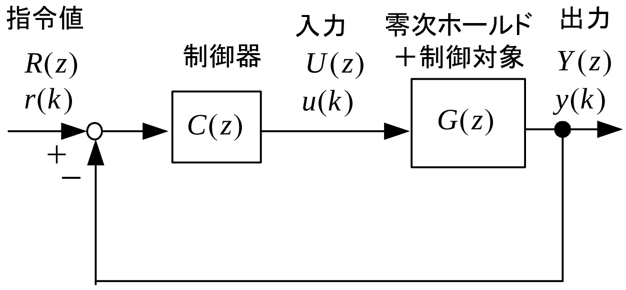


図 4-1 ディジタル制御系のブロック線図

図より，**閉ループパルス伝達関数**(closed loop pulse transfer function) $W(z)$ は次式で与えられる。

$$W(z) \equiv \frac{Y(z)}{R(z)} = \frac{C(z)G(z)}{1+C(z)G(z)} \tag{4-1}$$

一般に，$W(z)$ は次式のように表せる。

$$W(z) = \frac{b_1 z^{n-1} + b_2 z^{n-2} + \cdots + b_{n-1}z + b_n}{z^n + a_1 z^{n-1} + \cdots + a_{n-1}z + a_n} \tag{4-2}$$

$W(z)$ の分母を 0 と置いた

$$z^n + a_1 z^{n-1} + \cdots + a_{n-1}z + a_n = 0 \tag{4-3}$$

は**特性方程式(**characteristic equation)と呼ばれ，その n 個の根を**特性根**または**極**という。

いま(4-3)が，相異なる n 個の根 $p_1, p_2, \cdots p_n$（複素根でもよい）をもつものとすれば，$Y(z)$ は次のように部分分数展開することができる。

$$\frac{Y(z)}{z} = \frac{W(z)R(z)}{z} = \frac{q_1}{z-p_1} + \frac{q_2}{z-p_2} + \cdots + \frac{q_n}{z-p_n} + \left\{ \frac{R(z)}{z}\text{の根による展開項} \right\} \tag{4-4}$$

ここで，q_i は次式で求められる。

$$q_i = \lim_{z \to p_i}(z - p_i)\frac{W(z)R(z)}{z} \qquad (i = 1, 2, \cdots n) \tag{4-5}$$

(4-4)の両辺に z を掛けて逆 z 変換すると，出力 $y(k)$ は次式で与えられる。

$$y(k) = q_1(p_1)^k + q_2(p_2)^k + \cdots + q_n(p_n)^k$$

$$+\{ R(z) \text{ の根による展開項の逆 z 変換}\} \tag{4-6}$$

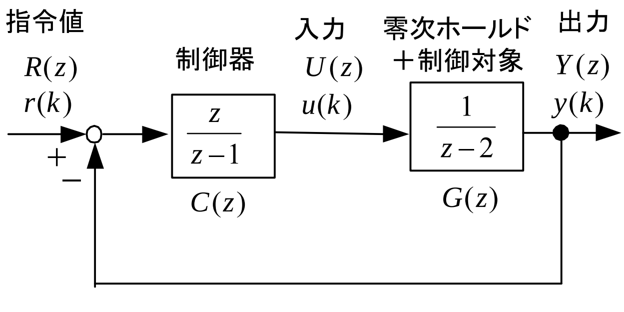
{ $R(z)$ の根による展開項の逆 z 変換}の項は，応答の定常項に相当し安定性とは関係ない。式(4-6)より，$k \to \infty$ のとき，$y(k)$ が有限であるためにはすなわち安定であるためには，すべての根 $p_1, p_2, \cdots p_n$ の絶対値がどれも 1 以下となることである。

安定条件(stability condition)： ディジタル制御系で閉ループパルス伝達関数の分母を 0 とおいた式を特性方程式という．この特性方程式の根を p_i $(i = 1, 2, \cdots n)$（重根も可）としたとき，安定であるための必要十分条件は

$$|p_i| < 1 \quad (i = 1, 2, \cdots n) \tag{4-7}$$

である。これは，ディジタル制御で最も重要な公式である。

[例題 4-1]　図のディジタル制御系で, $r(k) = 1$ なる指令を与えたとき, 出力 $y(k)$ を求めよ。このとき，特性方程式の根と安定性の関係を調べよ。



（解）閉ループパルス伝達関数 $W(z)$ は次式で求まる.

$$W(z) = \frac{CG}{1 + CG} = \frac{z}{z^2 - 2z + 2} \tag{①}$$

指令値を z 変換して，

$$R(z)=\frac{z}{z-1} \qquad ②$$

である。したがって，

$$Y(z)=W(z)R(z)=\frac{z}{z^2-2z+2}\frac{z}{z-1} \qquad ③$$

特性方程式 $z^2-2z+2=0$ の根を p_1, p_2 とすると，

$$p_1=1+j=\sqrt{2}e^{j\pi/4},\ p_2=1-j=\sqrt{2}e^{-j\pi/4} \qquad ④$$

となる。部分分数に展開して，

$$\frac{Y(z)}{z}=\frac{z}{(z-1)(z-p_1)(z-p_2)}=\frac{a}{z-1}+\frac{b}{z-p_1}+\frac{c}{z-p_2} \qquad ⑤$$

ここで，

$$a=(z-1)\frac{Y(z)}{z}\bigg|_{z=1}=1$$

$$b=(z-p_1)\frac{Y(z)}{z}\bigg|_{z=p_1}=\frac{p_1}{(p_1-1)(p_1-p_2)}=-\frac{1+j}{2} \qquad ⑥$$

$$c=(z-p_2)\frac{Y(z)}{z}\bigg|_{z=p_2}=\frac{p_2}{(p_2-1)(p_2-p_1)}=-\frac{1-j}{2} \qquad ⑦$$

⑤より，

$$Y(z)=\frac{z}{z-1}+\frac{bz}{z-p_1}+\frac{cz}{z-p_2} \qquad ⑧$$

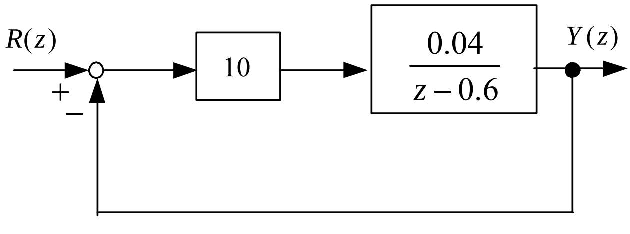
であり，これを逆 z 変換して，次式を得る。

$$y(k)=1+b(p_1)^k+c(p_2)^k$$

$$=1+(\sqrt{2})^k\sin(k\frac{\pi}{4})-(\sqrt{2})^k\cos(k\frac{\pi}{4}) \qquad (k=0,1,2,\cdots) \qquad ⑨$$

$k\to\infty$ のとき，$y(k)\to\infty$ となり系は不安定である。これは，$|p_1|=|p_2|=\sqrt{2}>1$ であることから理解できる。

[問題 4-1]　図のディジタル制御系を安定判別せよ。次に，$r(k)=1 \quad (k=0,1,2,\cdots)$ のとき，出力 $y(k)$ を求めよ。

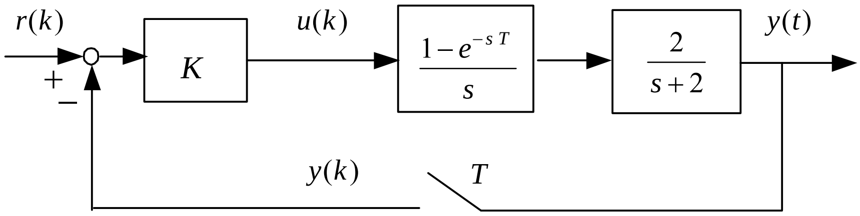


(答) $\dfrac{Y(z)}{R(z)} = \dfrac{0.4}{z-0.2}$ 特性方程式 $z-0.2=0$

$\therefore z = 0.2$ $|z|<1$ なので安定

$$\frac{Y}{z} = \frac{1}{2}\frac{1}{z-1} - \frac{1}{2}\frac{1}{z-0.2} \qquad \therefore\ y(k) = \frac{1}{2} - \frac{1}{2}(0.2)^k$$

[問題 4-2] 図のディジタル制御系が安定である K の条件を求めよ。



(答) (2-17)より $\dfrac{Y(z)}{U(z)} = \dfrac{1-a}{z-a}$ ただし，$a = e^{-2T} < 1$

特性方程式 $1 + K\dfrac{1-a}{z-a} = 0$

安定条件 $|z| = |a - K(1-a)| < 1 \quad \therefore -1 < K < \dfrac{1+a}{1-a}$

閉ループパルス伝達関数 $W(z)$ の極によって，出力 $y(k)$ のおおよその見当がつく。指令値を 0 としたときの応答（過渡項のみの応答 : (4-6)で最初の n 項の応答）は図 4-2 から推定できる。すなわち，z 平面で，$p_1, p_2, \cdots p_n$ のどれか 1 つでも単位円の外にあると応答は徐々に増加し不安定となる。単位円の外に根が存在するとき，正の実軸上だと振動しないで発散するが，複素数だと振動しながら発散し，負の実軸上だとサンプリング周期ごとに振動して発散する。単位円の内に根が存在するとき，正の実軸上だと振動しないで収束するが，複素数だと振動しながら収束し，負の実軸上だとサンプリング周期ごとに振動して収束する。なお，複素根は必ず共役根として存在する。根がたくさんある場合には，絶対値が最も大きい根に支配される。これを**代表根**(dominant root)という。これは，(4-6)で，絶対値が小さいとすぐ減衰して 0 となるためである。しかし，(4-6)で q_i が非常に小さいと，その項の影響がでるまでにかなりの時間を要する場合がある。これは，分子に分母と打ち消すような根（零点）が存在する場合におこる。したがって，応答を実際に計算して確認することが必要となる。ただ，q_i が非常に小さいといっても，不安定な極に対してはいずれ発散するので，安定判別の条件が変わることはない。

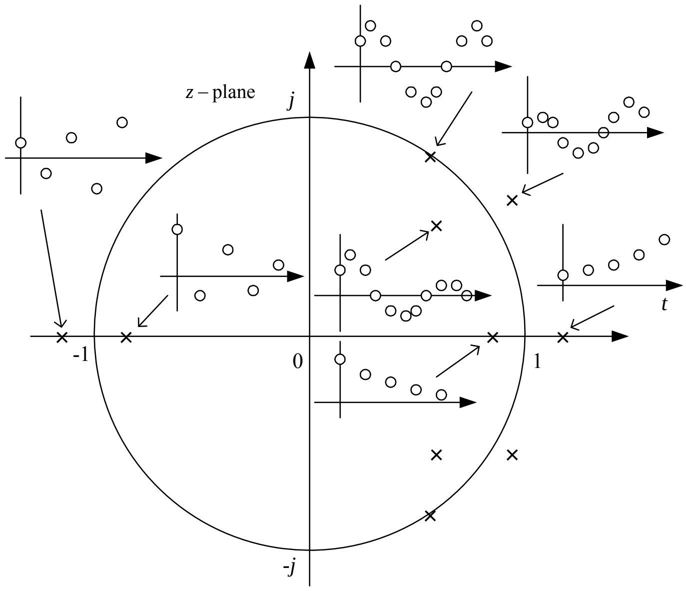


図 4-2 $W(z)$ のいろいろの極に対する応答

4.2　安定判別 (stability criterion)

安定判別は特性方程式（4-3）の根を求めることで行えるが，次数が高くなると計算機による数値計算で求める必要がある。そこで，連続系で用いたラウスの安定判別法が利用できる方法を紹介する。これは，z を次式で定義する s（ここの章のみ）に変換する。

$$z \equiv \frac{1+s}{1-s} \tag{4-8}$$

逆に，(4-8)より

$$s = \frac{z-1}{z+1} \tag{4-9}$$

である。これは，**双1次変換**(bilinear transformation)と呼ばれる（(3-29) 参照）。(4-8)を用いると，図 4-3 に示すように，z 平面の単位円内の領域が，s 平面の左半平面に写像される。

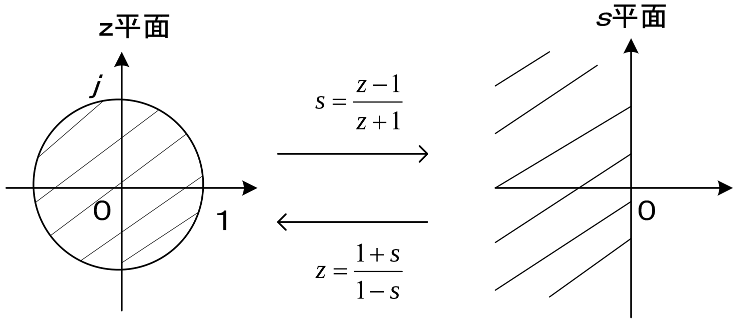


図 4-3　双 1 次変換

(4-3)に，(4-8)を代入して，

$$(\frac{1+s}{1-s})^n + a_1(\frac{1+s}{1-s})^{n-1} + \cdots + a_{n-1}(\frac{1+s}{1-s}) + a_n = 0 \tag{4-10}$$

これを，s のべき乗の形に整理すると次式のように書ける。

$$c_0 s^n + c_1 s^{n-1} + \cdots + c_{n-1}s + c_n = 0 \tag{4-11}$$

これに，**ラウスの方法** (Routh stability criterion)を適用して安定判別する。ラウスの方法については例を示しておく。

[例題 4-2]　連続系の特性方程式が　$s^4 + 5s^3 + 8s^2 + 16s + 20 = 0$　のとき安定判別せよ。

(解)ラウスの表は,

s^4	$c_0 = 1$	$c_2 = 8$	$c_4 = 20$
s^3	$c_1 = 5$	$c_3 = 16$	0
s^2	$d_1 = \frac{5\times8-16\times1}{5} = 4.8$	$d_2 = \frac{5\times20-0\times1}{5} = 20$	0
s	$e_1 = \frac{4.8\times16-5\times20}{4.8} = -4.83$	0	
s^0	20		

第一列の符号が 2 回変わるので(4.8 から-4.83 と-4.83 から 20)，不安定根が 2 個存在する。

なお，図 4-3 に示した<u>双 1 次変換は，安定判別をするためにだけ用いる変換</u>であり，連続系の根（s）とディジタル制御系の根（z）がそのような関係にあることを意味しない（6.5 で詳しく述べる)。従って，誤解を避けるため s の替わりに別の記号を用いることもある。

[問題 4-3]　　$z = \frac{1+s}{1-s}$ で，$s = j\omega \ (\omega : -\infty \to +\infty)$ のとき，z の軌跡を描け。

(答) $z = x + j\,y$ とおくと，$x + j\,y = \frac{1+j\omega}{1-j\omega} = \frac{1-\omega^2+j2\omega}{1+\omega^2}$　$\therefore$　$x^2 + y^2 = 1$(円)。

[問題 4-4]　　閉ループパルス伝達関数が

$$W(z) = \frac{cz+d}{z^2+az+b}$$

で与えられている。安定となるパラメータの範囲を図示せよ。

（答）特性方程式に双 1 次変換を利用し，次式を得る。

$$(\frac{1+s}{1-s})^2 + a(\frac{1+s}{1-s}) + b = 0$$

$$(1-a+b)s^2 + (2-2b)s + 1 + a + b = 0$$

ラウスの方法を適用して，2 次系の安定条件は係数が全て同符号であればよい。よって
$1+a+b>0,\ b<1,\ 1-a+b>0$　①　または　$1+a+b<0,\ b>1,\ 1-a+b<0$　②

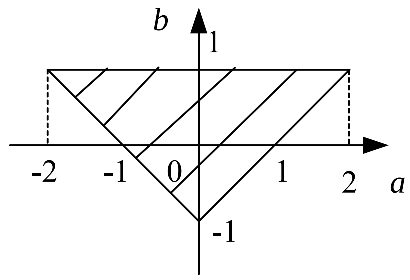
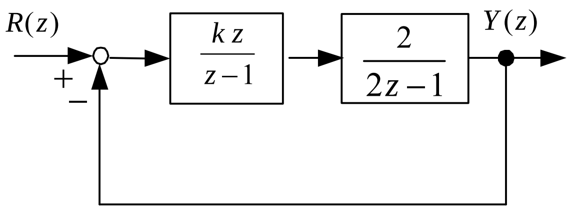


図 4-4　安定領域（②の解はない）

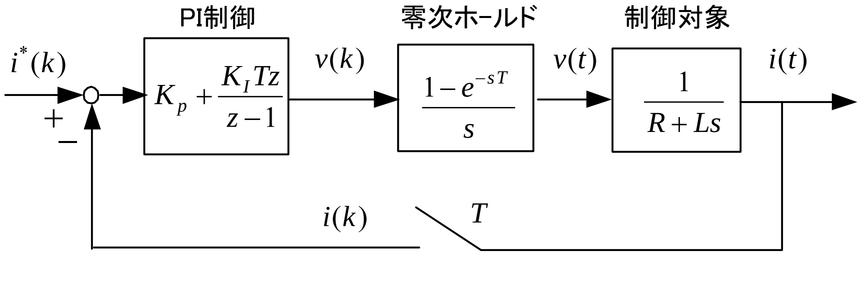
[問題 4-5]　図のディジタル制御系が安定となる k の範囲を求めよ。



（答）　$\dfrac{Y(z)}{R(z)} = \dfrac{2kz}{2z^2 - 3z + 2kz + 1}$　　特性方程式　$2z^2 - 3z + 2kz + 1 = 0$

$2(\frac{1+s}{1-s})^2 + (2k-3)\frac{1+s}{1-s} + 1 = 0$　　$(6-2k)s^2 + 2s + 2k = 0$　安定条件　$0 < k < 3$

[例題 4-3]　図に示すＲＬ回路のディジタル電流ＰＩ制御系で，安定となるためのＰＩ制御ゲインの条件を求め図示せよ。（ＰＩ制御には，後退矩形近似を用いている。）



（解）零次ホールドと制御対象をまとめた $v(k),i(k)$ 間のパルス伝達関数は

$$\frac{I(z)}{V(z)} = Z\left(\frac{1-e^{-sT}}{s}\frac{1}{R+Ls}\right)$$

$$= \frac{1}{R}\frac{1-a}{z-a}$$

ただし，$e^{-RT/L} = a$ とおく。

従って，$i^*(k),i(k)$ 間の閉ループ伝達関数は次式で求められる。

$$\frac{I(z)}{I^*(z)} = \frac{(K_P + \frac{K_I T z}{z-1})\frac{1}{R}\frac{1-a}{z-a}}{1+(K_P + \frac{K_I T z}{z-1})\frac{1}{R}\frac{1-a}{z-a}}$$

$$= \frac{(K_p + K_I T)(1-a)z - K_p(1-a)}{Rz^2 + \{(K_p + K_I T)(1-a) - R - Ra\}z + aR - K_p(1-a)}$$

分母を０とおいた特性方程式に双１次変換を行う。

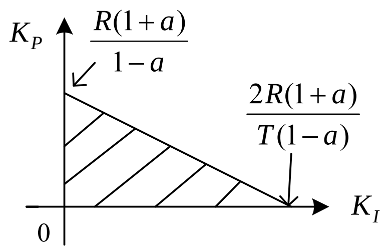
$$R\left(\frac{1+s}{1-s}\right)^2 + \{(K_p + K_I T)(1-a) - R - Ra\}\left(\frac{1+s}{1-s}\right) + aR - K_p(1-a) = 0$$

ゆえに，

$$\{2R(1+a) - (2K_P + K_I T)(1-a)\}s^2 + 2(R+K_P)(1-a)s + K_I T(1-a) = 0$$

２次系だから安定条件は係数が全て同符号であればよい。$a<1$ であり，一般に K_P, K_I は正に選ぶので，このとき安定条件は次式で与えられる。安定領域は図の斜線の領域である。

$$2K_P + K_I T < \frac{2R(1+a)}{1-a}$$



ディジタル制御特有の制御に**有限整定制御**（finite time settling control）または**デッドビート制御**（dead-beet control）がある。これは，特性方程式の根を全て０に設定することで実現できる。この例題で説明しておこう。

この例題では，特性方程式は次式で与えられる。

$$Rz^2 + \left\{(K_p + K_I T)(1-a) - R - Ra\right\}z + aR - K_p(1-a) = 0$$

従って，有限整定制御を行うためには，z^2 以外の係数を０とおいて，

$$(K_p + K_I T)(1-a) - R - Ra = 0$$

$$aR - K_p(1-a) = 0$$

よって，両式を解いて，

$$K_p = \frac{aR}{1-a} \quad , \quad K_I = \frac{R}{T(1-a)}$$

とすればよい。実際に，電流の指令値を $i^*(k)$ 　のステップ変化に対する応答を求めてみよう。K_p, K_I を閉ループ伝達関数に代入して，

$$\frac{I(z)}{I^*(z)} = \frac{(1+a)z-a}{z^2}$$

$i^*(k)=1$ より，$I^*(z) = \dfrac{z}{z-1}$ 　だから

$$\frac{I(z)}{z} = \frac{(1+a)z-a}{z^2(z-1)} = \frac{A}{z} + \frac{B}{z^2} + \frac{C}{z-1}$$

$$= \frac{(A+C)z^2 + (B-A)z - B}{z^2(z-1)}$$

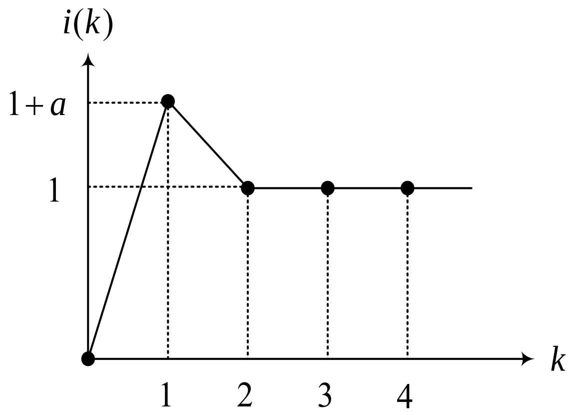
係数を比較して，$B=a$ ，$A=-1$ ，$C=1$ 　を得る。

$$\therefore \quad I(z) = -1 + \frac{a}{z} + \frac{z}{z-1}$$

逆 z 変換して，

$$i(k) = -\delta_{k,0} + a\delta_{k,1} + 1$$

なお，$\delta_{k,j}$ は $k=j$ 　のときのみ1である。



従って，

$i(0) = -1+0+1 = 0$

$i(1) = -0+a+1 = 1+a$

$i(2) = -0+0+1 = 1$

以下，全て1となる。

従って，図の様に $k=2$ で $i(k)$ は指令値と一致し，整定している。

なお，有限整定制御が理想的に行えるのは，制御対象のパラメータが正確に求まっていることが前提となっている。

なお，比例制御の場合には，$K_p = \dfrac{aR}{1-a}$ 　で有限整定制御が実現でき，

$$i(k) = -a\delta_{k,0} + a$$

すなわち，

$i(0)=0$ ，$i(1) = i(2) = \cdots = a$

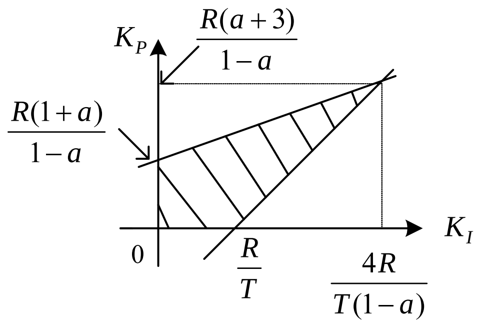
となる。1サンプリングで収束する。$e^{-RT/L}=a$　であるから，サンプリング周期Tを小さく選ぶと$a\simeq 1$　となり，定常偏差を小さくできる。しかし，このとき比例ゲインK_pは非常に大きくなるので，ノイズの影響を受け易くなる点は注意が必要である。

[問題 4-6]　例題 4-3 で，ＰＩ制御器を前進矩形近似および台形近似する場合の安定となるパラメータの範囲を図示せよ。K_P, K_I は正とする。また，アナログＰＩ制御したときの安定領域も求め，ディジタルＰＩ制御で$T\to 0$のときの安定領域と一致することを確認せよ。

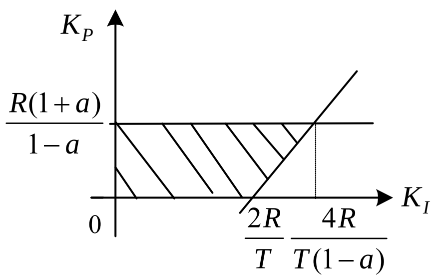
（答）安定条件　前進矩形近似：$K_p<\dfrac{T}{2}K_I+\dfrac{R(1+a)}{1-a}$，$K_p>TK_I-R$

台形近似：$K_p<\dfrac{R(1+a)}{1-a}$，$K_p>\dfrac{T}{2}K_I-R$

アナログＰＩ制御：$K_p>0$，$K_I>0$



前進矩形近似



台形近似

[問題 4-7]　例題 4-3 で，制御系が安定の場合，電流指令のステップ変化に対する実際の電流の定常値（$t\to\infty$ の時の値）を求めよ。

(答)　$i(\infty)=1$　：$I^*(z)=\dfrac{z}{z-1}$ で$I(z)$に最終値の定理$i(\infty)=\lim_{z\to 1}(z-1)I(z)$を用いる。

4.3　制御器の演算時間を考慮した解析

これまでマイコンの演算は瞬時に終り，制御対象へは遅れることなく入力が加えられるものと考えた。しかし，実際には A/D 変換の処理時間やマイコンの処理時間がサンプリング周期に対して無視できない場合も考えられる。また，例えば電圧の制御にトランジスタをスイッチとして用いる場合には，時間遅れが発生することもある。このような問題について，*RL* 回路の電流をディジタル PI 制御する場合を例に取り考えてみよう。

図 4-5 の制御対象で，電流を検出して電流 PI 制御を行う。PI 制御は(3-16)の後退矩形近似

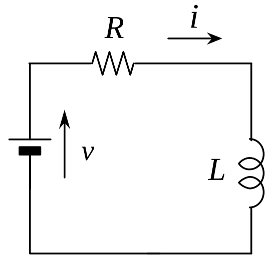


図 4-5　制御対象

を用い，次式で電圧指令を計算する。

$$v_r(k) = v_r(k-1) + K_P(e(k) - e(k-1)) + K_I Te(k)$$

ここで，偏差　$e(k) = i_r(k) - i(k)$，$i_r(k)$ ：電流指令

$t = (k-1)T$ の時点で電流 $i(k-1)$ を検出し電流 PI 制御により $v_r(k-1)$ を演算するが，演算に時間がかかり，$t = (k-1)T \sim kT$ の間には実際の電源電圧を変えることができず，サンプリング周期 T 遅れて $t = kT \sim (k+1)T$ の間 $v = v_r(k-1)$　とする。図 4-6 に時間経過を示す。

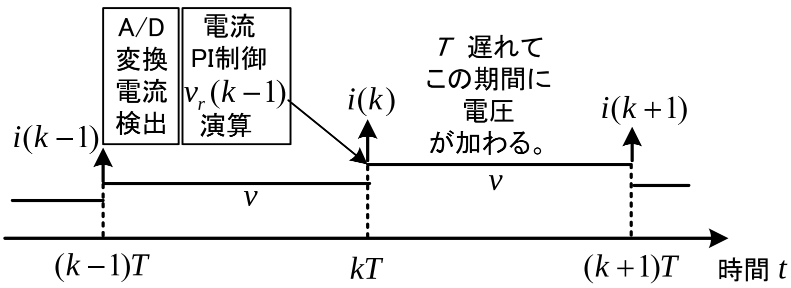


図 4-6 制御の時間経過

このときのシステム全体のブロック図を図 4-7 に示す。 (1-13) より，$Z\{v_r(k-1)\} = z^{-1}V_r(z)$ だから，**演算時間**(data processing time) T 遅れ部分のブロックは z^{-1} でよい。ディジタル PI 制御の伝達関数は (3-19) で求まっている。

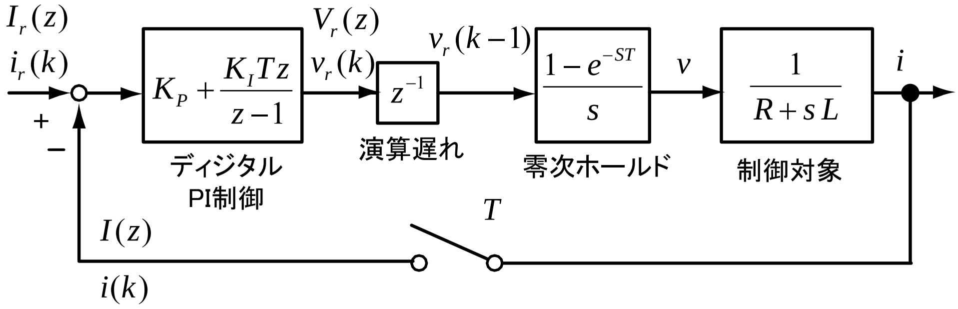


図 4-7　1 サンプリングの演算遅れを考慮したディジタル PI 制御系（文献(16)）

図 4-7 を z 変換したブロック線図を図 4-8 に示す。

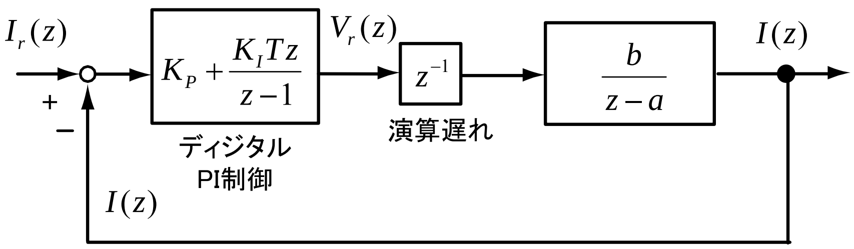


図 4-8　図 4-7 を z 変換したブロックだけに変形したブロック線図

電流のパルス伝達関数を求めると次式で与えられる。

$$\frac{I(z)}{I_r(z)} = \frac{\{(K_P + K_I T)z - K_P\}b}{z^3 - (1+a)z^2 + c\,z - K_P b} \qquad ①$$

但し, $a = \exp(-RT/L),\ b = (1-a)/R$

$$c = a + (K_P + K_I T)b\ ,\ T_I = K_P / K_I = L/R$$

これより，特性方程式は

$$z^3 + \alpha z^2 + \beta z + \gamma = 0 \qquad ②$$

但し，　$\alpha = -(1+a),\ \beta = c,\ \gamma = -K_P b$

双 1 次変換により

$$\left(\frac{1+s}{1-s}\right)^3 + \alpha\left(\frac{1+s}{1-s}\right)^2 + \beta\left(\frac{1+s}{1-s}\right) + \gamma = 0 \qquad ③$$

ラウスの方法により安定条件を求めると

$$a_0 = 1 - \alpha + \beta - \gamma > 0 \qquad ④$$

$$a_1 = 3 - \alpha - \beta + 3\gamma > 0 \qquad ⑤$$

$$a_2 = 3 + \alpha - \beta - 3\gamma > 0 \qquad ⑥$$

$$a_3 = 1 + \alpha + \beta + \gamma > 0 \qquad ⑦$$

$$a_1 a_2 - a_0 a_3 > 0 \qquad ⑧$$

④，⑦は常に成立する。安定条件は

⑤より　$4 > (4K_P + K_I T)b$　⑨

⑥より　$2K_P - K_I T + 2R > 0$　⑩

⑧より　$R + K_P a - K_I T - bK_P^2 > 0$　⑪

となる。ここで，次式を仮定する。

$$a \simeq 1 - RT/L,\ R \ll K_P,\ T \ll T_I$$

このとき安定条件は次のように簡単化される。

$$K_P < \frac{L}{T} \qquad ⑫$$

図 4-9 に，$R = 1.3\Omega, L = 0.01\mathrm{H}, T$=200$\mu$s とした場合，②式より求めた厳密な３つの極の軌跡を示す（共役の極は図示していない）。単位円の外に１つでも極があると制御系は不安定となる(8 の場合不安定)。⑫式によると，$K_p < 50$ が安定条件で，図 4-9 の厳密解析結果と比べると良い近似を与えることが判る。正の実軸上にある極は応答に振動を引き起こさないが，虚部が 0 でない極はその偏角に比例した周波数の振動を生じさせる。ただし，いずれの場合も極の絶対値が 1 よりある程度小さいなら減衰が速く応答に及ぼす影響は小さい。よって図の P_1 で 3,4,5 の極が応答に与える影響は小さい。図 4-2 や 6.5 節も参照せよ。

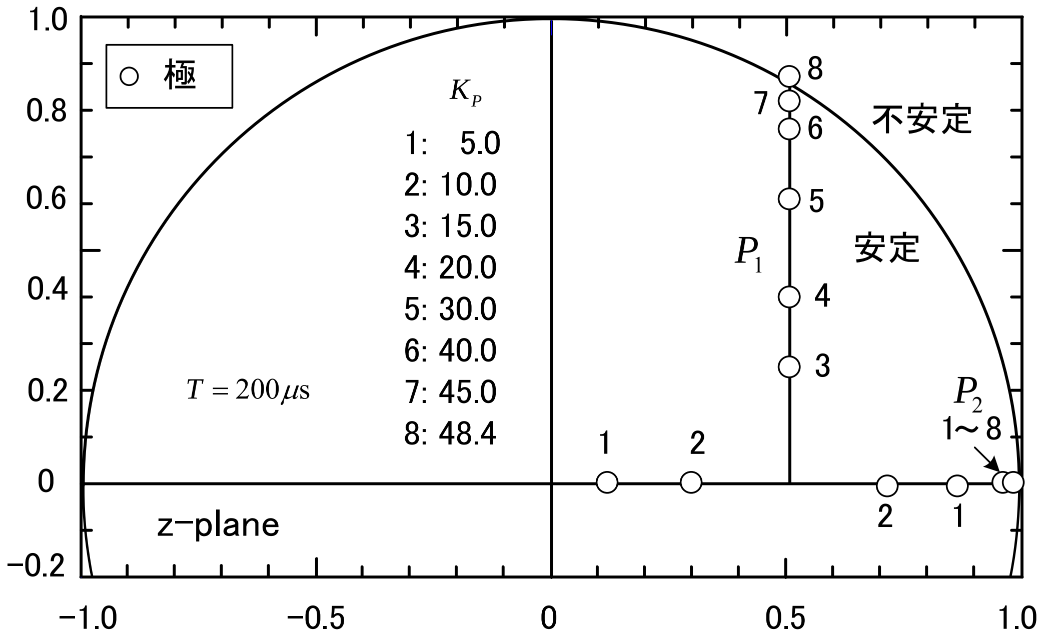


図 4-9　電流ディジタル制御系のゲインに対する極の軌跡（文献(16)）

Root trajectories of the digital system Fig.4-8 for the change of K_p.

第 5 章　周波数応答

連続系ではラプラス変換をして伝達関数を求め，$s = j\omega$ とおくことによりナイキスト線図やボード線図を描き，周波数応答を求めた。ディジタル制御系では，z 変換してパルス伝達関数を求めている。これと周波数応答はどのような関係にあるのだろうか？それを考えることにしよう。また，信号処理に良く用いられるフーリエ変換と z 変換の関係についても述べる。

5.1　周波数応答

サンプリングした複素正弦波（kT を時間 t に対応するものと考えよう）

$$r(k) = e^{j\omega kT} = \cos\omega kT + j\sin\omega kT \qquad (k = 0,1,2,\cdots) \tag{5-1}$$

を入力として考える。この z 変換は

$$R(z) = 1 + e^{j\omega T}z^{-1} + e^{j2\omega T}z^{-2} + e^{j3\omega T}z^{-3} + \cdots$$

だから

$$R(z) = \frac{z}{z - e^{j\omega T}} \tag{5-2}$$

である。パルス伝達関数を $G(z)$ とすると，出力の z 変換 $Y(z)$ は

$$Y(z) = G(z)R(z) = G(z)\frac{z}{z - e^{j\omega T}} \tag{5-3}$$

となる。これを，逆 z 変換すると出力 $y(k)$ が得られる。いま $G(z)$ 式が，相異なる m 個の極 $p_1, p_2, \cdots p_m$（複素根でもよい）をもつものとすれば，$Y(z)/z$ は(4-4)と同様に，次のように部分分数展開することができる。

$$\frac{Y(z)}{z} = \frac{q_1}{z - p_1} + \frac{q_2}{z - p_2} + \cdots + \frac{q_m}{z - p_m} + \frac{q_s}{z - e^{j\omega T}} \tag{5-4}$$

ここで，
$$q_i = \lim_{z \to p_i}(z - p_i)\frac{Y(z)}{z}$$

$$q_s = \lim_{z \to e^{j\omega T}}(z - e^{j\omega T})\frac{Y(z)}{z}$$

$$= G(e^{j\omega T}) \qquad \text{((5-3)より)} \tag{5-5}$$

$Y(z)$ を逆 z 変換して

$$y(k) = q_1(p_1)^k + q_2(p_2)^k + \cdots + q_m(p_m)^k + G(e^{j\omega T})(e^{j\omega T})^k \tag{5-6}$$

となる。安定な場合$|p_i|<1$であり，$k\to\infty$のとき最初のm項は全て0となり，最後の定常項のみが残る。定常項は

$$y_s(k)=G(e^{j\omega T})e^{jk\omega T}=\left|G(e^{j\omega T})\right|e^{j(k\omega T+\theta)} \tag{5-7}$$

ただし，$\theta=\arg(G(e^{j\omega T}))$

この結果，正弦波入力（Tはサンプリング周期だから，$\omega T\neq 2\pi$に注意）

$$r(k)=\sin k\omega T\quad (k=0,1,2,\cdots) \tag{5-8}$$

に対しては，複素入力の虚部に対応するから，(5-7)より

$$y_s(k)=\left|G(e^{j\omega T})\right|\sin(k\omega T+\theta) \tag{5-9}$$

となる。すなわち，パルス伝達関数のzを$e^{j\omega T}$とおいて，$G(e^{j\omega T})$の絶対値から振幅が，その偏角から位相が求められることが判った。これは，重要な結果である。$G(e^{j\omega T})$をディジタルシステムの**周波数応答**(frequency response)と呼ぶ。ωを変化させてグラフを書くとき，$G(e^{j\omega T})$の絶対値から**振幅特性**(amplitude characteristics)，その偏角から**位相特性**(phase characteristics)が求められる。後述の<u>信号を解析する</u>フーリエ変換の立場からは，振幅特性は**振幅スペクトル**(amplitude spectrum)，位相特性は**位相スペクトル**(phase spectrum)とよばれる。

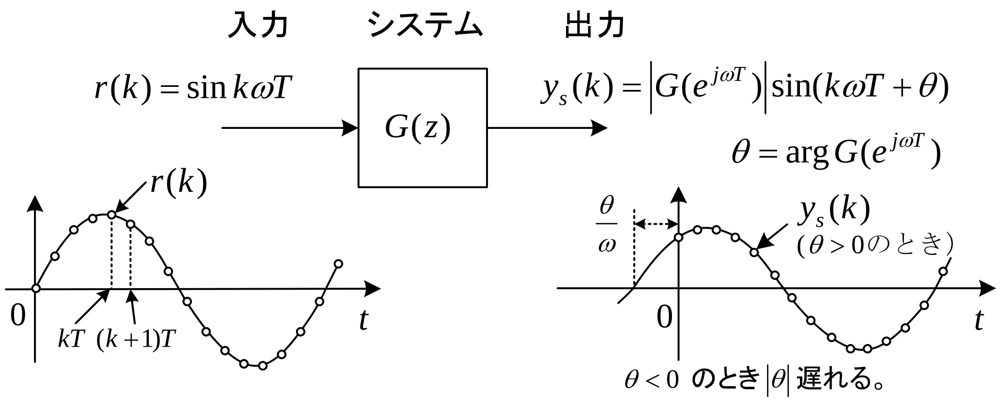


図 5-1 サンプリングした正弦波入力に対するシステムの出力(定常状態)

ところで，サンプリング周期Tに対して，**サンプリング角周波数**

$$\omega_s=\frac{2\pi}{T}\qquad（\textbf{サンプリング周波数}：\ f_s=\frac{1}{T}） \tag{5-10}$$

は重要な意味を持つ。サンプリングしたい正弦波の角周波数をω_0（周波数f_0，周期T_0）としたとき，ω_0が大きくて速く振動している場合には，サンプリング周期を短く（ω_sを大きく）しないと情報が失われる。次の条件を満たさないと，信号は再現できない。

$$T_0 \geq 2T \quad \text{あるいは,} \quad \omega_0 \leq \frac{\omega_s}{2} = \frac{\pi}{T} \quad \text{あるいは,} \quad f_0 \leq \frac{f_s}{2} = \frac{1}{2T} \tag{5-11}$$

これは，**サンプリング定理**(sampling theorem)と呼ばれる。サンプリング周期で決る$f_s/2$は**ナイキスト周波数**(Nyquist frequency)と呼ばれる。図5-2で，元の信号の少なくとも半周期に一つはサンプリングしないと周波数の情報が失われることがわかる。

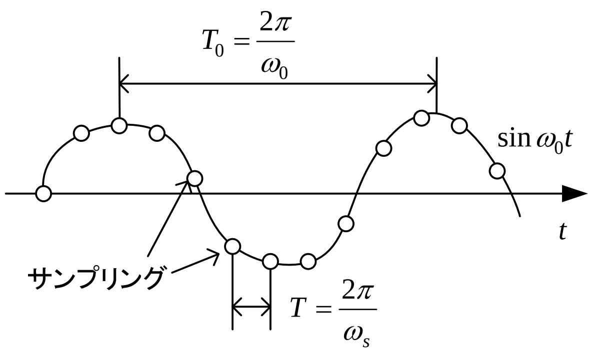


図5-2　サンプリング周期Tはどこまで大きく選べるか！

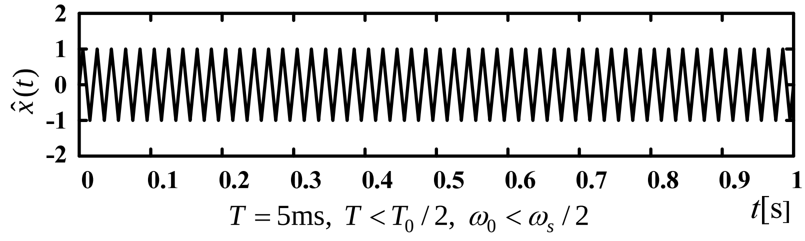
$x(t) = \sin\omega_0 t$で，$\omega_0 = 2\pi f_0$，$f_0 = 50\text{Hz}$, $T_0 = 1/f_0 = 20\text{ms}$のとき，種々のサンプリング周期$T$について$x(kT)$を計算し，それらの点を直線で結んだ（一種のD/A変換＋フィルタ）ときの波形を図5-3に示す（一種の再現した信号である）。

(a)の$T = 5\text{ms}$のとき，TがT_0の丁度$1/4$であり，最大と最小をサンプリングするので振幅が1の元の信号に近い波形が得られている。

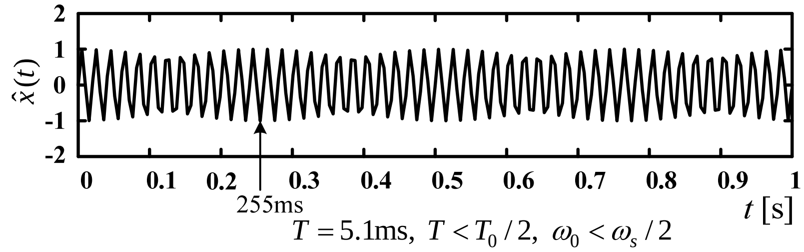
(b)の$T = 5.1\text{ms}$の場合には，1回サンプリングする度に(a)に比べると0.1msずれて51回目（$t = 0$を1回目とする）で時間は　$t = 50 \times 5.1 = 255\text{ms}$となるので，その時の$x(t)$ = -1を最小点としてサンプリングする。何故なら，255msから信号の周期20msの倍数を引くと，15ms余るので$15\text{ms}=3T_0/4$だから，$x(t)$ = -1となる。直線で結ぶのでなく補間をうまく行うと元の信号が再現できる。

Tが$T_0/2 = 10\text{ms}$より小さい場合には，元の信号の周波数は再現した信号でも変化しないが, (c)，(d), (e)の$T > 10\text{ms}$の場合には，再現した信号の周波数が変化して全く別の信号に化けている。これを**エイリアシング**(aliasing：偽信号)という。

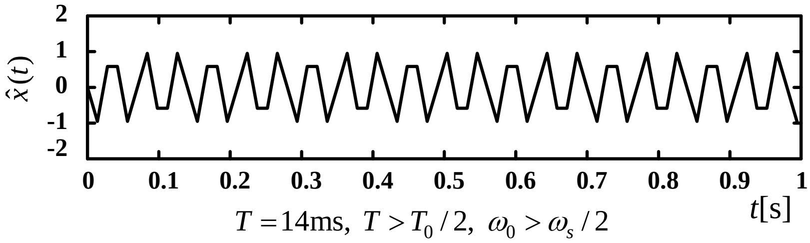
(e)の場合，ナイキスト周波数は$f_s/2 = 1/(2T) = 1000/38 = 26.3\text{Hz}$，この2倍の周波数は$52.6\text{Hz}$である。この周波数から元の信号の周波数を引くと$f_s - f_0 = 52.6 - 50 = 2.6\text{Hz}$となり，これが(e)の周波数である。(d)の周波数も同様に求めよ。$f_s = f_0$なら一定値（直流，0Hz）となる。 (e)の信号を観測したら本物と間違いそう。図5-9でも説明する。



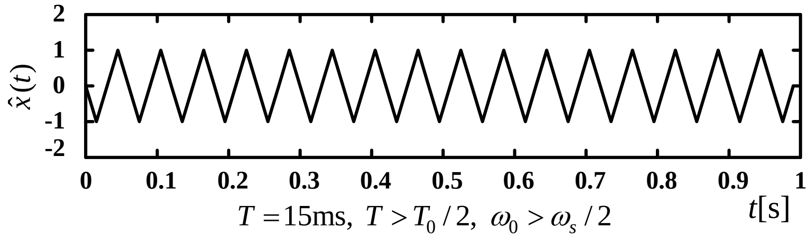
(a) 元の信号とほぼ一致



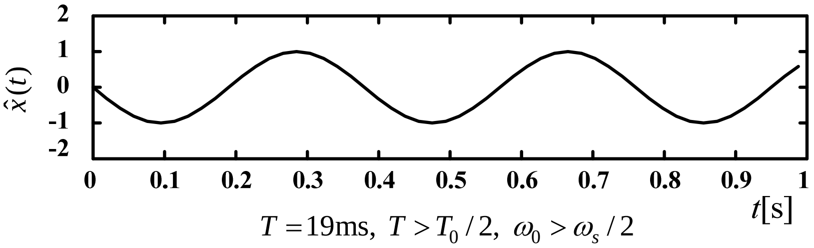
(b)　元の信号と周波数は一致



(c)　偽信号



(d)　偽信号



(e)　偽信号

図 5-3　周期 T_0=20ms の正弦波を種々のサンプリング周期 T でサンプルし直線で結んだ波形

[例題 5-1]　フィルタの出力 $y(k)$ が

$$y(k)=\frac{1}{2}(x(k)+x(k-1))$$

で与えられるとき，パルス伝達関数及び周波数応答(振幅特性及び周波数特性)を求めよ。ただし，サンプリングの角周波数　$\omega_s=2\pi/T$ を用い，$0\le\omega\le\omega_s$ の範囲で答えよ。

（解） z 変換してパルス伝達関数は次式で求まる。

$$G(z)=\frac{Y(z)}{X(z)}=\frac{1+z^{-1}}{2}$$

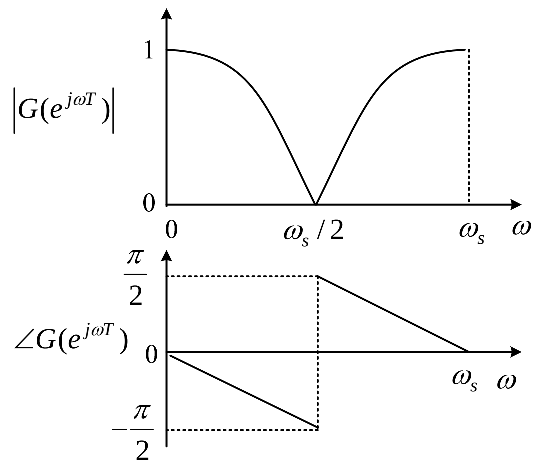
周波数応答は次式で求まる。

$$G(e^{j\omega T})=\frac{1+\cos\omega T-j\sin\omega T}{2}=\cos^2\frac{\omega T}{2}-j\sin\frac{\omega T}{2}\cos\frac{\omega T}{2}$$

$$=\cos\frac{\omega T}{2}(\cos\frac{\omega T}{2}-j\sin\frac{\omega T}{2})=\cos\frac{\omega T}{2}e^{-j\frac{\omega T}{2}}=\cos\frac{\omega\pi}{\omega_s}e^{-j\pi\frac{\omega}{\omega_s}}$$

振幅特性：　$\left|G(e^{j\omega T})\right|=\left|\cos\frac{\omega\pi}{\omega_s}\right|$

位相特性：$\angle G(e^{j\omega T})=\begin{cases}-\pi\dfrac{\omega}{\omega_s} & :\ 0\le\dfrac{\omega\pi}{\omega_s}<\dfrac{\pi}{2}\quad(0\le\omega<\dfrac{\omega_s}{2})\\ -\pi\dfrac{\omega}{\omega_s}+\pi & :\ \dfrac{\pi}{2}<\dfrac{\omega\pi}{\omega_s}<\pi\quad(\dfrac{\omega_s}{2}<\omega<\omega_s)\end{cases}$



＊　振幅特性で，0〜$\omega_s/2$ の範囲をまず考える。入力信号 $x(k)$ にはいろいろの周波数成分 ω が含まれる。この値 ω が 0〜$\omega_s/2$ の範囲なら，低周波成分は通過するが高周波数分（ノイズ成分）は振幅が小さくなってしゃ断される。つまり大切な低周波信号だけを取り出すローパスフィルタの特性である。位相は入力と出力間に遅れがない 0 が望ましいが，90 度まで ω に比例して遅れる。入力信号 $x(k)$ の周波数成分 ω が $\omega_s/2$ より大きい場合には，図の様に振幅が大きくなることがあるから，たとえば入力信号に含まれる $\omega_s/2$ より

大きい周波数のノイズの成分はこのフィルタで除去することはできない。従ってセンサのあとにアナログのローパスフィルタ（**アンチエイリアシングフィルタ**(anti-aliasing filter)という）を挿入してこれらを除去し，その出力を A/D 変換して検出することが行われる。

* $\omega=\omega_s$ の場合，正弦波の 1 周期に 1 回サンプリングするので，$x(k)$ は一定値となる。よって $y(k)$ は $x(k)$ に一致し，(5-9)より考えて，振幅特性の値は 1 となる。この場合(5-8)は初期位相を考える必要がある。

* $\cos\dfrac{\omega\pi}{\omega_s}>0$ のときは j の係数が位相となる。$\cos\dfrac{\omega\pi}{\omega_s}<0$ のとき，

$\cos\dfrac{\omega\pi}{\omega_s}=-\left|\cos\dfrac{\omega\pi}{\omega_s}\right|=\left|\cos\dfrac{\omega\pi}{\omega_s}\right|e^{j\pi}$ と考えてπを加える。－πしてもよい。

[例題 5-2]　パルス伝達関数が次式で与えられている。

$$G(z)=\frac{1-a}{z-a}$$

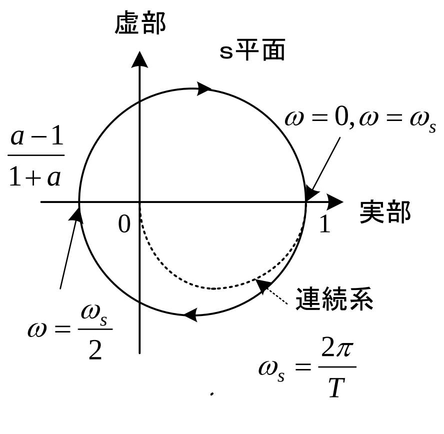
但し，$a<1$ である。この周波数応答をベクトル軌跡とボード線図に示せ。

（解）$G(e^{j\omega T})=\dfrac{1-a}{\cos\omega T-a+j\sin\omega T}\equiv x+jy$　とおいて，

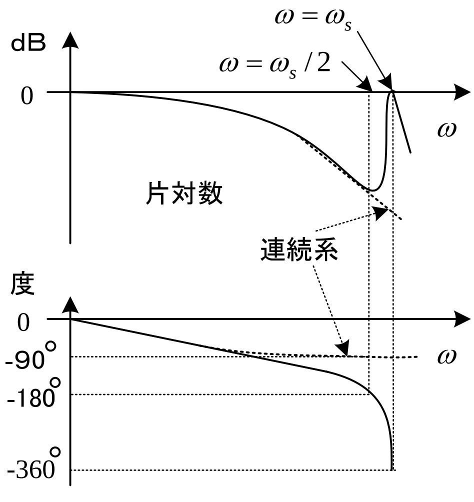
$$x=\frac{(1-a)(\cos\omega T-a)}{1-2a\cos\omega T+a^2}\quad ①\qquad y=-\frac{(1-a)\sin\omega T}{1-2a\cos\omega T+a^2}\quad ②$$

①,②より　$x^2+y^2=\dfrac{(1-a)^2}{1-2a\cos\omega T+a^2}$　であり，①より $\cos\omega T$ を x で表して代入すると，

$$x^2+y^2=\frac{1-a}{1+a}+\frac{2ax}{1+a}\quad\therefore\quad (x-\frac{a}{1+a})^2+y^2=\frac{1}{(1+a)^2}$$



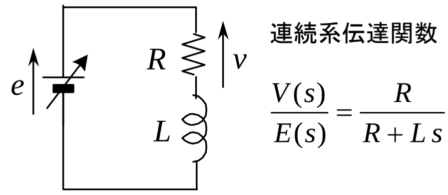
(a)　ベクトル軌跡



(b)　ボード線図

図 5-4 パルス伝達関数の周波数応答

図 5-4 に示した連続系は，対応する一次遅れ要素の伝達関数のボード線図である．例えば，



であり，対応するディジタル制御のパルス伝達関数は，

$$\frac{V(z)}{E(z)} = Z\left(\frac{1-e^{-sT}}{s}\frac{R}{R+Ls}\right) = \frac{1-a}{z-a} \qquad \text{但し，} \quad a = e^{-\frac{RT}{L}} < 1$$

である．このように，ディジタル制御では位相が遅れるので安定性に注意が必要である．

5.2　ナイキストの安定判別

連続系の場合と同じように周波数応答から安定判別が可能である。図 5-5 のディジタル制御系について考える。

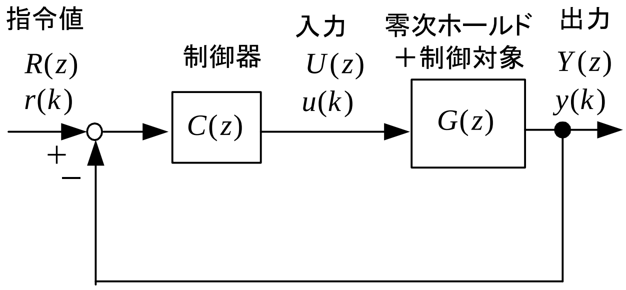


図 5-5 ディジタル制御系のブロック線図

一巡伝達関数$W_0(z)$は，分子の多項式$N(z)$と分母の多項式$D(z)$に分けて次式で表せる。

$$W_0(z) = C(z)G(z) \equiv \frac{N(z)}{D(z)} \tag{5-12}$$

図より，**閉ループ伝達関数**$W(z)$は次式で与えられる。

$$W(z) \equiv \frac{Y(z)}{R(z)} = \frac{C(z)G(z)}{1+C(z)G(z)} = \frac{N(z)}{D(z)+N(z)} \tag{5-13}$$

いま，

$$S(z) \equiv 1 + C(z)G(z) \tag{5-14}$$

とおくと，一般に$S(z)$は次式で表せる。

$$S(z)=\frac{D(s)+N(s)}{D(s)}=\frac{b_0z^n+b_1z^{n-1}+b_2z^{n-2}+\cdots+b_{n-1}z+b_n}{z^n+a_1z^{n-1}+\cdots+a_{n-1}z+a_n}$$

$$=b_0\frac{(z-z_1)(z-z_2)\cdots(z-z_n)}{(z-z_1')(z-z_2')\cdots(z-z_n')} \tag{5-15}$$

$S(z)$ の零点 z_i は閉ループ伝達関数の極であり，$S(z)$ の極 z_i' は一巡伝達関数の極と等しい。よって $S(z)$ の零点が単位円内にあれば図 5-5 は安定である。一方，一巡伝達関数の極は制御器と制御対象の極である。また(5-14)より極と零点の数は等しい。

(5-15)より $b_0>0$ と仮定すると

$$\angle S(z)=\sum_{i=1}^{n}\angle(z-z_i)-\sum_{i=1}^{n}\angle(z-z_i') \tag{5-16}$$

(5-16)をもとに，図 5-6 のように $z=e^{j\omega T}$ とおいて $\omega=0\rightarrow 2\pi/T$ と単位円上を 1 周させたときの $S(e^{j\omega T})$ の軌跡により安定判別法を考える。$S(z)$ の具体的な式が与えられないと軌跡は書けないが，零点や極が単位円の中か外かで，$S(e^{j\omega T})$ が原点を何回回るかが(5-16)より判る。図 5-6 で，$z=e^{j\omega T}$ を $\omega=0\rightarrow 2\pi/T$ と単位円上を 1 周させると，単位円の中にある極や零点について $z-z_i$ や $z-z_i'$ の偏角は 2π 変化し，単位円の外にある極や零点についてはある範囲に限られる。

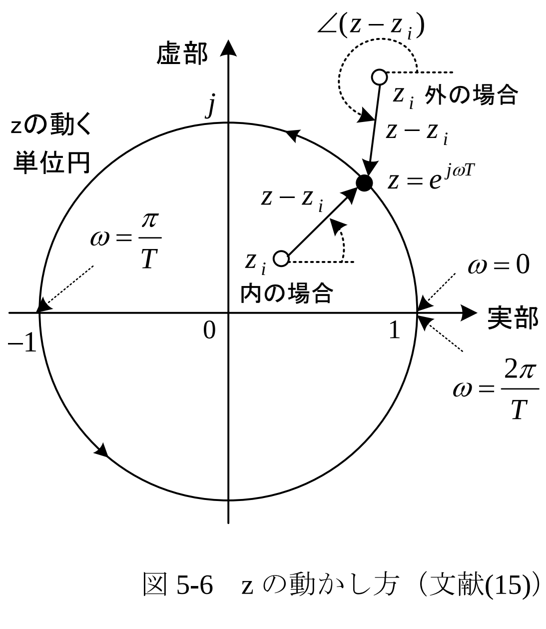


図 5-6　z の動かし方（文献(15)）

(5-16)より，$S(z)$ の偏角は極や零点について $z-z_i$ や $z-z_i'$ の偏角によって決まるので，例えば零点や極が全て単位円内なら打ち消しあって z が単位円を 1 周する間に $S(z)$ が原点を回る回数は 0 である。単位円の外にある極や零点についてはこの回転に関係ないと考え

られるので，結局

$S(z)$ が原点を反時計方向に回る回数

= 単位円の中にある零点 z_i の数－単位円の中にある極 z_i' の数　　　(5-17)

となる。安定の条件は，零点 z_i が全て単位円の中にあることだから，単位円の外にある極 z_i'（一巡伝達関数の不安定極）の数だけ $S(z)$ が原点を反時計方向に回る必要がある。

$$W_0(z) = S(z) - 1$$

だから，$W(e^{j\omega T})$ の軌跡は，$S(e^{j\omega T})$ の軌跡を-1 するだけなので以下の定理が得られる。

ナイキストの安定判別

定理 1．一巡伝達関数 $W_0(e^{j\omega T})$ の $\omega = 0 \to 2\pi/T$ における軌跡が，一巡伝達関数の不安定極の数だけ－1 の点を反時計回りに回れば図 5-5 のシステムは安定である。

定理 2．一巡伝達関数の不安定極の数が 0 の時，一巡伝達関数 $W_0(e^{j\omega T})$ の $\omega = 0 \to 2\pi/T$ における軌跡が－1 の点を回らなければ図 5-5 のシステムは安定である。

これは，連続系の場合と良く対応し，一巡伝達関数の周波数応答 $S(e^{j\omega T})$ から同じように安定性を知ることができる。ボード線図を描いて位相余裕やゲイン余裕も同様に考えてよい。ボード線図は片対数グラフ上に書く場合と等間隔で書く場合がある。

5.3　Z 変換とフーリエ変換の関係

連続周期信号の周波数解析には**フーリエ級数**(Fourier series)，連続非周期信号の周波数解析には**フーリエ変換**(FT：Fourier transform)が用いられる。フーリエ変換を連続周期信号の解析に用いると，フーリエ級数と同じスペクトルが得られる。離散時間非周期信号の周波数解析を行うフーリエ変換を**離散時間フーリエ変換**(DTFT : discrete-time Fourier transform)という。これは z 変換して周波数応答を求める場合の式と一致し，ディジタル制御と直接関係する。

コンピュータで数値的にフーリエ変換を計算する場合には，信号をサンプリングして離散時間信号として捉え，ある期間の信号を切り出してフーリエ変換する。これを**離散フーリエ変換**（DFT: discrete Fourier transform ）という。DFT の演算時間を短縮した**高速フーリエ変換**(FFT : fast Fourier transform)は信号処理に非常に良く利用されている。

表 5-1 公式の分類

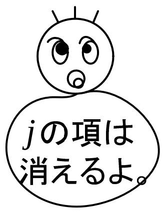
信号	連続周期	連続非周期	離散周期	離散非周期
フーリエ級数	（1）	適用不可	（3）	適用不可
フーリエ変換	（2）FT	（2）FT	（4）DTFT （5）DFT	（4） DTFT z変換と類似 （5） DFT 切り出し

（ ）は，以下で述べる節の番号である。(3)と(5)は実質的に同じ変換式

(1) 連続時間周期信号のフーリエ級数

周期T_0の連続時間周期信号 $f(t)$ (実数)のフーリエ級数は次式で定義される。

$$f(t) = \sum_{n=-\infty}^{\infty} c_n \, e^{j\,n\,\omega_o\,t}$$



$$= c_0 + \sum_{n=1}^{\infty} 2\left|c_n\right| \cos(n\ \omega_o\ t + \arg(c_n)) \qquad (5\text{-}18)$$

ただし， $f(t) = f(t+T_0)$, $\omega_o = 2\pi / T_0$ ：基本波の角周波数

このとき係数c_n（複素数）は次式で求められる（積分範囲は$-T_0/2 \sim T_0/2$ でもよい)。

$$c_n = \frac{1}{T_0}\int_0^{T_0} f(t) e^{-j\,n\,\omega_o\,t}\,d\,t \qquad (n = 0, \pm 1, \pm 2, \cdots), \qquad c_{-n} = \overline{c}_n \qquad (5\text{-}19)$$

$\overline{c}_n$ はc_nの共役複素数である。 (5-18)から(5-19)を導く。

$$\frac{1}{T_0}\int_0^{T_0} f(t) e^{-j\,n\,\omega_o\,t}\,d\,t = \frac{1}{T_0}\int_0^{T_0} \sum_{k=-\infty}^{\infty} c_k\, e^{j\,k\,\omega_o t} e^{-j\,n\,\omega_o\,t}\,d\,t$$ n と k は違う変化

$$= \sum_{k=-\infty}^{\infty} c_k \left(\frac{1}{T_0}\int_0^{T_0} e^{j(k-n)\,\omega_o\,t}\right) d\,t$$ $k = n$ 以外は 0

$$= c_n$$

(5-18)の第 2 式は n ＝ 1 なら ， $c_1 = \left|c_1\right| e^{j\theta}$, $c_{-1} = \left|c_1\right| e^{-j\theta}$ （(5-19)より）より得られる。

次に， $f(t)$ の２乗平均は

$$\frac{1}{T_0}\int_0^{T_0} f(t)^2 \, dt = \frac{1}{T_0}\int_0^{T_0} \left\{ \sum_{n=-\infty}^{\infty} c_n e^{jn\omega_o t} \right\}^2 dt$$

$$= \frac{1}{T_0}\int_0^{T_0} \left\{ \sum_{n=-\infty}^{\infty} c_n e^{jn\omega_o t} \right\} \left\{ \sum_{m=-\infty}^{\infty} c_m e^{jm\omega_o t} \right\} dt$$

$$= \sum_{n=-\infty}^{\infty} |c_n|^2 \qquad \text{三角関数の周期積分は 0} \qquad \text{(5-20)}$$

c_n を $f(t)$ のスペクトル，$|c_n|$ を振幅スペクトル，$\arg c_n$ を位相スペクトル，$|c_n|^2$ をパワースペクトルという。c_0 は直流分，$|c_1|$ は基本波分，$|c_2|$ が第 2 高調波分などを表す。ω_0 より低い交流の周波数成分がないのは，周期 T_0 より長い周期で繰り返す成分が元の波形にないからである。(5-18)から判るように，交流分の振幅は $2|c_n|$ になっていることに注意せよ。

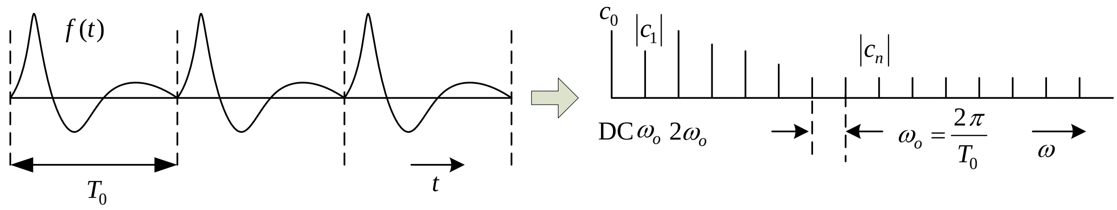


図 5-7 連続時間周期信号の振幅スペクトル

（2）連続時間信号のフーリエ変換

一般にフーリエ変換は次式で定義され，周期信号以外にも使用できる。

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} \, dt \qquad \text{(5-21)}$$

逆フーリエ変換は次式で与えられる。

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(\omega) e^{j\omega t} \, d\omega \qquad \text{(5-22)}$$

(5-18)を(5-21)を用いてフーリエ変換すると，周期信号について次式を得る。

$$F(\omega) = \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} c_n e^{jn\omega_o t} e^{-j\omega t} dt$$

$$= \sum_{n=-\infty}^{\infty} c_n \int_{-\infty}^{\infty} e^{j(n\omega_o - \omega)t} \, dt$$

$$= 2\pi \sum_{n=-\infty}^{\infty} c_n \ \delta(\omega - n\omega_0) \tag{5-23}$$

公式　$\delta(y) = \dfrac{1}{2\pi}\displaystyle\int_{-\infty}^{\infty} e^{jxy}\,dx$，$\delta(y) = \delta(-y)$

このように，周期関数に対するフーリエ変換は$\omega = n\omega_0$という離散的な点でしか値を持たず，フーリエ級数と同様の振幅スペクトルとなる。一般に非周期関数のフーリエ変換は，ωに対して連続な関数となる。

（3）離散時間周期信号のフーリエ級数

図 5-8 に示すように $f(k) \equiv f(kT)$ が周期 $KT(=T_0)$ をもつ離散時間周期信号で，$f(k) = f(k+K)$ が成りたつとき，(5-19)より，近似的にc_n を求めてみよう。

$$c_n = \frac{1}{T_0}\int_0^{T_0} f(t) e^{-jn\omega_0 t}\,dt \simeq \frac{1}{KT}\sum_{k=0}^{K-1} f(k)\,T\,e^{-jn\frac{2\pi}{KT}kT}$$

$$= \frac{1}{K}\sum_{k=0}^{K-1} f(k)\,e^{-j2\pi nk/K}$$

nが高調波次数，$\omega_o = 2\pi/(KT)$が基本波角周波数，$kT \Rightarrow t$に対応する。
このことから離散時間周期信号のフーリエ級数を次式で定義する（参考文献（12））。

$$c_n = \frac{1}{K}\sum_{k=0}^{K-1} f(k)\,e^{-j2\pi nk/K} \quad (n = 0,1,2,\cdots,K-1) \tag{5-24}$$

(5-24)より，$f(k)$ は次式で求められる（(5-25)の右辺に(5-24)を代入）。

$$f(k) = \sum_{n=0}^{K-1} c_n\,e^{j2\pi nk/K} \tag{5-25}$$

(5-24)より，(5-18)のc_nが数値的に計算できる。

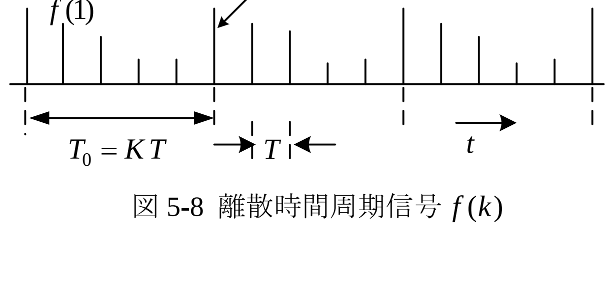


図 5-8　離散時間周期信号 $f(k)$

（4）離散時間非周期信号のフーリエ変換（**離散時間フーリエ変換:DTFT**）

連続的に変化する非周期信号 $f(t)$ に対して，サンプリング周期 T でサンプルした信号 $f^*(t)$ を次式で表す。

$$f^*(t) = \sum_{k=-\infty}^{\infty} f(k)\ \delta(t-k\ T)\ , \qquad \text{ただし，} f(k) = f(kT) \tag{5-26}$$

これは，(2-1)を $t<0$ まで拡張したものである。

$f^*(t)$ を(5-21)でフーリエ変換すると

$$F^*(\omega) = \int_{-\infty}^{\infty} \left\{ \sum_{k=-\infty}^{\infty} f(k)\delta(t-kT) \right\} e^{-j\ \omega t}\, d\,t = \sum_{k=-\infty}^{\infty} f(k) \int_{-\infty}^{\infty} \delta(t-kT) e^{-j\ \omega t}\, d\,t$$

$$= \sum_{k=-\infty}^{\infty} f(k)\ e^{-jk\ \omega\ T} \tag{5-27}$$

となる。$F^*(\omega)$ は**離散時間フーリエ変換**と呼ばれる。

逆変換は次式となる。

$$f(k) = \frac{1}{\omega_s} \int_0^{\omega_s} F^*(\omega) e^{jk\ \omega T}\, d\,\omega \tag{5-28}$$

ここで，　$\omega_s = 2\pi/T$　：サンプリング角周波数

(5-27)は周期性があり，次式が成り立つ。

$$F^*(\omega + n\omega_s) = F^*(\omega) \tag{5-29}$$

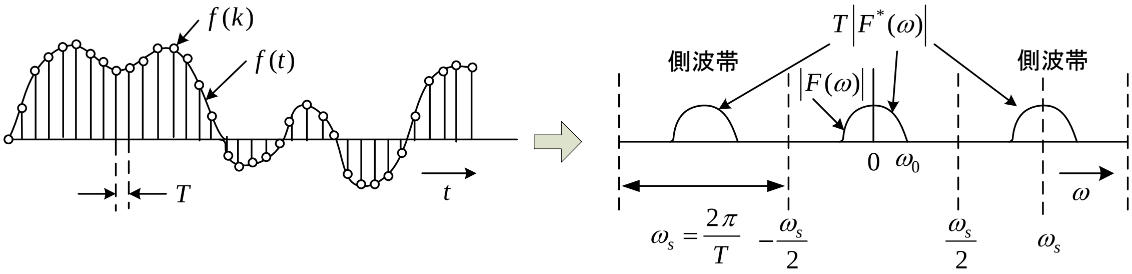
すなわち，　$\omega_s = 2\pi/T$ で波形が繰り返す。

(5-21)の連続信号 $f(t)$ のフーリエ変換 $F(\omega)$ と $F^*(\omega)$ には，(5-26)より次の関係が得られる（文献(5)）。

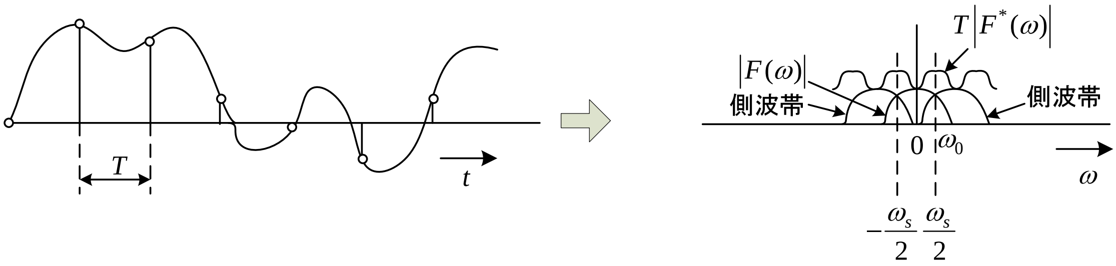
$$F^*(\omega) = \frac{1}{T}(F(\omega) + F(\omega-\omega_s) + F(\omega+\omega_s) + F(\omega-2\omega_s) + F(\omega+2\omega_s) + \cdots) \tag{5-30}$$

右辺第 2 項以降は**側波帯**(side band)と呼ばれる。側波帯はサンプリングによって連続信号の一部だけを取り出したことによる報いと思えばよい。離散時間フーリエ変換は，一般に図 5-9 のようにωに対して連続となる。サンプリング角周波数をもとの信号の最高周波数 ω_0 の 2 倍以上に取ると，図のように，$\left|F^*(\omega)\right|$ が孤立した分布になり，サンプリングした信号から元の信号が復元できる。しかし，$\omega_0 > \omega_s/2$ の場合は側波帯が低い周波数領域に入り込んでくる。この現象を**エイリアシング**(aliasing)という。このためサンプリングする前に連続信号の $\omega_s/2$ 以上の周波数成分を取り除く**アナログフィルタ**が用いられることは既に述べた。

(5-30)より，近似的にサンプラーは伝達関数 $1/T$ の比例要素と考えることができる。



(a) エイリアシングがない場合



(b) エイリアシングがある場合

図 5-9 離散時間非周期信号のフーリエ変換（**離散時間フーリエ変換**）

離散時間信号 $f(k)$ の z 変換は(1-8)により

$$F(z)=\sum_{k=0}^{\infty} f(k)\ z^{-k} \tag{5-31}$$

で定義された。一方， $f(k)$ のスペクトルを調べる場合には(5-27)より

$$F^*(\omega)=\sum_{k=0}^{\infty} f(k)\ e^{-jk\,\omega T} \tag{5-32}$$

のフーリエ変換を使えばよい。ただし，$k<0$ で信号が 0 とする。(5-31)と(5-32)を比べると，$z=e^{\,j\omega T}$ とすれば両者は一致する。

伝達関数 $G(z)$ で $z=e^{\,j\omega T}$ とおいた周波数応答 $G(e^{j\omega T})$ は， $k=0$ での単位インパルス（ $r(0)=1$ ）に対する出力信号の離散時間フーリエ変換である。なぜなら，$\{r(k)\}=\{1,0,0,0,\cdots\}$ のとき，その z 変換は $R(z)=Z\{r(k)\}=1$ なので，$Y(z)=G(z)R(z)\ =G(z)$ だから，出力の z 変換と伝達関数が等しいからである。

[例題 5-3] $f(t)=\sin\omega_0 t=\sin 100\pi t$ を周期 $T=19$ms でサンプリングするとき，偽信号の周波数が 2.6Hz となることを説明せよ。（図 5-3(e)の場合）

（解）　$f(t)=\sin\omega_0 t \quad (-\infty<t<\infty)$ のフーリエ変換 $F(\omega)$ は次式で与えられる。

$$F(\omega)=-j\pi\delta(\omega-\omega_0)+j\pi\delta(\omega+\omega_0)$$

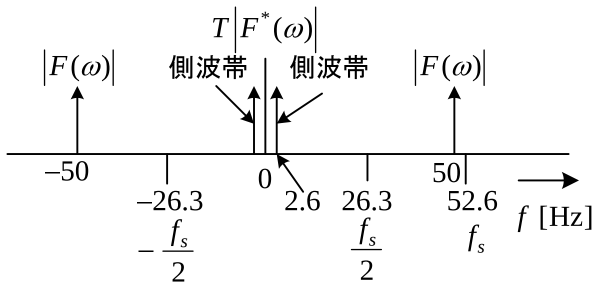
従って，$|F(\omega)|$ は，$\omega_0=2\pi f_0$，すなわち $f_0=\pm 50\text{Hz}$ にスペクトルをもつ。これはナイキスト周波数 $f_s/2=0.5/0.019=26.3\text{Hz}$ の外側である。

サンプルした信号の離散時間フーリエ変換は(5-30)より次式となる。

$$F^*(\omega)=\frac{1}{T}(F(\omega)+F(\omega-\omega_S)+F(\omega+\omega_S)+F(\omega-2\omega_S)+F(\omega+2\omega_S)+\cdots)$$

$$=\frac{1}{T}(-j\pi\delta(\omega-\omega_0)+j\pi\delta(\omega+\omega_0)-j\pi\delta(\omega-\omega_s-\omega_0)+j\pi\delta(\omega-\omega_s+\omega_0)$$
$$-j\pi\delta(\omega+\omega_S-\omega_0)+j\pi\delta(\omega+\omega_s+\omega_0)+\cdots)$$

右辺第 4 項より，側波帯が $\omega=\omega_s-\omega_0$ の低周波領域でスペクトルをもつ。周波数に直すと，$f_s-f_0=52.6-50=2.6\text{Hz}$ となる。すなわち，エイリアシングが生じて，低周波領域にスペクトルが現れるのである。右側の $f_s=52.6\text{Hz}$ を中心とした側波帯の負成分が 2.6Hz のところに顔を出す。 $f_s/2=26.3\text{Hz}$ で 50Hz のスペクトルが折り返すと考えても良い。



(5) 離散フーリエ変換（DFT）

(5-27)の $F^*(\omega)$ は，ω の連続な関数であり，数値計算に不適である。また，無限数列 $f(k)$ について計算することも現実的でない。そこで，サンプリング周期を T として K 個の連続サンプルを切り出し，これに対するフーリエ変換を行う。kT が時間に対応し，

$$\omega_0=\frac{2\pi}{T_0}=\frac{2\pi}{KT},\quad \omega=n\,\omega_0=\frac{2\pi n}{KT}$$

と考えると，(5-27)より次式が得られる。

$$F(n)=\sum_{k=0}^{K-1} f(k)\,e^{-j2\pi nk\ /K} \qquad (n=0,1,2,\cdots,K-1) \qquad \text{(5-33)}$$

(5-33)が**離散フーリエ変換**の定義式である。

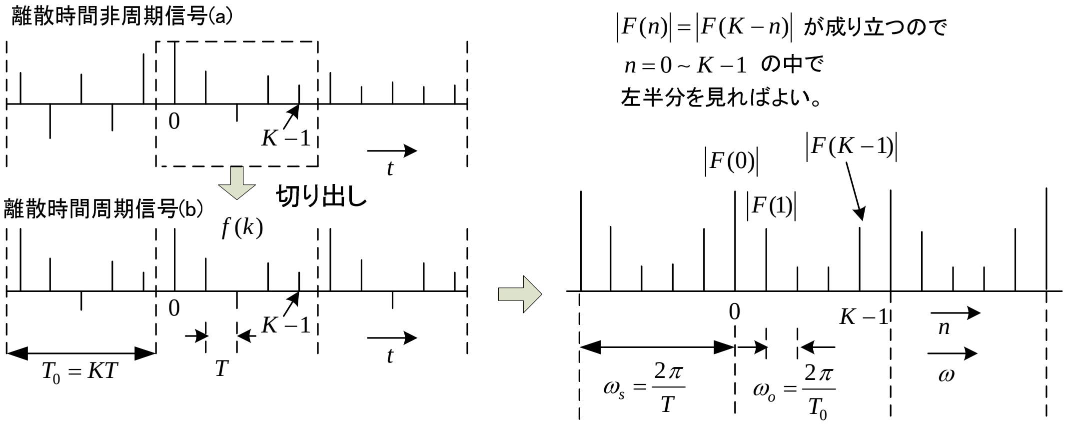
図 5-10 に信号の切り出しと離散フーリエ変換の結果を説明している。離散フーリエ変換

は，連続する K 個のデータを検出し，それらを周期 T_0 として $-\infty \sim +\infty$ に広げた離散時間周期信号のフーリエ級数の係数を求めている。離散フーリエ変換は，ω に対して連続ではなくサンプル数と同じ周波数に分割した成分が求まる。$F(1)$ は最もゆっくりした周波数 $1/T_0$ の成分であるが，フーリエ級数の基本波周波数とは限らない。しかし，もとの信号が周期信号で，その 1 周期分を切り出した場合については，基本波の周期は T_0 である。(5-33) の $F(n)$ は，(5-24) の離散時間周期信号のフーリエ級数の係数 c_n と，$F(n) = K\,c_n \quad \therefore \left|F(n)\right| = K\left|c_n\right|$ の関係になっていることが判る。この場合 $n=1$ のときが基本波で，$F(n)$ は第 n 調波を表わす。(5-18)より，(5-33)の離散フーリエ変換の絶対値をサンプル数 K で割って 2 倍すると各調波の振幅となる（参考文献(14)）。

逆離散フーリエ変換（IDFT: inverse discrete Fourier transform）は次式で求められる。

$$f(k) = \frac{1}{K}\sum_{n=0}^{K-1} F(n)\, e^{j2\pi nk\ /K} \qquad (k = 0, 1, 2, \cdots, K-1) \tag{5-34}$$

離散フーリエ変換の計算量は多いので，計算量を大幅に減少させた**高速フーリエ変換**(fast Fourier transform, FFT)が非常に良く利用されている。



T ：サンプリング周期，T_0 ：切り出した期間，K ：切り出したデータ数

図 5-10 信号の切り出しと離散フーリエ変換

第６章　時間領域での解析

ディジタル制御系の解析に z 変換を用いる方法をこれまで述べてきた。本章では，時間領域で直接解析する方法を述べる。このために行列を用いるが，変数が多い場合でも同じ式で理論が組み立てられ便利である。差分方程式で表わされるから応答の数値計算が容易で，制御器の式は実際にマイコンで使う式がそのまま利用できる。やや複雑とはなるがパルス伝達関数も行列を使って求められるので周波数応答も計算できる。

6.1　連立差分方程式

まず，かんたんな差分方程式から出発しよう。

$$x_1(k+1) = x_1(k) + 2x_2(k)$$
$$x_2(k+1) = 2x_1(k) + x_2(k) \tag{6-1}$$

但し，$k = 0,1,2,\cdots$ で，$x_1(0) = 1, x_2(0) = 0$ とする。

(6-1)を行列で表現すると，次式となる。

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} \tag{6-2}$$

これを，次式の記号で書く。それぞれ順番に対応する。

$$\boldsymbol{x}(k+1) = \boldsymbol{A}\,\boldsymbol{x}(k) \tag{6-3}$$

一般に，

$$\boldsymbol{x}(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_n(k) \end{bmatrix} \quad n\text{ 次元ベクトル，} \qquad \boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & \cdot & a_{1n} \\ a_{21} & a_{22} & & \cdot \\ \vdots & & \ddots & \vdots \\ a_{n1} & \cdot & \cdot & a_{nn} \end{bmatrix} \quad n \times n\text{ 行列}$$

a_{ij} ：i 行 j 列の成分

一般項を初期値で表そう。(6-3)より

$$\boldsymbol{x}(k) = \boldsymbol{A}\boldsymbol{x}(k-1) = \boldsymbol{A}^2\boldsymbol{x}(k-2) = \cdots = \boldsymbol{A}^k\boldsymbol{x}(0) \tag{6-4}$$

従って，$\boldsymbol{x}(k)$ を求めるには，$\boldsymbol{A}^k$ を求める必要がある。これには，**行列の対角化**が利用できる。

$\boldsymbol{A}$を n 行 n 列の行列（$n\times n$行列と書く）として述べよう。$\boldsymbol{A}$の**固有値**$\lambda_1,\lambda_2,\cdots,\lambda_n$（一般に複素数）が全て異なるとき，$\boldsymbol{A}$は次式で表される。

$$\boldsymbol{A}=\boldsymbol{P}\boldsymbol{Q}\boldsymbol{P}^{-1} \tag{6-5}$$

ここで，
$$\boldsymbol{Q}=\begin{bmatrix}\lambda_1 & & & \mathbf{0}\\ & \lambda_2 & & \\ & & \ddots & \\ \mathbf{0} & & & \lambda_n\end{bmatrix},\ \boldsymbol{P}=\begin{bmatrix}\boldsymbol{u}_1 & \boldsymbol{u}_2 & \cdots & \boldsymbol{u}_n\end{bmatrix} \tag{6-6}$$

$\boldsymbol{P},\boldsymbol{Q}$はいずれも，$n\times n$行列である。固有値$\lambda_i$に対する一つの**固有ベクトル**を$\boldsymbol{u}_i=\begin{bmatrix}u_i^1\\ \vdots\\ u_i^n\end{bmatrix}$

としている。(6-5)は，固有値の定義

$$\boldsymbol{A}\boldsymbol{u}_i=\lambda_i\boldsymbol{u}_i \qquad (i=1,2,\cdots,n) \tag{6-7}$$

より以下の様に導ける。行列と成分の掛け算はサイズが合えば可能である。

$$\boldsymbol{A}\boldsymbol{P}=\boldsymbol{A}[\boldsymbol{u}_1\ \boldsymbol{u}_2\cdots\boldsymbol{u}_n]=[\boldsymbol{A}\boldsymbol{u}_1\ \boldsymbol{A}\boldsymbol{u}_2\cdots\boldsymbol{A}\boldsymbol{u}_n]=[\lambda_1\boldsymbol{u}_1\ \lambda_2\boldsymbol{u}_2\cdots\lambda_n\boldsymbol{u}_n]=\boldsymbol{P}\boldsymbol{Q}$$

$$\because\quad \boldsymbol{PQ}=\begin{bmatrix}u_1^1 & u_2^1 & u_3^1\\ u_1^2 & u_2^2 & u_3^2\\ u_1^3 & u_2^3 & u_3^3\end{bmatrix}\begin{bmatrix}\lambda_1 & 0 & 0\\ 0 & \lambda_2 & 0\\ 0 & 0 & \lambda_3\end{bmatrix}=\begin{bmatrix}\lambda_1u_1^1 & \lambda_2u_2^1 & \lambda_3u_3^1\\ \lambda_1u_1^2 & \lambda_2u_2^2 & \lambda_3u_3^2\\ \lambda_1u_1^3 & \lambda_2u_2^3 & \lambda_3u_3^3\end{bmatrix}=\begin{bmatrix}\lambda_1\begin{bmatrix}u_1^1\\ u_1^2\\ u_1^3\end{bmatrix} & \lambda_2\begin{bmatrix}u_2^1\\ u_2^2\\ u_2^3\end{bmatrix} & \lambda_3\begin{bmatrix}u_3^1\\ u_3^2\\ u_3^3\end{bmatrix}\end{bmatrix}$$

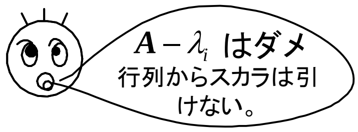
(6-5)を用いると，次式により$\boldsymbol{A}^k$が計算できる。対角化のすばらしい点である。

$$\boldsymbol{A}^k=\boldsymbol{P}\boldsymbol{Q}\boldsymbol{P}^{-1}\boldsymbol{P}\boldsymbol{Q}\boldsymbol{P}^{-1}\boldsymbol{P}\boldsymbol{Q}\boldsymbol{P}^{-1}\cdots\boldsymbol{P}\boldsymbol{Q}\boldsymbol{P}^{-1}$$

$$=\boldsymbol{P}\boldsymbol{Q}^k\boldsymbol{P}^{-1}$$

$$=\boldsymbol{P}\begin{bmatrix}\lambda_1^k & & & \mathbf{0}\\ & \lambda_2^k & & \\ & & \ddots & \\ \mathbf{0} & & & \lambda_n^k\end{bmatrix}\boldsymbol{P}^{-1} \tag{6-8}$$

(6-7)より，



$$(\boldsymbol{A}-\lambda_i \boldsymbol{I})\boldsymbol{u}_i = \boldsymbol{0}$$

$$\boldsymbol{I} = \begin{bmatrix} 1 & & \boldsymbol{0} \\ & \ddots & \\ \boldsymbol{0} & & 1 \end{bmatrix}$$ 単位行列 $n \times n$

$\boldsymbol{A}-\lambda_i \boldsymbol{I}$ の逆行列が存在すると，$\boldsymbol{u}_i = \boldsymbol{0}$ となり，つまらない。$\boldsymbol{A}$ の固有値は，この逆行列が存在しない条件，

$$|\boldsymbol{A}-\lambda \boldsymbol{I}| = 0 \qquad \text{または，} \qquad |\lambda \boldsymbol{I}-\boldsymbol{A}| = 0$$

より計算できる。$[\ \ ]$ 行列，$|\ \ |$ 行列式をしっかり区別すること。行列式はスカラである。

話を，(6-2)へ戻そう。固有値，固有ベクトルを求め対角化してみる。

$$|\boldsymbol{A}-\lambda \boldsymbol{I}| = \begin{vmatrix} 1-\lambda & 2 \\ 2 & 1-\lambda \end{vmatrix} = (\lambda-3)(\lambda+1) = 0$$

$\therefore \lambda = 3, -1$ （固有値）

(1) $\lambda_1 = 3$ のとき，

$$\begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = 3\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$\therefore u_1 = u_2$

この式を満足する固有ベクトルは無数にある。全成分が 0 以外ならどれでも良いが，簡単な次の固有ベクトルを選ぶ。

$$\boldsymbol{u}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

(2) $\lambda_2 = -1$ のとき，

$$\begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = -\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$\therefore u_1 = -u_2$

同様に，次の固有ベクトルを選ぶ。

$$\boldsymbol{u}_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

よって，

$$\boldsymbol{P} = [\boldsymbol{u}_1 \quad \boldsymbol{u}_2] = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad \boldsymbol{P}^{-1} = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad \boldsymbol{A} = \boldsymbol{PQP}^{-1} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 3 & 0 \\ 0 & -1 \end{bmatrix}(\frac{1}{2})\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

すると，

$$A^k = PQ^kP^{-1} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 3 & 0 \\ 0 & -1 \end{bmatrix}^k (\frac{1}{2}) \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$= (\frac{1}{2}) \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 3^k & 0 \\ 0 & (-1)^k \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$= (\frac{1}{2}) \begin{bmatrix} 3^k + (-1)^k & 3^k - (-1)^k \\ 3^k - (-1)^k & 3^k + (-1)^k \end{bmatrix}$$

よって，

$$x(k) = A^k x(0)$$

$$= (\frac{1}{2}) \begin{bmatrix} 3^k + (-1)^k & 3^k - (-1)^k \\ 3^k - (-1)^k & 3^k + (-1)^k \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$= (\frac{1}{2}) \begin{bmatrix} 3^k + (-1)^k \\ 3^k - (-1)^k \end{bmatrix}$$

[問題 6-1]　$x_1(0) = 1,\ x_2(0) = 0$ のとき，

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 2 & -2 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}$$

の解 $x_1(k),\ x_2(k)$ を行列の対角化を利用して求めよ。また，k が∞のときどうなるか。

(答)　$A = \begin{bmatrix} 2 & -2 \\ 1 & 0 \end{bmatrix}$　とおくと，

固有値は，$\lambda_1 = 1 + j = \sqrt{2}(\cos\frac{\pi}{4} + j\sin\frac{\pi}{4}) = \sqrt{2}\,e^{j\pi/4}$，$\lambda_2 = 1 - j = \sqrt{2}\ e^{-j\pi/4}$

$$A = \begin{bmatrix} 1+j & 1-j \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1+j & 0 \\ 0 & 1-j \end{bmatrix} (\frac{1}{2j}) \begin{bmatrix} 1 & -(1-j) \\ -1 & 1+j \end{bmatrix}$$

$$A^k = \begin{bmatrix} (\sqrt{2})^{k+1}\sin\frac{(k+1)\pi}{4} & (\sqrt{2})^{k+2}\cos\frac{(k+2)\pi}{4} \\ (\sqrt{2})^k \sin\frac{k\pi}{4} & (\sqrt{2})^{k+1}\cos\frac{(k+1)\pi}{4} \end{bmatrix}$$

$$\begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} = \begin{bmatrix} (\sqrt{2})^{k+1}\sin\frac{(k+1)\pi}{4} \\ (\sqrt{2})^k \sin\frac{k\pi}{4} \end{bmatrix}$$

k が∞のとき，発散する。これには，$|\lambda_1| = |\lambda_2| = \sqrt{2}$ が関係していることが判る。

6.2 ディジタル制御系の安定判別

図 6-1 のディジタル制御系（制御器，零次ホールド，制御対象）は，(1-31)に示したように一般に差分方程式を用いて表すことができる。すなわち次式で記述できる。

状態方程式

$$\boldsymbol{x}(k+1) = \boldsymbol{A}\,\boldsymbol{x}(k) + \boldsymbol{b}\,r(k) \qquad (k = 0,1,2,\cdots) \tag{6-9}$$

出力方程式

$$y(k) = \boldsymbol{c}\,\boldsymbol{x}(k) \tag{6-10}$$

ここで，$\boldsymbol{x}$ ：状態変数（$n \times 1$のベクトル），r：指令値，y：出力

$\boldsymbol{A}$ ：系行列（$n \times n$の行列），$\boldsymbol{b}$ ：$n \times 1$のベクトル，$\boldsymbol{c}$ ：$1 \times n$のベクトル

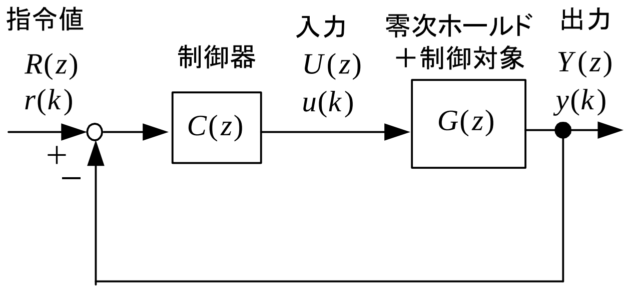


図 6-1　ディジタル制御系

(6-9)が本質的な式で，(6-10)の出力方程式は $\boldsymbol{x}$ の中のどれをセンサで検出するのかを明確にする（$\boldsymbol{c}$ のどれかの要素が 1 になることが多い）。$\boldsymbol{x}$ は，制御対象と制御器の変数で，それらを式で表すのに最低限必要となるものに限られる。ただし，$\boldsymbol{x}$ の選び方は何通りも考えられ，独立な変数であれば本質的な違いはない。例えば，$\boldsymbol{x}$ の中の変数としては，コイルの電流，コンデンサの電圧，磁束，回転速度，積分器（I 制御）の出力などである。

(6-9)で，$r(k) = r$ (一定）とし，$\boldsymbol{x}(1), \boldsymbol{x}(2), \boldsymbol{x}(3)$ と順々に計算して，$\boldsymbol{x}(0)$ のみで表すと次式を得る。

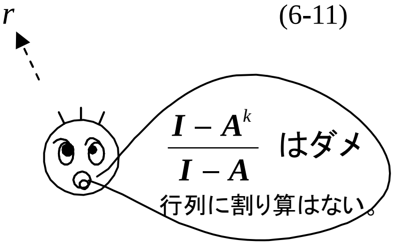
$$k = 0 \qquad \boldsymbol{x}(1) = \boldsymbol{A}\,\boldsymbol{x}(0) + \boldsymbol{b}\,r$$

$$k = 1 \qquad \boldsymbol{x}(2) = \boldsymbol{A}\,\boldsymbol{x}(1) + \boldsymbol{b}\,r = \boldsymbol{A}^2\,\boldsymbol{x}(0) + (\boldsymbol{A} + \boldsymbol{I})\boldsymbol{b}\,r$$

$$k = 2 \qquad \boldsymbol{x}(3) = \boldsymbol{A}\,\boldsymbol{x}(2) + \boldsymbol{b}\,r = \boldsymbol{A}^3\,\boldsymbol{x}(0) + (\boldsymbol{A}^2 + \boldsymbol{A} + \boldsymbol{I})\boldsymbol{b}\,r$$

・・・

$$\boldsymbol{x}(k)=\boldsymbol{A}^k\boldsymbol{x}(0)+(\boldsymbol{A}^{k-1}+\boldsymbol{A}^{k-2}+\cdots+\boldsymbol{A}+\boldsymbol{I})\boldsymbol{b}\,r \tag{6-11}$$



ただし，$\boldsymbol{I}: n\times n$ 単位行列
行列の対角化を利用した(6-8)を用いると，

$$\boldsymbol{x}(k)=\boldsymbol{P}\boldsymbol{Q}^k\boldsymbol{P}^{-1}\boldsymbol{x}(0)+\boldsymbol{P}(\boldsymbol{Q}^{k-1}+\boldsymbol{Q}^{k-2}+\cdots+\boldsymbol{Q}+\boldsymbol{I})\boldsymbol{P}^{-1}\boldsymbol{b}\,r \quad \text{より}$$

$$\boldsymbol{x}(k)=\boldsymbol{P}\begin{bmatrix}\lambda_1^k & & & \boldsymbol{0}\\ & \lambda_2^k & & \\ & & \ddots & \\ \boldsymbol{0} & & & \lambda_n^k\end{bmatrix}\boldsymbol{P}^{-1}\boldsymbol{x}(0)+\boldsymbol{P}\begin{bmatrix}\dfrac{1-\lambda_1^k}{1-\lambda_1} & & & \boldsymbol{0}\\ & \dfrac{1-\lambda_2^k}{1-\lambda_2} & & \\ & & \ddots & \\ \boldsymbol{0} & & & \dfrac{1-\lambda_n^k}{1-\lambda_n}\end{bmatrix}\boldsymbol{P}^{-1}\boldsymbol{b}\,r \tag{6-12}$$

$$\because\ \boldsymbol{Q}^k=\begin{bmatrix}\lambda_1 & 0 & 0\\ 0 & \lambda_2 & 0\\ 0 & 0 & \lambda_3\end{bmatrix}^k=\begin{bmatrix}\lambda_1^{\,k} & 0 & 0\\ 0 & \lambda_2^{\,k} & 0\\ 0 & 0 & \lambda_3^{\,k}\end{bmatrix}$$

$$\boldsymbol{I}+\boldsymbol{Q}+\boldsymbol{Q}^2+\cdots+\boldsymbol{Q}^{k-1}=\begin{bmatrix}1+\lambda_1+\lambda_1^{\,2}+\cdots+\lambda_1^{\,k-1} & 0 & 0\\ 0 & 1+\lambda_2+\lambda_2^{\,2}+\cdots+\lambda_2^{\,k-1} & 0\\ 0 & 0 & 1+\lambda_3+\lambda_3^{\,2}+\cdots+\lambda_3^{\,k-1}\end{bmatrix}$$

$k\to\infty$ とした定常状態を考えると，$\boldsymbol{x}(k)$ がある値に収束する必要十分条件は，$|\lambda_i|<1\ (i=1,2,\cdots,n)$ である。これから以下の定理が得られる。

$\boldsymbol{x}(k+1)=\boldsymbol{A}\boldsymbol{x}(k)+\boldsymbol{b}r(k)$ で表される**ディジタル制御系が安定である必要十分条件**は，系行列 $\boldsymbol{A}$ の**固有値**を $\lambda_1,\lambda_2,\cdots,\lambda_n$ とすると，

$$|\lambda_i|<1 \qquad (i=1,2,\cdots,n) \tag{6-13}$$

である。なお，この安定条件は固有値が重根の場合にも成立する。

極めて重要な定理であるから，しっかり覚えておこう。上記の証明では，指令値 $r(k)$ は一

定と仮定したが，$\boldsymbol{x}(k)$ に無関係であれば一定である必要はない。$r(k)$ が $\boldsymbol{x}(k)$ に関係する場合は，$r(k)$ を消去した状態方程式を新たに作って安定判別しないといけない。

なお，(6-13)は，第 4 章で z 変換を用いて得られた(4-7)と同じものである。この関連を以下に示す。ベクトルの z 変換を各変数の数列の z 変換と定義(≡) し，以下の記号を用いる。

$$Z\{\boldsymbol{x}(k)\} \equiv \begin{bmatrix} Z\{x_1(k)\} \\ Z\{x_2(k)\} \\ \vdots \\ Z\{x_n(k)\} \end{bmatrix} = \begin{bmatrix} X_1(z) \\ X_2(z) \\ \vdots \\ X_n(z) \end{bmatrix} = \boldsymbol{X}(z) \tag{6-14}$$

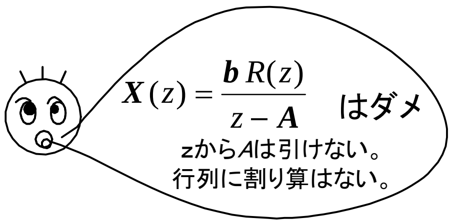
(6-9),(6-10)を z 変換して次式を得る。

$$z\,\boldsymbol{X}(z) - z\,\boldsymbol{x}(0) = \boldsymbol{A}\,\boldsymbol{X}(z) + \boldsymbol{b}\,R(z) \tag{6-15}$$

$$Y(z) = \boldsymbol{c}\,\boldsymbol{X}(z) \tag{6-16}$$

パルス伝達関数を求める場合には，(6-15)で初期値 $\boldsymbol{x}(0) = \boldsymbol{0}$ とおいて，

$$(z\boldsymbol{I} - \boldsymbol{A})\boldsymbol{X}(z) = \boldsymbol{b}\,R(z) \tag{6-17}$$



左から，$(z\boldsymbol{I} - \boldsymbol{A})^{-1}$ を掛けて，更に，左から $\boldsymbol{c}$ を掛けて，

$$Y(z) = \boldsymbol{c}(z\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{b}\,R(z) \tag{6-18}$$

よって，

$$\begin{aligned} G(z) &= \frac{Y(z)}{R(z)} = \boldsymbol{c}(z\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{b} \\ &= \frac{\boldsymbol{c}\ \mathrm{adj}(z\boldsymbol{I} - \boldsymbol{A})\ \boldsymbol{b}}{\left| z\boldsymbol{I} - \boldsymbol{A} \right|} \end{aligned} \tag{6-19}$$

従って，特性方程式は，

$$\left| z\boldsymbol{I} - \boldsymbol{A} \right| = 0 \tag{6-20}$$

となる。この根（極）は，$\boldsymbol{A}$ の固有値と一致することが判る。固有値は習慣的に λ，特性

方程式の根はディジタル制御では z の記号を用いるが，両者は同じ値になる。従って，(6-13)の条件は， z 変換で求めた安定条件(4-7)と全く同じことを言っている。なお周波数応答は(6-19)より計算できる。

◎　逆行列の公式

$\boldsymbol{A}$ の第 i 行と第 j 列を省いてできた $(n-1)\times(n-1)$ 次元の行列の行列式に $(-1)^{i+j}$ を掛けたものを M_{ij} とする。

$$\boldsymbol{A}=\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & & a_{2n} \\ \vdots & & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \quad \text{のとき，例えば，} \quad M_{21}=(-1)^{2+1}\begin{vmatrix} a_{12} & a_{13} & \cdots & a_{1n} \\ a_{32} & a_{33} & & a_{3n} \\ \vdots & & \ddots & \vdots \\ a_{n2} & a_{n3} & \cdots & a_{nn} \end{vmatrix}$$

このとき，

行列式： $$|\boldsymbol{A}|=\sum_{j=1}^{n} a_{ij} M_{ij} = \sum_{i=1}^{n} a_{ij} M_{ij}$$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{\underbrace{ad-bc}_{|\boldsymbol{A}|}} \overbrace{\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}}^{\mathrm{adj}(\boldsymbol{A})}$$

逆行列： $$\boldsymbol{A}^{-1}=\frac{\mathrm{adj}(\boldsymbol{A})}{|\boldsymbol{A}|}$$

余因子行列： $$\mathrm{adj}(\boldsymbol{A})=\begin{bmatrix} M_{11} & \cdots & M_{1n} \\ \vdots & M_{ij} & \vdots \\ M_{n1} & \cdots & M_{nn} \end{bmatrix}^{T}$$ 　T は，転置行列を意味する。

[問題 6-2]　次式で示されるディジタル制御系，

$$\boldsymbol{x}(k+1)=\boldsymbol{A}\boldsymbol{x}(k)+\boldsymbol{b}r(k)$$

が安定であるとき，定常値 $\boldsymbol{x}(\infty)$ を求めよ。ただし， $r(k)$ は一定値 r とする。また，(6-12)より，得られる結果と比較せよ。

（答）定常状態なので， $\boldsymbol{x}(k+1)=\boldsymbol{x}(k)=\boldsymbol{x}(\infty)$ と置くと， $\boldsymbol{A}=\boldsymbol{P}\boldsymbol{Q}\boldsymbol{P}^{-1}$ より

$$\boldsymbol{x}(\infty)=(\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{b}r=(\boldsymbol{P}(\boldsymbol{I}-\boldsymbol{Q})\boldsymbol{P}^{-1})^{-1}\boldsymbol{b}r=\boldsymbol{P}(\boldsymbol{I}-\boldsymbol{Q})^{-1}\boldsymbol{P}^{-1}\boldsymbol{b}r$$

$$=\boldsymbol{P}\begin{bmatrix} 1/(1-\lambda_1) & & 0 \\ & \ddots & \\ 0 & & 1/(1-\lambda_n) \end{bmatrix}\boldsymbol{P}^{-1}\boldsymbol{b}r$$

この結果は，安定なら $|\lambda_i|<1$ なので，(6-12)で $k\to\infty$ とした結果と一致する。

[問題 6-3]　ディジタル制御系

$$x_1(k+1)=x_1(k)+2x_2(k)+r(k) \qquad ①$$

$$x_2(k+1)=2x_1(k)+x_2(k) \qquad ②$$

がある。ただし，$x_1(0)=1, x_2(0)=1$ とする。指令値 $r(k)=1$ とするとき，$x_1(k), x_2(k)$ を次の２つの方法で求めよ。この系は安定といえるか。

(1)　z 変換と逆 z 変換を用いる方法

(2)　行列の対角化による方法

（答）(1) ①を z 変換して

$$zX_1 - z\,x_1(0) = X_1 + 2X_2 + R \qquad \therefore \quad zX_1 - z = X_1 + 2X_2 + \frac{z}{z-1} \qquad ③$$

②を z 変換して初期値を代入すると

$$zX_2 - z = 2X_1 + X_2 \qquad ④$$

③，④を解いて

$$X_1 = \frac{z^2+2z}{z^2-2z-3},\quad X_2 = \frac{z(z^2+1)}{(z^2-2z-3)(z-1)}$$

$$\therefore \quad \frac{X_1}{z} = \frac{z+2}{(z-3)(z+1)},\quad \frac{X_2}{z} = \frac{z^2+1}{(z-3)(z+1)(z-1)}$$

部分分数展開してから両辺に z を掛け，逆 z 変換すると

$$x_1(k) = \frac{5}{4}3^k - \frac{1}{4}(-1)^k \quad , \quad x_2(k) = -\frac{1}{2} + \frac{5}{4}3^k + \frac{1}{4}(-1)^k \qquad ⑤$$

$t \to \infty$ のとき，ともに∞となり，不安定である。

(2) ①，②を行列表示して

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} r(k)$$

これを，$\boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{b}r(k)$　書くと，$\boldsymbol{A}$ を対角化して

$$\boldsymbol{A} = \boldsymbol{PQP}^{-1} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 3 & 0 \\ 0 & -1 \end{bmatrix} (\frac{1}{2}) \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

(6-12)より　$\boldsymbol{x}(k) = (\frac{1}{2}) \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 3^k & 0 \\ 0 & (-1)^k \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

$$+(\frac{1}{2}) \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \dfrac{1-3^k}{1-3} & 0 \\ 0 & \dfrac{1-(-1)^k}{1+1} \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} 1$$

これより⑤が得られる。なお，3,-1 が固有値 λ_1, λ_2 に対応する。

[問題 6-4]　ディジタル制御系

$$x(k+2)+x(k+1)+x(k)=r(k)$$

が安定な系か調べよ。ただし，$r(k)$ は指令値である。

（答）$x_1(k)=x(k)$,$x_2(k)=x(k+1)=x_1(k+1)$　 とおくと与式より

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix}=\begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix}\begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}+\begin{bmatrix} 0 \\ 1 \end{bmatrix}r(k)$$

$\begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix}$ の固有値は，$\lambda=\dfrac{-1\pm j\sqrt{3}}{2}$ であり，

$|\lambda|=1$ なので安定限界（不安定）である。

6.3　ＰおよびＰＩディジタル制御系

本節では，(6-9)の状態方程式を導く手順について説明する。図 6-2 に示すディジタル制御系を考える。制御器としては，Ｐ(比例）またはＰＩ（比例＋積分）制御とする。

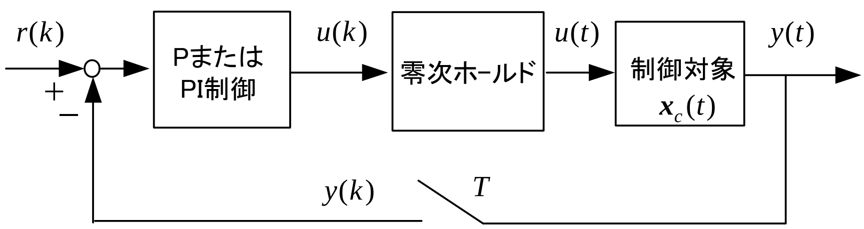


図 6-2 ＰまたはＰＩディジタル制御系

制御器に関しては差分方程式で表されることを既に 3 章で示した。そこで，零次ホールドと制御対象の部分について差分方程式を導き，あとで両者を組み合わせて系全体の差分方程式を求めよう。

制御対象は連続系だから，微分方程式をまとめた状態方程式と出力方程式で表すことができる。添字 c を用いる。

制御対象の状態方程式

$$\frac{d\boldsymbol{x}_c(t)}{dt}=\boldsymbol{A}_c\boldsymbol{x}_c(t)+\boldsymbol{b}_c u(t) \tag{6-21}$$

制御対象の出力方程式

$$y(t)=\boldsymbol{c}_c\boldsymbol{x}_c(t) \tag{6-22}$$

ここで，$\boldsymbol{x}_c$ ：状態変数（$m\times 1$ のベクトル），u：入力，y：出力

$\boldsymbol{A}_c$ ：$m\times m$ の行列，$\boldsymbol{b}_c$ ：$m\times 1$ のベクトル，$\boldsymbol{c}_c$ ：$1\times m$ のベクトル

(6-21)の解は，次式で与えられる。（初期値を満足し，(6-23)を微分すると(6-21)になる。）

$$\boldsymbol{x}_c(t) = e^{\boldsymbol{A}_c t}\boldsymbol{x}_c(0) + \int_0^t e^{\boldsymbol{A}_c(t-\tau)}\boldsymbol{b}_c u(\tau)\,d\tau \tag{6-23}$$

ここで，**状態遷移行列**(state transition matrix)と呼ばれる $e^{\boldsymbol{A}t}$ は，次式で与えられる。

$$e^{\boldsymbol{A}t} = \boldsymbol{I} + \boldsymbol{A}t + \frac{\boldsymbol{A}^2t^2}{2!} + \frac{\boldsymbol{A}^3t^3}{3!} + \cdots \tag{6-24}$$

上式より次の性質が成り立つ。

(1) $\dfrac{d}{dt}e^{\boldsymbol{A}t} = \boldsymbol{A}e^{\boldsymbol{A}t}$

(2) $e^{\boldsymbol{A}(t+\tau)} = e^{\boldsymbol{A}t}e^{\boldsymbol{A}\tau}$　　一般に $e^{\boldsymbol{A}+\boldsymbol{B}} \neq e^{\boldsymbol{A}}e^{\boldsymbol{B}}$

(3) $e^{\boldsymbol{A}t} = L^{-1}(s\boldsymbol{I}-\boldsymbol{A})^{-1}$　　L^{-1}：逆ラプラス変換

[例題 6-1]　$\boldsymbol{A} = \begin{bmatrix} -2 & -1 \\ 2 & -5 \end{bmatrix}$ のとき，$e^{\boldsymbol{A}t}$ を求めよ。

(解)　$e^{\boldsymbol{A}t} = L^{-1}(s\boldsymbol{I}-\boldsymbol{A})^{-1} = L^{-1}\begin{bmatrix} s+2 & 1 \\ -2 & s+5 \end{bmatrix}^{-1} = L^{-1}\dfrac{1}{(s+3)(s+4)}\begin{bmatrix} s+5 & -1 \\ 2 & s+2 \end{bmatrix}$

$$= L^{-1}\begin{bmatrix} \dfrac{2}{s+3} - \dfrac{1}{s+4} & \dfrac{-1}{s+3} + \dfrac{1}{s+4} \\ \dfrac{2}{s+3} - \dfrac{2}{s+4} & \dfrac{-1}{s+3} + \dfrac{2}{s+4} \end{bmatrix} = \begin{bmatrix} 2e^{-3t} - e^{-4t} & -e^{-3t} + e^{-4t} \\ 2e^{-3t} - 2e^{-4t} & -e^{-3t} + 2e^{-4t} \end{bmatrix}$$

図 6-3 に制御対象の入力と状態変数の時間変化を示す。$\boldsymbol{x}_c(kT)$ と $\boldsymbol{x}_c((k+1)T)$ の関係を導出しよう。$kT \le t < (k+1)T$ において，零次ホールドにより入力が $u(t) = u(kT)$（一定）とする。(6-23)より $t = (k+1)T$ とおいて

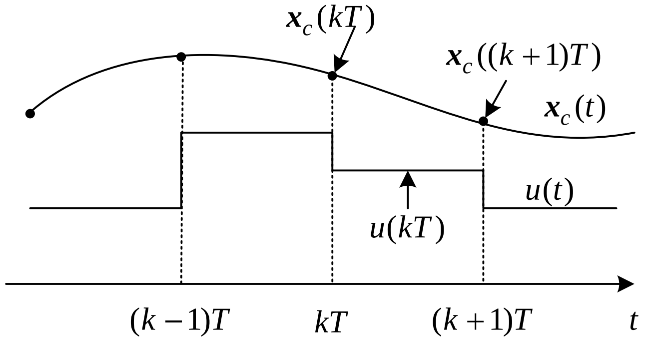


図 6-3　制御対象の入力と状態変数（$\boldsymbol{x}_c$ は実際はベクトル）

$$\boldsymbol{x}_c((k+1)T) = e^{\boldsymbol{A}_c(k+1)T}\boldsymbol{x}_c(0) + \int_0^{(k+1)T} e^{\boldsymbol{A}_c((k+1)T-\tau)}\boldsymbol{b}_c u(\tau)d\tau$$

$$= e^{\boldsymbol{A}_cT}(e^{\boldsymbol{A}_ckT}\boldsymbol{x}_c(0) + \int_0^{kT} e^{\boldsymbol{A}_c(kT-\tau)}\boldsymbol{b}_c u(\tau)d\tau) + \int_{kT}^{(k+1)T} e^{\boldsymbol{A}_c((k+1)T-\tau)}\boldsymbol{b}_c u(\tau)d\tau$$

$$= e^{\boldsymbol{A}_cT}\boldsymbol{x}_c(kT) + \int_{kT}^{(k+1)T} e^{\boldsymbol{A}_c((k+1)T-\tau)}\boldsymbol{b}_c u(\tau)d\tau$$

第２項で，$kT \le t < (k+1)T$ で $u(t) = u(kT)$ （一定）として積分の外に出し，

$(k+1)T - \tau = \tau'$

とおき，変数変換すると，$-d\tau = d\tau'$, $\tau : kT \to (k+1)T$ のとき $\tau' : T \to 0$ だから，

$$第２項 = -\int_T^0 e^{\boldsymbol{A}_c\tau'}d\tau'\boldsymbol{b}_c u(kT) = \int_0^T e^{\boldsymbol{A}_c\tau'}d\tau'\boldsymbol{b}_c u(kT) = \int_0^T e^{\boldsymbol{A}_c\tau}d\tau\boldsymbol{b}_c u(kT)$$

従って，

$$\boldsymbol{x}_c((k+1)T) = e^{\boldsymbol{A}_cT}\boldsymbol{x}_c(kT) + \int_0^T e^{\boldsymbol{A}_c\tau}d\tau\boldsymbol{b}_c u(kT) \qquad (6\text{-}25)$$

これを簡単に，

$$\boldsymbol{x}_c(k+1) = e^{\boldsymbol{A}_cT}\boldsymbol{x}_c(k) + \int_0^T e^{\boldsymbol{A}_c\tau}d\tau\boldsymbol{b}_c u(k)$$

$$\equiv \boldsymbol{A}_p\boldsymbol{x}_c(k) + \boldsymbol{b}_p u(k) \qquad (6\text{-}26)$$

ただし，$\boldsymbol{A}_p = e^{\boldsymbol{A}_cT}$，$\boldsymbol{b}_p = \int_0^T e^{\boldsymbol{A}_c\tau}d\tau\boldsymbol{b}_c$

と書く。これで，零次ホールドと制御対象の差分方程式が導出できた。(6-26)は，(6-21)を厳密に解いて，サンプリング周期ごとの値の関係を表したものである。

○ ディジタルP制御時のシステム全体の状態方程式

図 6-4 にディジタル P 制御系を示す。零次ホールドと制御対象の差分方程式が(6-26)で導出できたので，P 制御の式を加えて制御系全体の状態方程式を導出しよう。

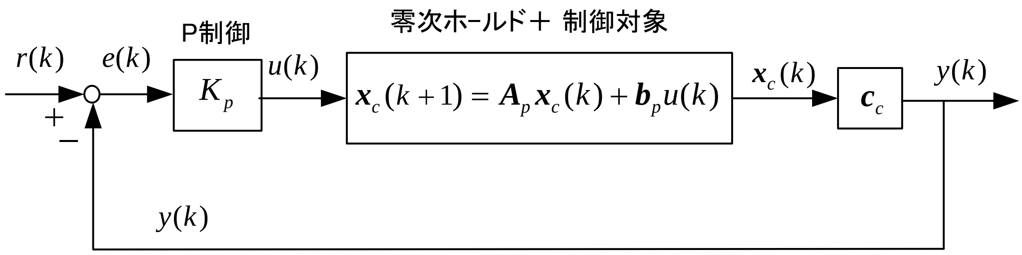


図 6-4　ディジタル P 制御系

ディジタルＰ制御は次式で与えられる。

$$u(k)=K_p(r(k)-y(k)) \tag{6-27}$$

(6-22)より，$y(k)=\boldsymbol{c}_c\boldsymbol{x}_c(k)$ だから，(6-27)に代入して

$$u(k)=K_p(r(k)-\boldsymbol{c}_c\boldsymbol{x}_c(k)) \tag{6-28}$$

(6-28)を(6-26)に代入して，次式のシステム全体の状態方程式が得られる。

$$\boldsymbol{x}_c(k+1)=(\boldsymbol{A}_p-K_p\boldsymbol{b}_p\boldsymbol{c}_c)\boldsymbol{x}_c(k)+K_p\boldsymbol{b}_p r(k) \tag{6-29}$$

この式は，(6-9)に対応し，

$$\boldsymbol{x}(k)=\boldsymbol{x}_c(k),\ \boldsymbol{A}=\boldsymbol{A}_p-K_p\boldsymbol{b}_p\boldsymbol{c}_c,\ \boldsymbol{b}=K_p\boldsymbol{b}_p \tag{6-30}$$

である。P 制御の場合には状態変数は増えない。系行列 $\boldsymbol{A}$ の固有値でシステム全体の安定判別ができる。K_p を変えると，その固有値が変化することは言うまでもない。なお，出力方程式は変数が変わらないので，そのまま利用でき，

$$y(k)=\boldsymbol{c}_c\boldsymbol{x}(k) \tag{6-31}$$

である。

○ ディジタルPI制御時のシステム全体の状態方程式

図 6-5 にディジタル PI 制御系を示す。PI 制御の式を加えて制御系全体の状態方程式を導出しよう。

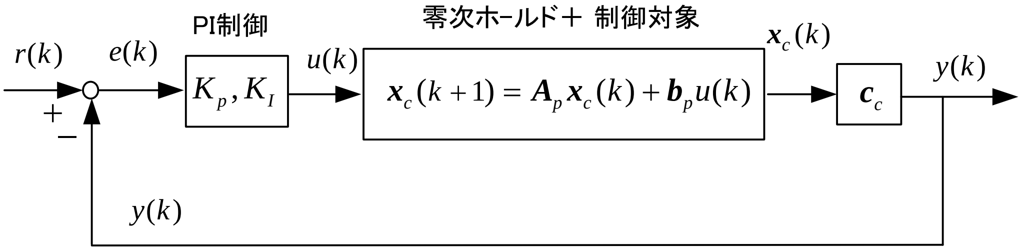


図 6-5　ディジタル PI 制御系

ディジタルＰＩ制御については，次の３つの方法を述べた。

① $$u(k)=K_P e(k)+K_I\sum_{m=0}^{k}e(m-1)T \quad \text{（前進矩形近似）} \tag{6-32}$$

② $$u(k)=K_P e(k)+K_I\sum_{m=0}^{k}e(m)T \quad \text{（後退矩形近似）} \tag{6-33}$$

③ $$u(k)=K_P e(k)+\frac{K_I}{2}\sum_{m=0}^{k}(e(m)+e(m-1))T$$ （台形近似） (6-34)

ここで， $e(k)=r(k)-y(k)$

ここでは, (6-33)の後退矩形近似のＰＩ制御器を考える。実際に，マイコンで演算する場合，(6-33)ではなく(3-16)で演算するが，システム全体の状態方程式を導く場合には，(6-33)より出発した方が都合が良い。

新たに，変数として次式を定義する。

$$w(k)\equiv\sum_{m=0}^{k}e(m) \tag{6-35}$$

$w(k-1)=\sum_{m=0}^{k-1}e(m)$ だから，次式が得られる。

$$w(k)=w(k-1)+e(k)=w(k-1)+r(k)-y(k)=w(k-1)+r(k)-\boldsymbol{c}_c\,\boldsymbol{x}_c(k) \tag{6-36}$$

また，(6-33)より，

$$u(k)=K_P e(k)+K_I T w(k)$$

$$=(K_P+K_I T)(r(k)-\boldsymbol{c}_c\boldsymbol{x}_c(k))+K_I T w(k-1) \tag{6-37}$$

(6-26)に代入して，

$$\boldsymbol{x}_c(k+1)=(\boldsymbol{A}_p-(K_p+K_I T)\boldsymbol{b}_p\boldsymbol{c}_c)\boldsymbol{x}_c(k)+K_I T\,\boldsymbol{b}_p w(k-1)+(K_p+K_I T)\boldsymbol{b}_p r(k) \tag{6-38}$$

(6-36), (6-38)より

$$\begin{bmatrix}\boldsymbol{x}_c(k+1)\\ w(k)\end{bmatrix}=\begin{bmatrix}\boldsymbol{A}_p-(K_p+K_I T)\boldsymbol{b}_p\boldsymbol{c}_c & K_I T\,\boldsymbol{b}_p\\ -\boldsymbol{c}_c & 1\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_c(k)\\ w(k-1)\end{bmatrix}+\begin{bmatrix}(K_p+K_I T)\boldsymbol{b}_p\\ 1\end{bmatrix}r(k) \tag{6-39}$$

が得られる。ＰＩ制御の場合には，変数が１つ増えて，(6-9)に以下のように対応する。

$$\boldsymbol{x}(k)=\begin{bmatrix}\boldsymbol{x}_c(k)\\ w(k-1)\end{bmatrix},\ \boldsymbol{A}=\begin{bmatrix}\boldsymbol{A}_p-(K_p+K_IT)\boldsymbol{b}_p\boldsymbol{c}_c & K_IT\,\boldsymbol{b}_p\\ -\boldsymbol{c}_c & 1\end{bmatrix},\ \ \boldsymbol{b}=\begin{bmatrix}(K_p+K_IT)\boldsymbol{b}_p\\ 1\end{bmatrix} \qquad (6\text{-}40)$$

$\boldsymbol{x}(k)$ の成分は，k にそろっていないがこれは全く問題ない。$w(k)\equiv w'(k+1)$ と定義しなおせば済むことで，わざわざそうする必要もない。数学的に差分方程式の形になっていればよいのである。(6-37)で，$w(k)$ を $w(k-1)$ で表わした理由は，差分方程式の形を作るためであった。$\boldsymbol{A}$ の固有値の絶対値が全て 1 以下なら，この図 6-5 のＰＩ制御系は安定である。

[問題 6-5]　ＰＩ制御に(6-34)の台形近似を用いたとき，制御系全体の差分方程式を導け。制御対象は，次式で与えられるものとする。

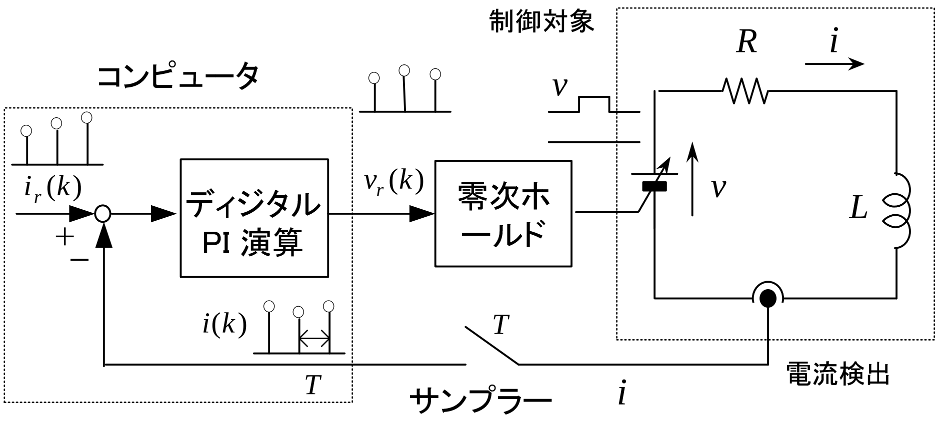
$$\boldsymbol{x}_c(k+1)=\boldsymbol{A}_p\boldsymbol{x}_c(k)+\boldsymbol{b}_pu(k)$$

$$y(k)=\boldsymbol{c}_c\boldsymbol{x}_c(k)$$

（答）$w(k)=\sum_{m=0}^{k}(r(m)-y(m))$　とおく。

$$\begin{bmatrix}\boldsymbol{x}_c(k+1)\\ w(k)\end{bmatrix}=\begin{bmatrix}\boldsymbol{A}_p-(K_p+\frac{K_IT}{2})\boldsymbol{b}_p\boldsymbol{c}_c & K_IT\,\boldsymbol{b}_p\\ -\boldsymbol{c}_c & 1\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_c(k)\\ w(k-1)\end{bmatrix}+\begin{bmatrix}(K_p+\frac{K_IT}{2})\boldsymbol{b}_p\\ 1\end{bmatrix}r(k)$$

[問題 6-6]　図に示すＲＬ回路のディジタル電流ＰＩ制御系で，制御系全体の状態方程式を示し，特性方程式を求めよ。ただし，ＰＩ制御には，後退矩形近似を用いる。



(答)　制御対象の微分方程式は，

$$v=L\frac{di}{dt}+Ri \qquad \therefore \frac{di}{dt}=-\frac{R}{L}i+\frac{1}{L}v \qquad (6\text{-}41)$$

零次ホールド回路により，サンプリング周期ごとに v は一定値 $v_r(k)$ だから，

$$
\begin{aligned}
i(k+1) &= e^{-\frac{R}{L}T} i(k) + \int_0^T e^{-\frac{R}{L}\tau} d\tau \frac{1}{L} v_r(k) \\
&= e^{-\frac{R}{L}T} i(k) + \frac{1}{R}(1 - e^{-\frac{R}{L}T}) v_r(k) \\
&\equiv a\, i(k) + b\, v_r(k)
\end{aligned}
\tag{6-42}
$$

ここで，$a = e^{-\frac{R}{L}T}$，$b = (1-a)/R$

[　(6-26)を用いたが，(1-23)でもよい．]

PI コントローラについては，後退矩形近似を用いると

$$
v_r(k) = K_P(i_r(k) - i(k)) + K_I T \sum_{m=0}^{k} (i_r(m) - i(m)) \tag{6-43}
$$

ここで，$w(k) \equiv \sum_{m=0}^{k} (i_r(m) - i(m))$　(6-44)

とおくと，

$$
w(k) = w(k-1) + i_r(k) - i(k) \tag{6-45}
$$

(6-44)を(6-43)に代入し，それを(6-42)に代入した式と(6-45)より，システム全体の状態方程式が以下のように得られる．

$$
\begin{bmatrix} i(k+1) \\ w(k) \end{bmatrix} = \begin{bmatrix} a - (K_p + K_I T)b & K_I T\, b \\ -1 & 1 \end{bmatrix} \begin{bmatrix} i(k) \\ w(k-1) \end{bmatrix} + \begin{bmatrix} (K_p + K_I T)b \\ 1 \end{bmatrix} i_r(k) \tag{6-46}
$$

特性方程式は，以下のように計算できる．

$$
\begin{aligned}
\left| z\boldsymbol{I} - \boldsymbol{A} \right| &= \begin{vmatrix} z - a + (K_p + K_I T)b & -K_I T\, b \\ 1 & z-1 \end{vmatrix} \\
&= z^2 + \left\{ (K_p + K_I T)b - a - 1 \right\} z + a - K_p b = 0
\end{aligned}
\tag{6-47}
$$

当然ながら，特性方程式は，例題 4-3 と一致する。

(6-46)に，**有限整定制御（デッドビート制御）**を適用してみよう。一般にディジタル制御系全体の状態方程式が，

$$
\boldsymbol{x}(k+1) = \boldsymbol{A}\,\boldsymbol{x}(k) + \boldsymbol{b}\, r(k)
$$

で与えられるとき，$\boldsymbol{A}$ の固有値を 0 に設定することで，**有限整定制御**が実現できる。固有値は，(6-47)で求まるので，それを 0 とする条件は，

$$
(K_p + K_I T)b - a - 1 = 0 \quad , \quad a - K_p b = 0
$$

よって，　$K_p = \dfrac{a}{b}$　，　$K_I = \dfrac{1}{Tb}$　(6-48)

当然ながら，例題 4-3 の結果と一致する。

実際に，電流の指令値 $i_r(k)$ の単位ステップ変化に対する応答を求める。

(6-46)より，(6-48)を代入して

$$\begin{bmatrix} i(k+1) \\ w(k) \end{bmatrix} = \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} i(k) \\ w(k-1) \end{bmatrix} + \begin{bmatrix} 1+a \\ 1 \end{bmatrix} i_r(k) \tag{6-49}$$

である，$k=0,1,2,\cdots$を(6-49)に代入して順に値を求める。

$i_r(k)=1$, $i(0)=0$, $w(-1)=0$ として，

$$\begin{bmatrix} i(1) \\ w(0) \end{bmatrix} = \begin{bmatrix} 1+a \\ 1 \end{bmatrix} \quad , \quad \begin{bmatrix} i(2) \\ w(1) \end{bmatrix} = \begin{bmatrix} 1 \\ 1-a \end{bmatrix}$$

$$\begin{bmatrix} i(3) \\ w(2) \end{bmatrix} = \begin{bmatrix} 1 \\ 1-a \end{bmatrix} \quad \cdots$$

が得られ，$k=2$ 以降値は変化しない。この結果も，例題 4−3 の結果と一致する。

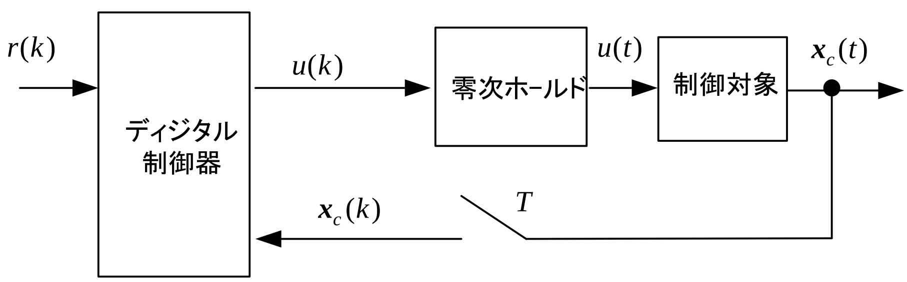
6.4 一般的なディジタル制御系の解析

○ 一般のディジタル制御システムの解析

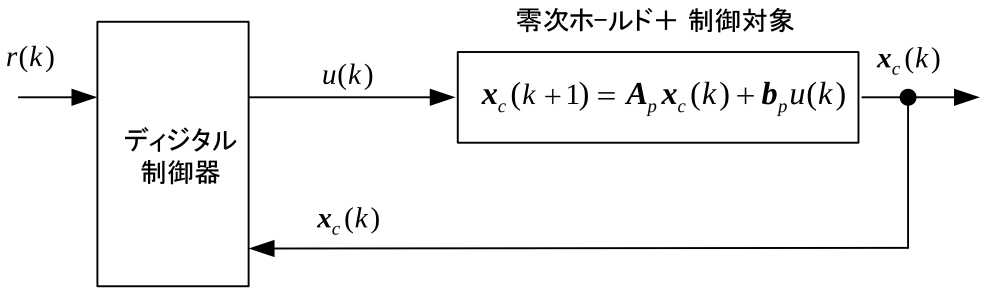
［例題 2-5］のマイナーループを有するような一般的なディジタル制御系の状態方程式の導出を考える。図 6-6 に一般的なディジタル制御系を示す。

まず，零次ホールドと制御対象の差分方程式は既に(6-26)に示した。

$$\boldsymbol{x}_c(k+1) = \boldsymbol{A}_p \boldsymbol{x}_c(k) + \boldsymbol{b}_p u(k) \tag{6-26}$$



(a) 実際のシステム



（b）零次ホールド＋制御対象を差分方程式で表したモデル

図 6-6　一般的なディジタル制御系

次に，ディジタル制御器の差分方程式を求めると

$$\boldsymbol{z}(k+1) = \boldsymbol{A}_w \boldsymbol{z}(k) + \boldsymbol{B}_x\, \boldsymbol{x}_c(k) + \boldsymbol{B}_r\, r(k) \tag{6-50}$$

のように書くことができる。PI 制御の場合は(6-36)がこれに相当する（$\boldsymbol{z}(k+1) = w(k)$ とおけばよい）。入力は

$$u(k) = \boldsymbol{F}_x\, \boldsymbol{x}_c(k) + \boldsymbol{F}_w\, \boldsymbol{z}(k) + \boldsymbol{F}_r\, r(k) \tag{6-51}$$

と書ける（(6-37)参照）。(6-26)，(6-50)，(6-51)よりシステム全体の状態方程式は次式で表せる。

$$\begin{bmatrix} \boldsymbol{x}_c(k+1) \\ \boldsymbol{z}(k+1) \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_p + \boldsymbol{b}_p \boldsymbol{F}_x & \boldsymbol{b}_p \boldsymbol{F}_w \\ \boldsymbol{B}_x & \boldsymbol{A}_w \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_c(k) \\ \boldsymbol{z}(k) \end{bmatrix} + \begin{bmatrix} \boldsymbol{b}_p \boldsymbol{F}_r \\ \boldsymbol{B}_r \end{bmatrix} r(k) \tag{6-52}$$

これにより安定解析や応答の計算などが行える。

○ ディジタル制御器の演算時間を考慮した解析 (文献(16))

1 サンプリング周期だけ入力が遅れる図 4-7 の制御系を今度は時間領域で解析しよう。

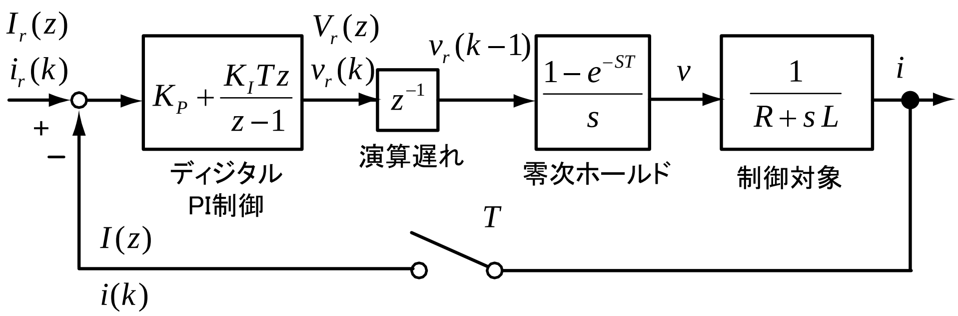


図 4-7　1 サンプリングの演算遅れを考慮したディジタル PI 制御系 (再掲)

(6-42)より，演算遅れがあるので，$kT \le t < (k+1)T$ の電源電圧は $v_r(k-1)$ だから，

$$i(k+1) = a\,i(k) + b\,v_r(k-1) \tag{6-53}$$

ただし， $a = \exp(-RT/L),\ b = (1-a)/R$

PI コントローラについては，後退矩形近似を用いているので(6-43)より

$$v_r(k) = K_P(i_r(k) - i(k)) + K_I T\,w(k) \tag{6-54}$$

(6-45)より，

$$w(k) = w(k-1) + i_r(k) - i(k) \tag{6-55}$$

(6-55)を(6-54)に代入して

$$v_r(k) = (K_P + K_I T)(i_r(k) - i(k)) + K_I T\,w(k-1)$$

よって，

$$\begin{bmatrix} i(k+1) \\ v_r(k) \\ w(k) \end{bmatrix} = \begin{bmatrix} a & b & 0 \\ -(K_p + K_I T) & 0 & K_I T \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} i(k) \\ v_r(k-1) \\ w(k-1) \end{bmatrix} + \begin{bmatrix} 0 \\ K_p + K_I T \\ 1 \end{bmatrix} i_r(k) \tag{6-56}$$

演算遅れがある場合には，入力を新しく状態変数に選ぶことで状態方程式が得られる。

状態方程式は　$\boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{b}r(k)$　であり，

$$\boldsymbol{x}(k) = \begin{bmatrix} i(k) \\ v_r(k-1) \\ w(k-1) \end{bmatrix},\ r(k) = i_r(k)$$

である。(　)の中が k に揃っていなくても，＋１したものが $\boldsymbol{x}(k+1)$ になっていれば良い。

(6-56)より，$\boldsymbol{A}$ の固有値を求める特性方程式は

$$\begin{vmatrix} z-a & -b & 0 \\ K_p + K_I T & z & -K_I T \\ 1 & 0 & z-1 \end{vmatrix} = 0$$

$$\therefore \quad z^3 + \alpha z^2 + \beta z + \gamma = 0 \tag{6-57}$$

但し，　$\alpha = -(1+a),\ \beta = a + b(K_p + K_I T),\ \gamma = -K_P b$

となる。これは，4.3 節①の特性方程式と一致している。このように時間遅れがある場合でも状態方程式による解析が可能である。

○　連続系とディジタル制御系の固有値の関係

指令は安定性に無関係だから 0 とすると，連続系の状態方程式は

$$\frac{d\boldsymbol{x}(t)}{dt} = \boldsymbol{A}\,\boldsymbol{x}(t) \tag{6-58}$$

で表せる。これをサンプリング周期 T で離散化すると次式を得る。

$$\boldsymbol{x}(k+1) = e^{\boldsymbol{A}T}\boldsymbol{x}(k) \tag{6-59}$$

(6-59)はディジタル制御系である。逆に $\boldsymbol{x}(k+1) = \boldsymbol{B}\,\boldsymbol{x}(k)$ のディジタル制御系（例えば(6-39)）が得られたなら，$\boldsymbol{B}$ の固有値とそれに対応した連続系の固有値の関係は，(6-58)と(6-59)の関係から求められよう。

$e^{\boldsymbol{A}T}$ の固有値を λ_i，$\boldsymbol{A}$ の固有値を s_i とすると次式の関係が証明できる（文献(5)）。

$$\lambda_i = e^{s_i T} \qquad (i = 1, 2, \cdots, n) \tag{6-60}$$

証明） $\boldsymbol{A}$ の固有値 s_i に対する固有ベクトルを $\boldsymbol{u}_i$ とすると

$$\boldsymbol{A}\boldsymbol{u}_i = s_i\,\boldsymbol{u}_i \quad , \quad \boldsymbol{A}^2\boldsymbol{u}_i = \boldsymbol{A}\,s_i\,\boldsymbol{u}_i = s_i\,\boldsymbol{A}\boldsymbol{u}_i = s_i^2\ \boldsymbol{u}_i$$

同様にして，$\boldsymbol{A}^n\boldsymbol{u}_i = s_i^n\ \boldsymbol{u}_i$

従って，
$$\begin{aligned} e^{\boldsymbol{A}T}\boldsymbol{u}_i &= (\boldsymbol{I} + \boldsymbol{A}T + \frac{\boldsymbol{A}^2T^2}{2!} + \frac{\boldsymbol{A}^3T^3}{3!} + \cdots)\boldsymbol{u}_i \\ &= (1 + s_iT + \frac{s_i^2T^2}{2!} + \frac{s_i^3T^3}{3!} + \cdots)\boldsymbol{u}_i \\ &= e^{s_iT}\boldsymbol{u}_i \end{aligned}$$

よって，$e^{\boldsymbol{A}T}$ の固有値は e^{s_iT} でそれに対する固有ベクトルは $\boldsymbol{u}_i$ であることが判る。（終）

いま，$s_i = \alpha + j\omega$ とすると，

$$\lambda_i = e^{s_iT} = e^{\alpha T}(\cos\omega T + j\sin\omega T) \tag{6-61}$$

$\alpha < 0$ のとき，

$$|\lambda_i| = e^{\alpha T} < 1 \tag{6-62}$$

であるから，連続系の左半面の根が z 平面の単位円内に対応し安定条件となっている。

(6-61), (6-62)は，図 4-9 で示した極（固有値と同じ）の軌跡を考える場合役立つ。すなわち，連続系の α, ω に対応する量が，ディジタル制御系ではそれぞれ原点からの距離と偏角であることが判る。

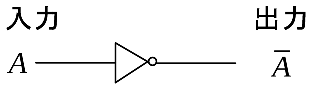
第 7 章　マイコン制御システム

はじめに論理回路と 2 進数のかんたんな話をしよう。

○ 論理回路

基本となる論理回路である NOT 回路，AND 回路, NAND 回路，OR 回路，NOR 回路について真理値表，ベン図を以下に示す。記号は MIL 規格(military standard)を用いている。ここでは高電位 H を 1，低電位 L を 0 に対応させた **H 駆動（正論理）**(Active High)を考える。高電位 H を 0，低電位 L を 1 に対応させた **L 駆動(負論理)**(Active Low)は後で述べる。

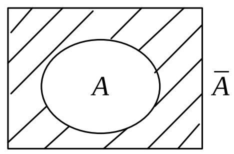
(1)　NOT 回路



論理回路

A	$\overline{A}$
0	1
1	0

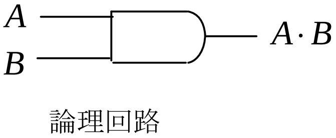
真理値表



ベン図

NOT とは否定の意味である。論理回路で A が入力， $\overline{A}$ が出力である。 $\overline{A}$ は A の NOT を表わしている。真理値表は，入力と出力の関係を示す。変数には 0, 1 以外の値はない。**ベン図**はもともと集合についての関係を表わすものとして考案されたが，論理演算を視覚的に分かりやすく表現するものとしてしばしば用いられている。円の中が $A=1$，円の外が $A=0$ を意味する。斜線部は出力が 1 となる領域を表す。ベン図の円の中の A が 0 か 1 と考えてはいけない。NOT 回路を２つつなぐと元に戻る。

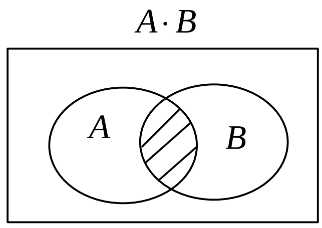
(2) AND 回路



論理回路

A	B	$A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

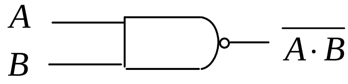
真理値表



ベン図

論理積回路とも言われる。出力は論理式で $A \cdot B$ と書き，掛け算の結果と一致するので分かりやすい。ベン図の斜線領域は出力 $A \cdot B$ が 1 となる領域を示す。

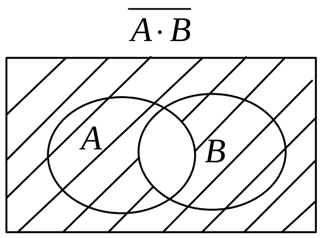
(3) NAND 回路



論理回路

A	B	$\overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

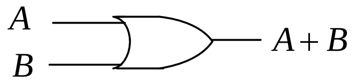
真理値表



ベン図

NAND は AND と NOT を組み合わせたものである。○印が NOT に相当する。$\overline{A \cdot B}$ は $A \cdot B$ の NOT である。NAND に NOT をつなぐと AND になる。

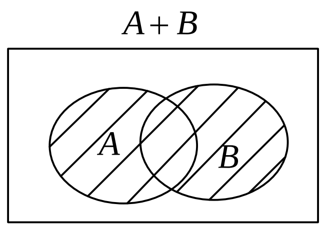
(4) OR 回路



論理回路

A	B	$A+B$
0	0	0
0	1	1
1	0	1
1	1	1

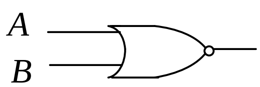
真理値表



ベン図

OR の出力は論理式で $A+B$ と表現される。どちらかの入力が 1 なら出力も 1 になる。

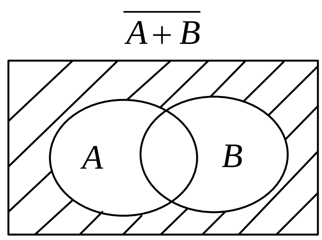
(5) NOR 回路



論理回路

A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

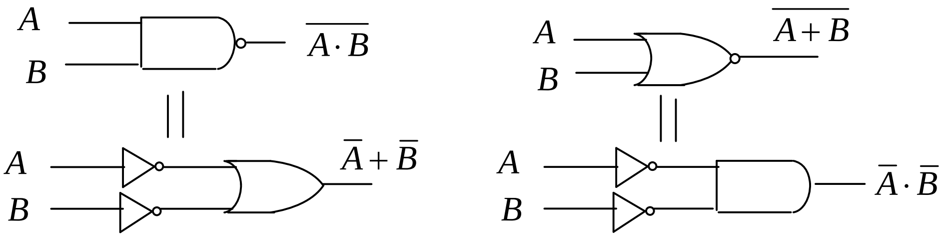
真理値表



ベン図

ド・モルガンの定理より成り立つ等価な論理回路を示す。形式的には反転の○を入力側に移すと AND は OR，OR は AND に変化する。

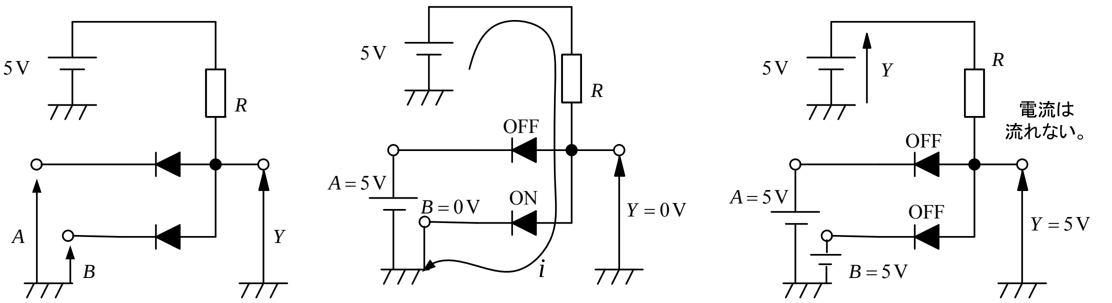
ド・モルガンの定理： $\overline{A \cdot B} = \overline{A} + \overline{B}$，$\overline{A+B} = \overline{A} \cdot \overline{B}$



等価な論理回路（ド・モルガンの定理）

H 駆動(Active High)と **L 駆動**(Active low)について，簡単に述べておこう。ディジタル信号で，何かの動作をしようとするとき，その出力ピンの電圧が 5V で目的を果たす場合と 0V で目的を果たす場合がある。前者が H 駆動で後者が L 駆動である。例えば，5V 出力信号で発光ダイオードをつけるか，0V 信号で発光ダイオードをつけるかの違いである。

図の回路で具体的に考える。電圧 A, B が入力で，電圧 Y が出力である。図の様に $A = 5\text{V}$, $B = 0\text{V}$ とすると，ダイオードを通って電流が流れ，$Y = 0\text{V}$ となる(ダイオードの電圧を無視する)。$A = 5\text{V}$, $B = 5\text{V}$ とすると，電流は流れず，抵抗の電圧は 0 だから $Y = 5\text{V}$ となる。従って，H = 5V, L = 0V で書いた真理値表は図のようになる。これは，実際の電圧の表である。

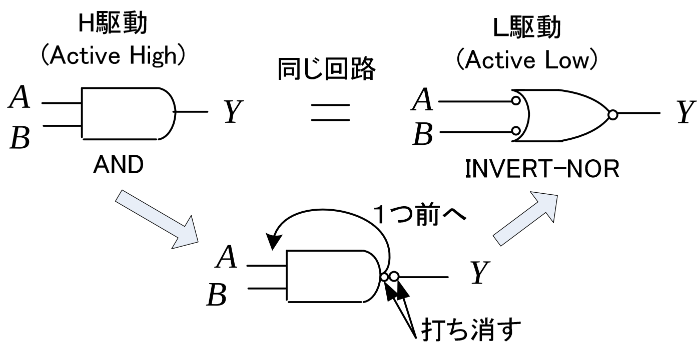


H 駆動時 AND 回路　　A=H, B=L の場合　　A=H, B=H の場合

A	B	Y
L	L	L
H	L	L
L	H	L
H	H	H

真理値表

L = 0V
H =5V



H 駆動(Active High)と L 駆動(Active low)の論理回路の書き方

（どちらで書いても間違いではないが目的で分けた方が設計者の意図が明確）

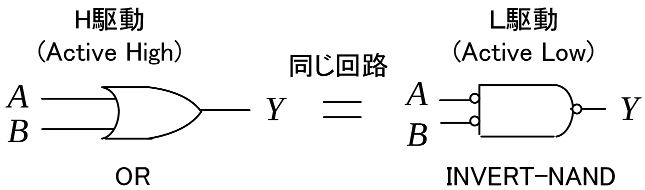
H 駆動とは $Y=\mathrm{H}$ のとき次段の素子を動作させるので，入力 A,B のどちらも H にならないと出力は出ないという見方である。よってこの回路は AND と考えられる。一方，**L 駆動**とは，出力が L になったら次段の素子を動作させるので，入力のどちらかが L なら動作するので，L に注目すると真理値表よりこの回路は OR と考えられる。そこで，設計者がどちらを考えているか記号を分けた方が判りやすい。AND の出力に反転の○を２つつけても等価で，そのうちの 1 つを入力側に移すとド・モルガンの定理より AND が OR の記号となり，INVERT-NOR の記号が得られる。両者の回路は全く同じで，論理記号にも矛盾はない。しかし書き方を変えることで，H 駆動と L 駆動のどちらで考えているかが判りやすく，L 駆動の場合 OR のイメージが得られる。

同様に H 駆動 OR の場合を考える。真理値表の L に着目した L 駆動の場合には，入力 A, B のどちらも L にならないと出力は L にならず駆動しないので AND と考えられる。よって L 駆動の場合には図の INVERT-NAND で書いた方が判りやすい。この L 駆動の論理回路は後述のマイコンの接続で利用する。

A	B	Y
L	L	L
H	L	H
L	H	H
H	H	H

真理値表

L = 0V
H =5V



○ 2 進数，16 進数

通常我々は 10 進数を用いているが，コンピュータでは 2 進数や 16 進数が用いられている。10 進数の場合には，0 から 9 の数字を使うが，2 進数の場合には 0 と 1 しか使えない。16 進数では，0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F の 16 種の記号を使う。A は 10 進数の 10 に相当し，順に対応して F が 15 になる。10 進数の足し算で 10 以上になると桁上がりとなるが，2 進数では 2 以上，16 進数では 16 以上になると桁上がりする。2 進数と 16 進数は，よく対応しており，2 進数の 4 桁をまとめて 16 進数の 1 つの数値になる。

2 進数を 10 進数に直す場合は，各桁(けた)の重みを 2^n（2 桁目が $n=1$）として 2 進数の数値に掛けて加えるとよい。10 進数でも各桁は10^n の重みを持っている。

10000010

$2^7=128$　$2^1=2$ ⟶ $128+2=130$

10 進数を 2 進数に直すには，2 で割り算を行って，余りを右側に書いて下から順に並べると良い。12（10 進数）＝1100（2 進数）

	余り
2) 12	
2) 6	0
2) 3	0
1	1

表 7-1 8ビットデータの表現

2 進数 (Binary)	16 進数 (Hexa Decimal)	符号無し表現とみた場合の 10 進数の値 (Decimal)	符号付き表現とみた場合の 10 進数の値 (Decimal)
0000 0000	00	0	0
0000 0001	01	1	1
0000 0010	02	2	2
・	・	・	・
・	・	・	・
・	・	・	・
0111 1111	7F	127	127
1000 0000	80	128	-128
1000 0001	81	129	-127
1000 0010	82	130	-126
・	・	・	・
・	・	・	・
・	・	・	・
1111 1110	FE	254	-2
1111 1111	FF	255	-1

16 進数の場合，数値の後に H を付ける。例 10H（10 進数では 16）

2 進数の 1 桁を 1 **ビット**(bit)という。表は 8 桁なので，8 ビットである。8 ビットで表現できる数値は 0 から 255 の 256（ $2^8=256$ ）種類である。なお 8 ビットで負の数を表現しようとすると，表のように最上位ビットが 1 の場合を割り当てる。2 の補数表現と呼ばれるが，深入りはしないでおく。

○ 量子化

コンピュータで信号を処理するためには，A/D 変換器によって，センサから得られるアナログ量をディジタル量に変換する必要がある。A/D 変換器によるデータの取り込みに関しては，**量子化**の問題がある。

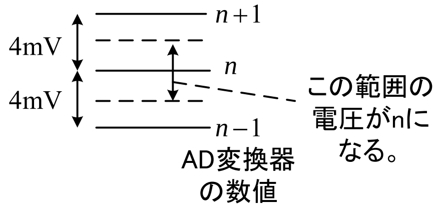
A/D 変換器は 8bit や 12bit などの限られた数値に変換される。例えば，8 ビットの A/D 変換器なら，センサで検出するデータを 0 から 255（または-128 から 127）の 256 種類に割り当てる必要がある。つまりある範囲をまとめて 1 つの数値に直さないといけない。これを量子化という。次の問題を考えよう。

問　A/D 変換器によって 0～1V 信号電圧を 8 ビットのディジタル信号に量子化する。量子化誤差の最大値に最も近いのはどれか。

(1) ±0.2mV　(2) ±2mV　(3) ±20mV　(4) ±200mV　(5) ±2V

(解) 8 ビットの分解能は $2^8=256$ (0 から 255 まで), $1/256=0.0039$ より, 約 4mV である。つまり，4mV の幅でどれかの数値に割り当てる必要がある。

量子化するときに四捨五入するなら図の範囲の電圧が同じ数値 n (=0～255)になる。よって図より正解は(2)



A/D 変換の量子化

このように，ディジタル信号はアナログ信号より精度が劣化するが，一旦ディジタル量になると雑音の影響はほとんどなくなる。

○ コンピュータのソフトウェア

ソフトウェアは**基本ソフト**と**応用ソフト（アプリケーションソフト）**に分類される。基本ソフトには，**オペレーティングシステム** OS，**プログラミング言語**，ユーティリティープログラム，オブジェクトプログラムなどがある。応用ソフトは，ワープロソフト，エディタなど沢山ある。

オペレーティングシステム（OS）は, 応用ソフトが動くための土台と成るソフトで，ハードウェアに直接関係するソフトである。MS-DOS, Linux, UNIX, Android, Windows7 などがある。これらの OS 上でソフトウェアを開発すれば, メモリや入出力の番地は考えなくても良くなり, 開発がし易くなる。

プログラミング言語は，ソフトウェアを開発するためのものである。コンピュータを動かすには, 最終的には 1 と 0 の組み合わせとして, メモリに入れないといけない。しかし，これは人間にとって大変わかりにくい。そこで人間がわかり易い言葉すなわち言語が作られた（これを**高級プログラミング言語**という）。1 と 0 の組み合わせはコンピュータにとって理解できる唯一の言語で，これを**機械語**という。汎用言語の C, C++，事務処理用言語 COBOL, 科学技術計算用言語 FORTRAN などは高級プログラミング言語である。これらの言語で書かれたプログラムは, 実行する前に**コンパイラ**（翻訳するプログラム）によって**コンパイル**して機械語に直す必要がある。BASIC は高級プログラミング言語であるが, 翻訳と実行を**インタプリタ**で行う。高級プログラミング言語に対して，**低級プログラミング言語**には先ほど述べた機械語の他に**アセンブリ言語**がある。ニーモニックコードで書くことで機械語に比べると人間に判りやすいが, CPU によって命令が異なりプログラム開発は大変である。アセンブリ言語を機械語に変換するソフトを**アセンブラ**という。

7.1　CPU

マイコンは**マイクロコンピュータ**の略称で，**CPU**(central processing unit)，メモリ，入出力装置から成る。本節では**8ビット**の CPU である **Z80**（**ぜっとはちまる**）について述べる。Z80 は一世を風靡（ふうび）した CPU であるが現在でも使われており，簡単で理解が容易なことからマイコンの原理を知るためには適している。

図 7-1 に Z80 の構成を示す（文献(4)）。これは他の **LSI**(Large Scale Integration：大規模集積回路)についても言えることであるが，Z80 は 5V 単一電源で動作し，**TTL**(transistor-transistor logic)コンパチブルである。図でⓉと記入されている端子は，“H”レベル(5V)と“L”レベル(0V)の他に線が切れた“高インピーダンス”状態を持つもので**トライステート**と呼ばれる。ピンの矢印は信号の伝わる向きを示す。制御線の○印やバー￣はL駆動（Active Low）を表すもので，“L”レベルのときにそのピンのもつ働きが有効となる。例えば，$\overline{\mathrm{RD}}$ は読み込み動作をしようとするとき“L”レベルとなり，$\overline{\mathrm{INT}}$ は“L”レベルになると割り込みを CPU に要求する。図で A，B，C，D，E，H，L，I の各**レジスタ**は 8 ビット， IX，IY，SP，PC の各レジスタは 16 ビットである。レジスタは CPU 内のメモリと思えばよい。表 7-2 に Z80 のピンの主な機能を示す。

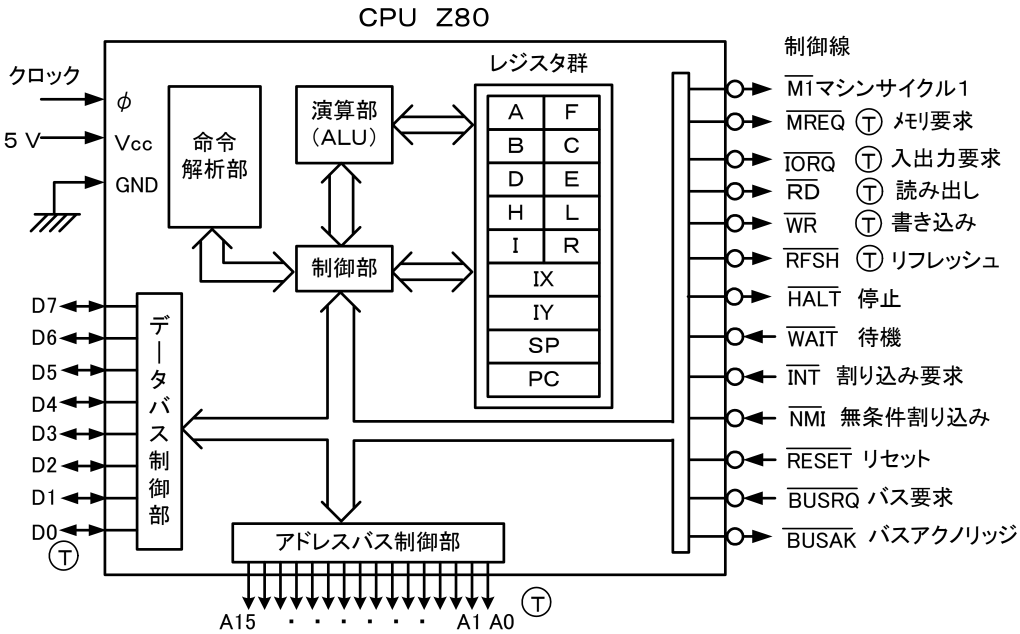


図 7-1　Z80 の構成

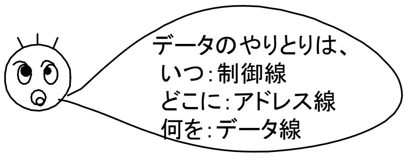


表 7-2　Z80 のピンの主な機能

ピ　ン	名　　前	機　　能
ϕ	Clock	このクロックパルスを基準に CPU が動作する。 Z80 は 2.5MHz 以下，Z80 A は 4MHz 以下
A0～A15	Address bus	メモリや**入出力装置**（IO）の番地指定に用いる線である。
D0～D7	Data bus	メモリや IO と命令やデータをやりとりする線である。
$\overline{\mathrm{MREQ}}$ $\overline{\mathrm{IORQ}}$	Memory request IO request	CPU がメモリ(LD 命令)または IO (IN, OUT 命令)のどちらとつながるかを決める。
$\overline{\mathrm{RD}}$ $\overline{\mathrm{WR}}$	Read Write	CPU がメモリや IO からデータを読み込むのか，逆に書き込むのかを決める。
$\overline{\mathrm{INT}}$	Interrupt request	CPU に割り込みを要求する。
$\overline{\mathrm{RESET}}$	Reset	CPU を初期状態にする。

7.2　CPU とメモリ

(a) CPU とメモリの接続

半導体メモリは, CPU から直接読み出しと書き込みの両方が可能な **RAM**（Random Access Memory）と読み出し専用の **ROM**（Read Only Memory）に大別できる。RAM は電源を切るとデータが消えてしまうのが普通で，電源を入れたままでもデータが消える**ダイナミック RAM** もある。ダイナミック RAM でない RAM を**スタティック RAM** という。ROM には，素子の製造中にデータを固定してしまい，後で変更できないマスク ROM，紫外線を照射することにより何度もデータの書き込みが可能な EPROM などがある。ここでは仕組みを理解する目的でメモリ容量 8K バイトの EPROM と 8K バイトのスタティック RAM を用いる（容量は小さいが原理を理解することが目的である）。

図 7-2 に Z80 とメモリの接続例を示す。**アドレス線** A0～A15（16 本）で指定できる番地の数は，$2^{16}=2^6×2^{10}$＝64×1024=65536=64K である。コンピュータの分野では K（キロ）が 1024 を意味することがある。メモリの各番地は 8 ビット構成であり，**1バイト＝8ビット**であるから，図の 8K バイト(64K ビット)メモリの番地の数は 8K である。8K の番地を指定するアドレス線の数は，$2^{13}=2^3×2^{10}$=8K であるから A0～A12 までの 13 本で良く，残り A13～A15 は**アドレスレコーダ** 74LS138 に接続してメモリ IC の選択に用いる（メモリを何個もつなぐ場合で，最近はこのようなケースは少ないだろう）。74LS138 を図のように接続すると，表 7-3 に示すように，A13～A15 に対し，Y0～Y7 のいずれかの端子が“L”となる（他は“H”）。

ROM は，$\overline{\mathrm{CE}}$ (**chip enable**)が“L”，$\overline{\mathrm{OE}}$ **(output enable)**が“L”ならばアドレス信号（A0～A12）の番地のデータを，データバス（D0～D7）に出力する。CPU はそれを読み込むことになる。**RAM** は，$\overline{\mathrm{CE}}$ が“L”，$\overline{\mathrm{WR}}$ (RAM)が“L”ならば，$\overline{\mathrm{OE}}$ の状態に無関係に，アドレス信号の示す番地に，データバスの内容を記憶する。データの読み出しは，$\overline{\mathrm{CE}}$ が“L”，$\overline{\mathrm{OE}}$ が“L”，$\overline{\mathrm{WR}}$ (RAM)を“H”とする。アドレスデコーダの出力と $\overline{\mathrm{CE}}$ の接続から，図の回路の**メモリマップ**は表 7-4 のようになる。A0～A12 までの 13 本(13 ビット)あることに注意せよ。

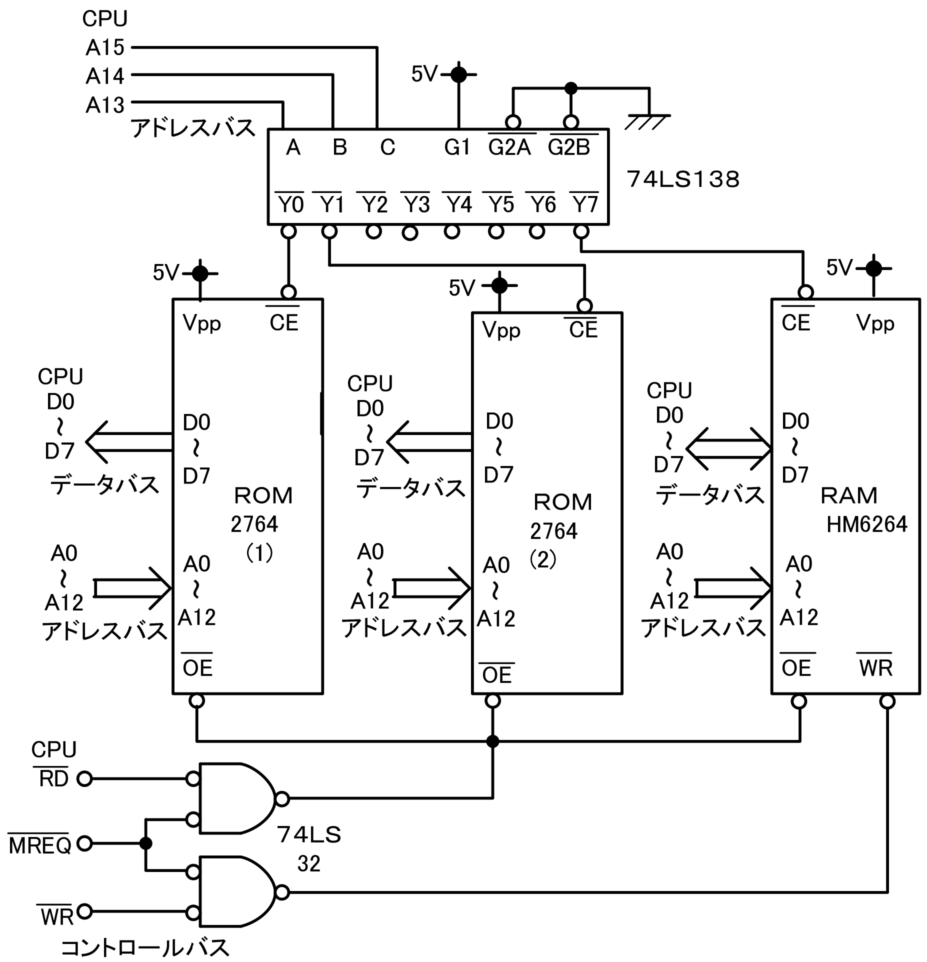


図 7-2　Z80 とメモリの接続

表 7-3　74LS138 の入出力表

A15	A14	A13	“L” となるピン
0	0	0	$\overline{Y0}$
0	0	1	$\overline{Y1}$
0	1	0	$\overline{Y2}$
0	1	1	$\overline{Y3}$
1	0	0	$\overline{Y4}$
1	0	1	$\overline{Y5}$
1	1	0	$\overline{Y6}$
1	1	1	$\overline{Y7}$

G1＝“H”, $\overline{G2A}$＝$\overline{G2B}$＝“L” のとき

表 7-4　メモリマップ

番　地	メ　モ　リ
０００００Ｈ ： １ＦＦＦＨ	ROM(1)
２０００Ｈ ： ３ＦＦＦＨ	ROM(2)
：	空　き
Ｅ０００Ｈ ： ＦＦＦＦＨ	RAM

(b)プログラムの実行

CPU の基本的な動作は，メモリに保存されている命令を次々に読み出して実行することにある。ここでは簡単なプログラムを CPU が実行してゆく過程を説明する。図 7-3 にプログラムの実行を説明するためのハードウェアの構成とメモリの内容の一例を示す。CPU とメモリの接続は図 7-2 と同じものとする。

1．リセットスイッチを押して，$\overline{\mathrm{RESET}}$ が“L”になると，CPU は CPU 内の**プログラムカウンタ(PC)**を **0** にする。

2．リセットスイッチを切って，$\overline{\mathrm{RESET}}$ が“H”になると，PC が示す 0 番地の命令を**フェッチ**（fetch：行って取って来る）し，解析する。（**フェッチサイクル**＝4 クロック）

① CPU は PC の値をアドレスバスに出力する。A0＝A1＝・・・＝A15＝0 なので，ROM(1)の 0 番地が指定される。その後 **CPU は PC に1を加え，PC＝1とする。**

② CPU の $\overline{\mathrm{MREQ}}$，$\overline{\mathrm{RD}}$ が共に“L”となる。その結果，ROM(1)の $\overline{\mathrm{OE}}$ が“L”となり（L 駆動の考え方が役立つ），ROM はデータバス上に 0 番地の値 3AH を出力する。

③ CPU はデータバス上の 3AH を読み込み，この命令を解析する。この結果，CPU は次の番地（1 番地）のデータを下位番地，更に次の番地（2 番地）のデータを上位番地としたメモリからデーを読み込み，A レジスタに入れる命令であることを知る。（表 7-5 参照)。

3．読み出したいデータが入っている番地をメモリから読み込む。
メモリリードサイクル（3 クロック）×2＝6 クロック

① CPU は PC の値をアドレスバスに出力する。ROM（1）の 1 番地が指定される。その後，**PC＝2**となる。

② ２の②と同様にして，ROM（1）はデータバス上に 1 番地のデータ 00H を出力する。

③ CPU はデータバス上の 00H を読み込む。

④ ①～③と同様にして 2 番地のデータ 20H を読み込む。**PC＝3**となる。

4．CPU は読み出したデータの入っている番地 2000H が分ったので，ROM（2）からこれを読み込む。この結果 10H が，レジスタ A に入る。

5．CPU は PC の値をアドレスバス上に出力し，次の命令のフェッチサイクルが始まる。
２と同様にして，CPU はメモリより 32H を読み込み，それを解析して，A レジスタのデータを次の番地（4 番地）のデータを下位番地，更に次の番地（5 番地）のデータを上位番地としてメモリへ書き込む命令であることを知る。**PC＝4**となる。

6．CPU はデータを書き込む番地をメモリから読み出す。３と同様にして，4 番地の 00H, 5 番地の E0H を読み込む。**PC＝6**となる。

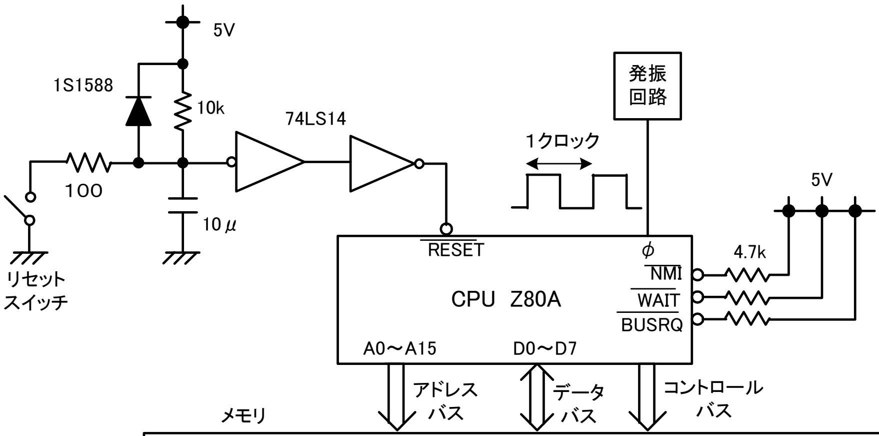
7．CPU は E000H 番地に A レジスタの値（10H）を書き込む。
（**メモリライトサイクル**＝3 クロック）

① CPU はアドレスバスに E000H を，データバスに A レジスタの値を出力する。RAM の E000H 番地が選定される。

②　CPU の $\overline{\mathrm{MREQ}}$，$\overline{\mathrm{WR}}$ が共に“L”となり，その結果 RAM の $\overline{\mathrm{WR}}$ が“L”となる。

③　RAM はデータバスの値を E000H 番地へ保存する。

以下，同様にして次々と命令が実行されることになる。正確にはメーカが公表したタイミング図により各信号の変化が定まっているが，プログラムの基本的な実行過程は理解できたであろう。簡単に言うと，CPU にある **PC の値を自動的に1つずつ増やして，それが示す番地の命令を読み出し実行しているのである。**上述の例でも判るように，命令の実行はいくつかのサイクルから構成されている。それには，①フェッチサイクル　②メモリリードサイクル　③メモリライトサイクル　④IO リードサイクル　⑤IO ライトサイクル　⑥割り込み応答サイクルなどがある。フェッチサイクルはどの命令にも必要である。



メモリ

番地（１６ビット）	データ（８ビット）	メモリ
０H	３AH	
１H	００H	
２H	２０H	
３H	３２H	ROM
４H	００H	（１）
５H	E０H	
:	:	
１FFFH	:	
２０００H	１０H	
２００１H	FFH	ROM
:	:	（２）
３FFFH	:	
		空　き
E０００H	:	
:	:	RAM
FFFFH	:	

図 7-3　プログラムの実行（データは一例）

7.3　アセンブリ言語

16 進数で表現される**機械語**は人間には理解しにくいので，ニーモニックという機械語に対応した英記号を使って作った**アセンブリ言語**を用いる。**アセンブリ言語**で書かれたプログラムを機械語に自動的に置き換えるプログラムは**アセンブラ**と呼ばれる。表 7-5 に Z80 の命令を示す。

フラグが変化する**加算命令**（ADD）と**減算命令**（SUB）について説明しておく。表 7-1 は 8 ビットデータの表現法とそれを**符号無し表現**とみた場合，または**符号付表現**（2 の補数表現）とみた場合の 10 進数での値を示している。表 7-1 で，8 ビットデータを符号無し表現とみるか，それとも符号付表現とみるかはプログラマー自身が決めることで，加算命令や減算命令で両者が区別して実行される訳ではない。ただし，CPU 内にある **F レジスタの値（フラグ）**が有用な情報を提供してくれる。例えば，正の数だけを対象とした符号無し表現としてデータを扱う場合，加算，減算の結果，桁上り（255 を超えるとき），桁下り（負の数になるとき）があると，**キャリフラグ**（C フラグ）が 1 になり，桁上り，桁下りが無いときは 0 となる。符号付表現としてデータを扱う場合，加算，減算の結果-128～+127 以外の数を表現する必要が生じたとき**オーバフローフラグ**（V フラグ）が 1 となり，範囲内であれば 0 となる。従ってフラグの変化を調べながら演算を行う必要がある。

表 7-5　Z80 の命令

ニーモニック	機械語 （16 進数）	ｸﾛｯｸ ｻｲｸﾙ	機能説明
LD　A, n	3E　n	7	8 ﾋﾞｯﾄ定数 n を A ﾚｼﾞｽﾀに入れる。 A←n
LD　A, B	78	4	B ﾚｼﾞｽﾀの値を A ﾚｼﾞｽﾀへ入れる。 B は不変　　A←B
LD　A, (lm)	3A　m　l	13	lm 番地のﾃﾞｰﾀを A ﾚｼﾞｽﾀへ入れる。 A←(lm)
LD　(lm), A	32　m　l	13	(lm)←A
ADD　A, B	80	4	A←A+B　ﾌﾗｸﾞ変化あり
SUB　B	90	4	A←A-B　ﾌﾗｸﾞ変化あり
IN　A, (n)	DB　n	11	n 番地(8 ﾋﾞｯﾄ)の IO ﾎﾟｰﾄから A ﾚｼﾞｽﾀへ読み込む。
OUT　(n), (A)	D3　n	11	A ﾚｼﾞｽﾀの内容を n 番地の IO ﾎﾟｰﾄへ出力する。
PUSH　BC	C5	11	BC ﾚｼﾞｽﾀの値をｽﾀｯｶへ転送する。 (SP-1)←B(上位) (SP-2)←C(下位) SP←SP-2
POP　BC	C1	10	ｽﾀｯｶの値を BC ﾚｼﾞｽﾀへ転送する。 C(下位)←(SP) B(上位)←(SP+1) SP←SP+2

JP　lm	C3　m　l	10	lm 番地へｼﾞｬﾝﾌﾟする。 PC←lm
CALL lm	CD　m　l	17	ｻﾌﾞﾙｰﾁﾝｺｰﾙ PC をｽﾀｯｶへ PUSH し PC←lm
RET	C9	10	ｻﾌﾞﾙｰﾁﾝからのﾘﾀｰﾝ PC へｽﾀｯｶより POP
EI	FB	4	割り込み許可
DI	F3	4	割り込み禁止

LD 命令では $\overline{\mathrm{MREQ}}$ が“L”，IN,OUT 命令では $\overline{\mathrm{IORQ}}$ が“L”になる。

最近の 32 ビットなどの CPU では，アセンブリ言語のかわりに **C 言語**でプログラムを書くので便利である。また変数が浮動小数点演算できるのでフラグも気にしないですむ。

7.4　マイコンの構成

CPU を何かの目的に使うとき，必ずメモリ以外からのデータの入出力を必要とする。

（a）汎用入出力 LSI　8255

例えば，8 ビットデータにより 8 個の発光ダイオードを点滅する場合を考えてみよう。この 8 ビットのデータはデータバスから出力することになるが，データバスは CPU とメモリがデータをやりとりする場合にも使われるので，データバスに直接発光ダイオードをつなぐとどちらのデータか区別できず誤動作することになる。また，一度出力したデータは次のデータが来るまで保持する（**ラッチ**する）ことが望ましい。更に，発光ダイオードだけではなく他の周辺装置とデータのやりとりをしたい場合もあり，データバスをうまく使い分ける必要がある。このような場合，3 つの 8 ビット入出力ポートをもつ 8255 が利用できる。図 7-4 に Z80 と 8255 の接続例を示す。図において，デコーダ 74LS138 で 8255 をはじめ後述する各周辺 LSI を選択する。図より，アドレスバスの下位 8 ビットが E0H～E3H のとき $\overline{\mathrm{CS}}$ が“L”となり，8255 が動作する。CPU と 8255 の A, B, C ポート及び **CW（コントロールワード）レジスタ**のいずれとを接続するかは，A0 と A1 の信号で決る。また，データの入出力の区別は 8255 の $\overline{\mathrm{RD}}$， $\overline{\mathrm{WR}}$ 信号で行う。表 7-6 に 8255 の基本機能を示す。

8255 には，モード 0，モード 1，モード 2 の使い方があるが，単なる IO ポートとして使う場合にはモード 0 で使用する。図 7-5 にモード 0 での CW レジスタへの書き込み形式を示す。モード 0 において，出力ポートとして使うときデータは保持されるが，入力ポートとして使うときは保持されない。図 7-4 の回路で，ポート B のスイッチがオンのとき，各ビットに対するポート A の発光ダイオードを点燈するプログラムを示す。

① LD A, 10000010B ; モード 0 で A，C ポート出力。B ポート入力。

② OUT (0E3H), A ; CW レジスタへの書き込み

③ IN A, (0E1H) ; B ポートからデータの読み込み

④ CPL ; A レジスタのビット反転（0→1，1→0）

⑤ OUT (0E0H), A ; A ポートへ出力（発光ダイオード点燈）

②の OUT （0E3H)，A （機械語 D3 E3）の実行過程を以下に説明する。

１．フェッチサイクル（4 クロック）

CPU はメモリより D3H を読み込み，命令を解析する。

２．メモリリードサイクル（3 クロック）

CPU はメモリより E3H を読み込む。

３．IO ライトサイクル（4 クロック）

① CPU はアドレスバスの下位 8 ビットに E3H を出力する。これにより, 8255 が動作する。

② CPU はデータバスに A レジスタの内容を出力する。

③ CPU の $\overline{\mathrm{IORQ}}$， $\overline{\mathrm{WR}}$ が共に“L”となり，その結果 8255 の $\overline{\mathrm{WR}}$ が“L”となり，8255 はデータバスの内容を CW レジスタへ取り込む。

（注意）IN または OUT 命令では， $\overline{\mathrm{IORQ}}$ 端子が“L”になるので，E3H,E0H 番地は同じ番地がメモリにあっても構わない。メモリとは LD 命令を使い， $\overline{\mathrm{MREQ}}$ が“L”になる。

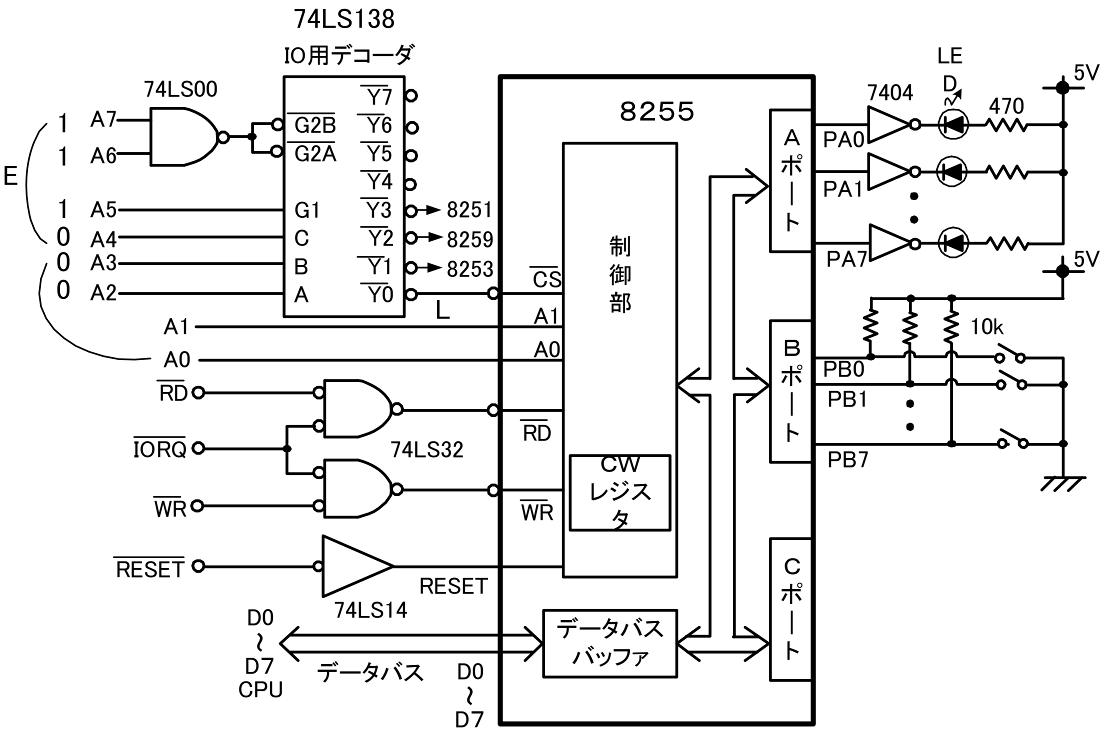


図 7-4 Z80 と 8255 の接続

表 7-6　8255 の基本機能

$\overline{\text{CS}}$	A１	A０	番　地	$\overline{\text{RD}}$	$\overline{\text{WR}}$	機能（データの流れ）
L	0	0	E０H	L	H	CPU←ポートA
L	0	1	E１H	L	H	CPU←ポートB
L	1	0	E２H	L	H	CPU←ポートC
L	0	0	E０H	H	L	CPU→ポートA
L	0	1	E１H	H	L	CPU→ポートB
L	1	0	E２H	H	L	CPU→ポートC
L	1	1	E３H	H	L	CPU→CW レジスタ
H	X	X	E0H～E3H 以外	X	X	ﾃﾞｰﾀﾊﾞｽは高ｲﾝﾋﾟｰﾀﾞﾝｽ状態

Xは“H”または“L”

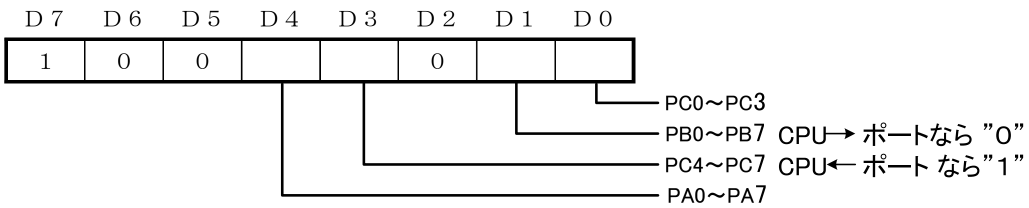


図 7-5　モード 0 における CW レジスタへの書き込み形式

（b）割り込み制御用 LSI　8259

割り込みの話をする前に，まず**サブルーチン**（または関数）について述べる。同じ手順のプログラム（データは異なっていてもよい）を複数回用いるときには，そのプログラムをサブルーチンにして，必要に応じて呼び出すことにすれば，全体のプログラムが短くなり，また，情報の流れが見やすくなる。図 7-6 (a)と(b)は全く等価なプログラムであるが，(b)の方が処理 A をサブルーチンにした分だけ短くなっている。(b)図のプログラム実行過程を説明する。

①　処理 1 を実行する。

②　CALL　SUB を実行する。（表 7-5 参照）

処理 2 の先頭番地が自動的に**プログラムカウンタ（PC）**に入るが，これを**スタックポインタ（SP）**の示す番地のメモリへしまう。これは，PUSH　PC に相当する命令（実際には無い）を実行したことになる。次に，ラベル名 SUB の所すなわち処理 A の先頭番地へジャンプする。PC と SP は CPU の中のレジスタである（図 7-1 見よ）。

③　処理 A を実行する。

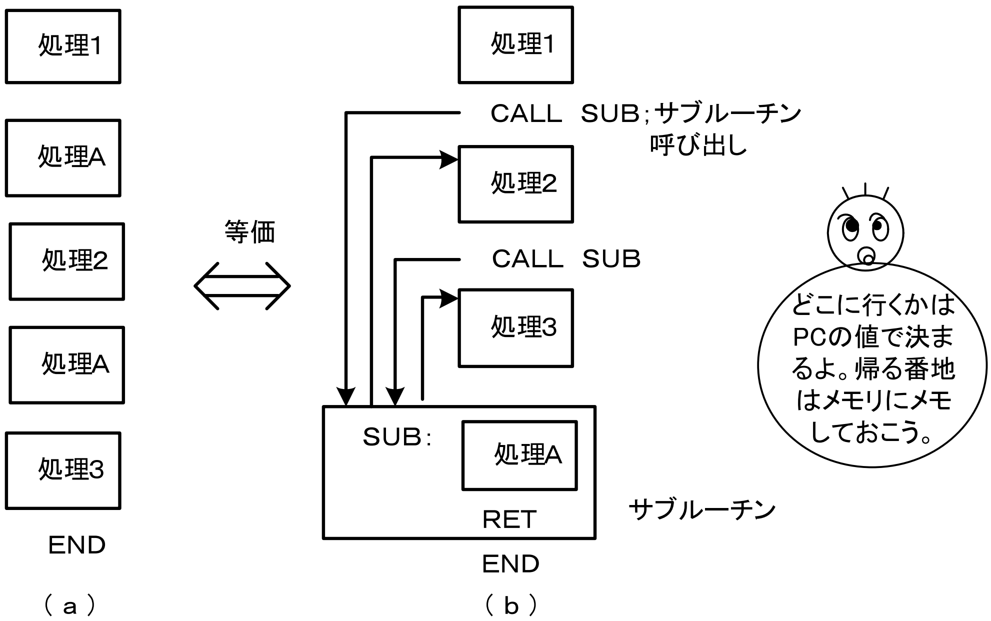


図 7-6　サブルーチン

④　RET 命令を実行する。（表 7-5 参照）

SP の示す番地のデータを PC にしまう。これは POP　PC に相当する命令（実際には無い）が実行されることになる。この結果，処理 2 の先頭番地へジャンプする。

⑤　処理 2 を実行する。

⑥　CALL　SUB を実行する。

⑦　処理 A を実行する。

⑧　RET を実行する。

⑨　処理 3 を実行する。

以上のことから，(a) のプログラムと (b) のプログラムは等価であることがわかる。スタックポインタ（SP）が番地を管理しているメモリ（RAM）の領域は**スタッカ**と呼ばれる。サブルーチンコールでは戻り番地を自動的にスタッカへ退避し，RET 命令でこれを PC に入れて元のプログラムへ復帰している。PUSH，POP の動作の詳細は，表 7-5 を参照されたい。PUSH，POP 命令やサブルーチンを用いる場合には必ずプログラムの最初で SP のイニシャライズ（初期設定）を行う必要がある。

次に割り込みについて述べる。サブルーチンはプログラム実行中に CALL 命令があると実行されるが，外部からのハード的な信号（割り込み信号）によってもサブルーチン（**割り込み処理プログラム**）を実行することができる。これを**割り込み**（Interrupt）という。CPU が同時に実行できる命令は 1 つであるが，割り込みを使うことで複数の異なる処理を時分割で実現できる。ここでは割り込み制御用 LSI として 8259 を取り上げる。

図 7-7 に Z80 と 8259 の接続例を示す。マイコンの外部で作られた割り込み要求の信号が 8259 の IR0 ピンに入力されるとする。8259 にはコントロールワード(CW)レジスタがあり，CPU に実行して欲しい割り込み処理プログラムが置かれているメモリの先頭番地を予め設定しておく。この初期設定が終わった後で，EI 命令をどこかで実行して，Z80 が割り込み受け付け可能な状態にする。EI 命令を実行していないと CPU は割り込みを受け付けない。

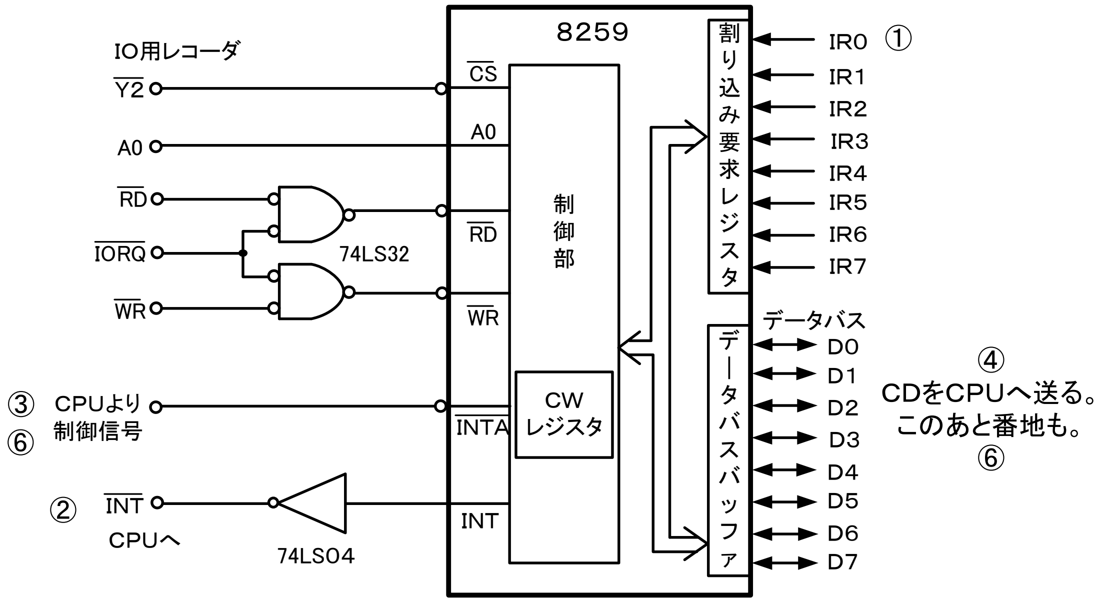


図 7-7　Z80 と 8259 の接続

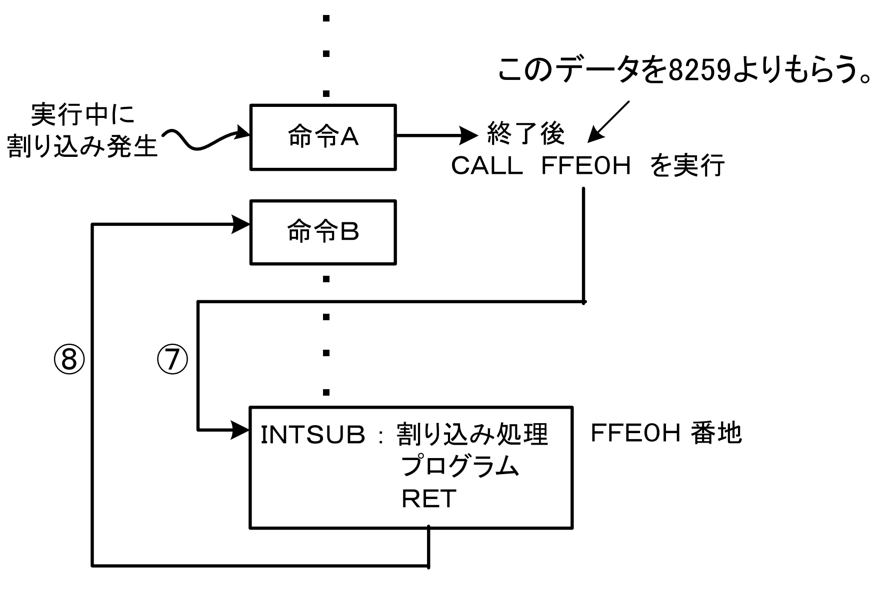


図 7-8　割り込み発生時に CPU が実行するプログラムの順序

図 7-8 のプログラムを用いて割り込みシーケンス説明する。初期設定で IR0 に対しては，FFE0H 番地が 8259 の CW の中に書いてあるとする。①～⑧は図中の番号に対応する。

① 命令 A を実行中に 8259 の IR0 端子が外部回路により“L”から“H”へ変化する。
② 8259 は CPU に $\overline{\text{INT}}$ 信号を出力する。“割り込みが来たよ”と知らせる。
③ EI 命令後とすると CPU は割り込みを受け付け，命令 A の実行の後 $\overline{\text{M1}}$ と $\overline{\text{IORQ}}$ を共に“L”にする。これらの信号によって，8259 の $\overline{\text{INTA}}$ に“L”信号が入力される（そのような論理回路が必要）。
④ 8259 はこの $\overline{\text{INTA}}$ 信号を受けて, データバスに CALL 命令(マシン語 CDH)を出力する。
⑤ CPU は CDH を読み込むと，CALL 命令と判るので，次いでジャンプ先の番地を読み込むためのメモリリードサイクルとなる。
⑥ CPU の $\overline{\text{RD}}$ が“L”となるので， $\overline{\text{INTA}}$ が再び“L”となる（そのような論理回路が必要）。この結果，8259 はジャンプ先番地の下位 E0H をデータバスに出力し，これを CPU が読み込む。同様にして，CPU はジャンプ先番地の上位 FFH を読み込む。これで 3 バイトの CALL 命令が読み終わった。
⑦ CPU は CALL　FFE0H を実行する。すなわち，命令 B の格納されている番地をスタッカへしまい，FFE0H へジャンプする。この結果 FFE0H から始まる割り込み処理プログラムが実行される。
⑧ 割り込み処理プログラムの最後の RET 命令で，スタッカを POP して中断していた処理プログラム（命令 B）へ戻り実行する。

以上のように<u>割り込みはハード的に引き起こされたサブルーチンコール</u>である。8259 は CPU へ割り込み処理プログラムが置かれているジャンプ先を知らせる役目がある。また，IR1～IR7 には別の割り込み処理プログラムの先頭番地を割り当てておけば，いろいろな処理が可能となる。同時に割り込みが入ることもあるので，8259 で優先順位を決めておくことができる。

7.5　割り込みを利用したマイコン制御システム

割り込みを利用したディジタル制御システムの例を図 7-9 に示す。*RL* 回路の電流を制御する最も簡単なマイコン制御システムである。第 1 章では，電源電圧は自由に変えられることを前提としたが，実際にはトランジスタを電子スイッチとして使ってゲート信号でオン，オフし，周期 T ごとの平均値を変える方法が用いられる（文献(20)）。つまり，トランジスタ Q がオンしたら $v = E$, Q がオフしたらダイオード D を通って電流が流れ $v = 0$ となる。ダイオード D がないと Q をオフした瞬間にコイルのスパイク電圧でトランジスタが破壊される。以上により，周期 T ごとの v の平均値は図 7-10 より

$$\overline{v} = \frac{T_{on}}{T} E \tag{7-1}$$

となる（ダイオードは理想的と仮定）。T_{on} をマイコンで計算し，ゲート信号発生器でその長さのパルスを作り，トランジスタのゲートに加えれば T_{on} 期間 Q をオンできる。

割り込みの利用について述べる。ゲート信号は周期 T ごとにオン電圧とオフ電圧としてトランジスタに加える（電圧は低いが波形は v と同じ）。このオン期間の長さで平均電圧 $\overline{v}$ が決るので，周期 T の間制御の計算は 1 回でよい。何故なら数回計算してもオン期間は周期 T ごとに 1 回しか変えられないからである。よって検出する電流も周期 T ごとに 1 回としよう。周期 T はゲート信号発生器で決まるので，T ごとにマイコンに割り込みをかける。図 7-11 に CPU が実行する制御プログラムの構成を示す。**メインプログラム**は起動時に実行され，初期値の設定（一度行うだけでよい）を行ったあと接続するパソコンとの通信を行う無限ループに入る。これは装置の電源がオフされるまで実行され続ける。**割り込み処理プログラム**はゲート信号発生器からのパルスで周期 T ごとに実行される。**割り込み処理プログラム**の中に A/D 変換器よる電流検出，電流制御演算（例 PI 制御），ゲート信号発生器への T_{on} 出力などを書いておく。CPU は同時に 2 つのプログラムを実行することはできず，割り込み処理プログラムが動いていない間だけメインプログラムが動く。

サンプリング周期 T は 100μs 程度に短くできるから，連続して v が変わる（連続系）と考えることもある。v がサンプリング周期 T ごとに階段状に変化すると仮定すれば零次ホールドがある場合のディジタル制御系として解析できる。

最近のマイコンは，図の点線で囲んだものに更に A/D 変換器までを内蔵したシングルチップになっているものも多い。安価でコンパクトさらに高速大容量へと進化している。本稿で述べたことはそのような場合でも基本的に変わることはない。

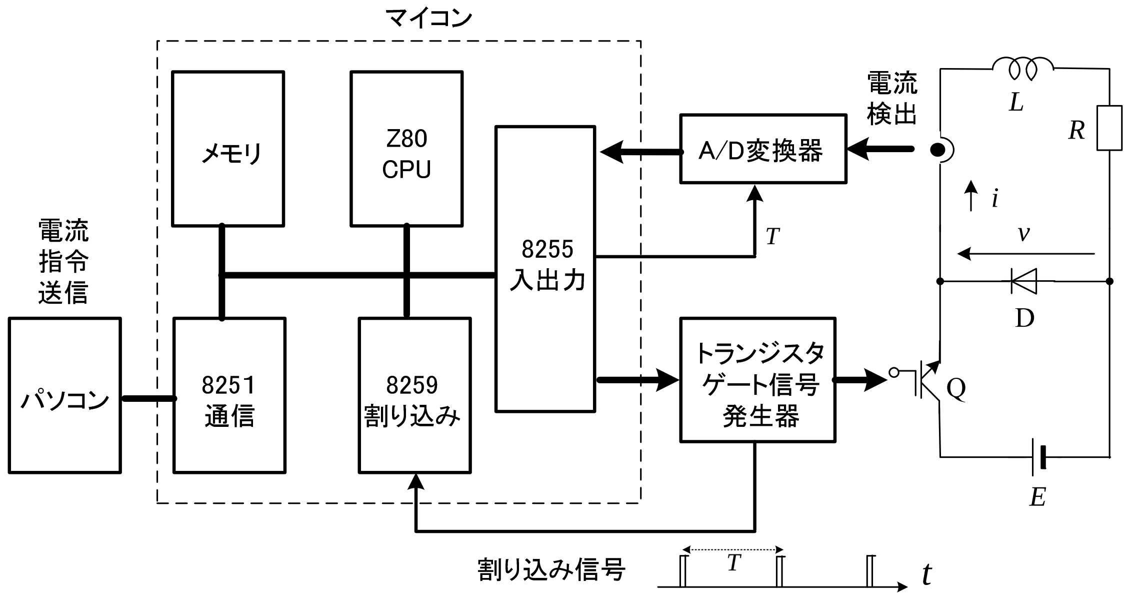


図 7-9　割り込みを利用したディジタル制御システム

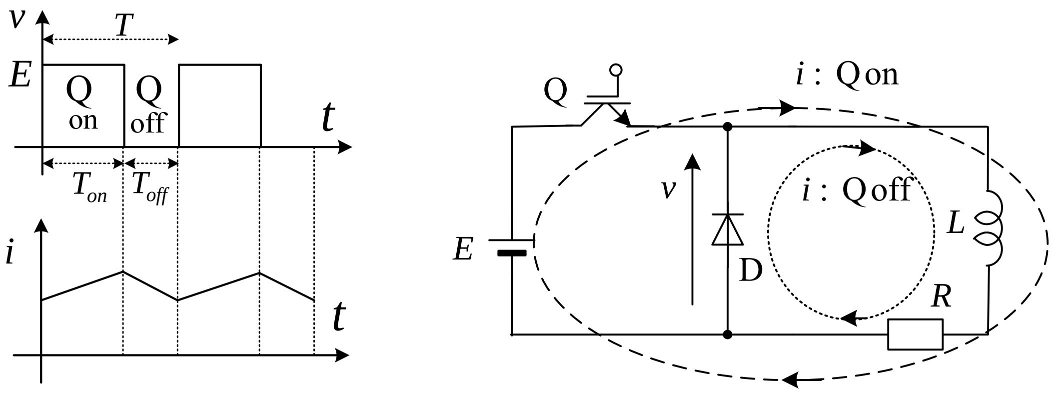


図 7-10　*RL* 負荷の電圧と電流

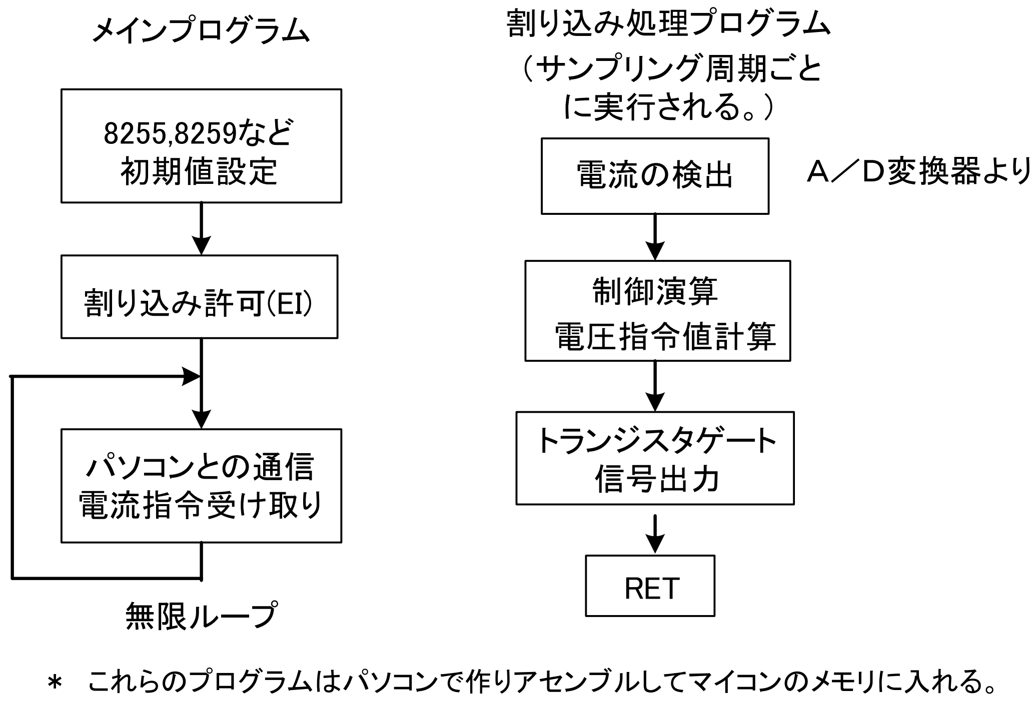


図 7-11　CPU が実行する制御プログラムの構成

[問題 7-1]　表 7-4 のメモリマップを導出せよ。

[問題 7-2]　ソフトウェア（プログラム）が CPU で実行される仕組みを説明せよ。

[問題 7-3]　キャリフラグ（C フラグ）とオーバフローフラグ（V フラグ）は何故必要か説明せよ。

[問題 7-4]　8255 の働きを説明せよ。

[問題 7-5]　割り込みとは何か。何故必要か。

[問題 7-6]　8259 の働きを説明せよ。

参考文献

(1) 成田誠之助：ディジタルシステム制御　理論と応用（昭晃堂），1980

(2) 明石一，今井弘之：詳解制御工学演習　8 章サンプル値制御系（共立出版），1981

(3) G.F.フランクリン，J.D.パウエル，羽根田博正訳：ダイナミックシステムのディジタル制御（森北出版），1981

(4) 横田英一：マイクロコンピュータ Z-80 の使い方（オーム社），1981

(5) 美多勉，原辰次，近藤良：基礎ディジタル制御（コロナ社），1988

(6) 金原昭臣，黒須茂：ディジタル制御入門（日刊工業新聞社）,1990

(7) Charles L. Phillips, H. Troy Nagle, JR. , 横山隆一，佐川雅彦，貴家仁志訳：ディジタル制御システム　解析と設計（日刊工業新聞社），1990

(8) 荒木光彦：ディジタル制御理論入門（朝倉書店），1991

(9) 相良節夫，和田清，中野和司：ディジタル制御の基礎（コロナ社），1992

(10) 岩田彰：ディジタル信号処理（コロナ社），1995

(11) 高木章二：ディジタル制御入門　改訂 2 版（オーム社），1999

(12) 高橋進一，池原雅章：ディジタルフィルタ（倍風館），1999

(13) 樋口龍雄，川又政征：ディジタル信号処理（昭晃堂），2000

(14) 三谷政昭，有井貴志：DSP によるディジタル信号処理プログラミング入門（森北出版），2000

(15) 松尾芳樹：ディジタル制御（昭晃堂），2001

(16) 辻峰男，坪井克剛，山田英二：誘導機ディジタルベクトル制御系の安定解析法，電気学会論文誌 D，第 123 巻，1 号，pp.21-29 , 2003

(17) 森泰親：演習で学ぶディジタル制御（森北出版）, 2012

(18) 辻峰男：電気回路講義ノート，長崎大学学術研究成果リポジトリ(NAOSITE),2014

(19) 辻峰男：自動制御の理論と応用，長崎大学学術研究成果リポジトリ(NAOSITE),2015

(20) 辻峰男：パワーエレクトロニクスと電動機制御入門，長崎大学学術研究成果リポジトリ(NAOSITE),2015

＊(18),(19),(20)は Google でタイトルを入力して自由にダウンロード可能

付録

零次ホールドの伝達関数のゲインと位相特性

$s = j\omega$ とおくと，以下の式が得られる。

$$G_{oh}(j\omega) = \frac{1-e^{-j\omega T}}{j\omega} = \frac{1-\cos\omega T + j\sin\omega T}{j\omega}$$

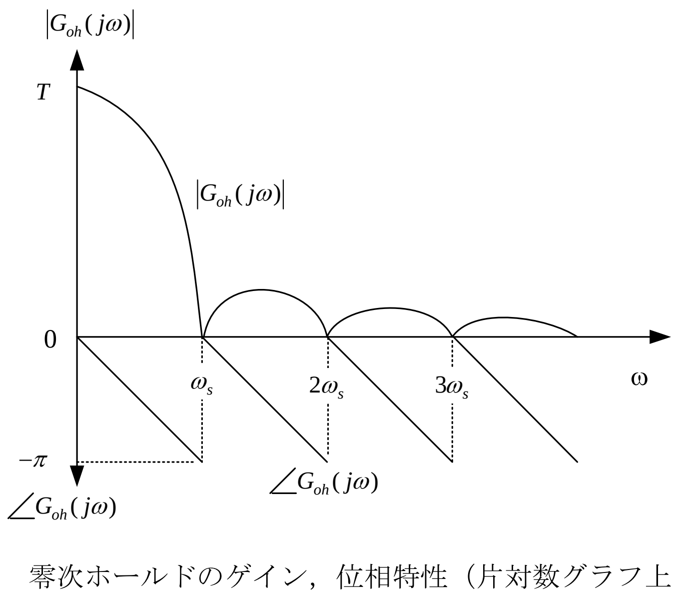
$$= \frac{1}{\omega}\{\sin\omega T - j(1-\cos\omega T)\} = \frac{2}{\omega}\sin(\frac{\omega T}{2})e^{-j\omega T/2}$$

$$= \frac{2}{\omega}\sin(\frac{\omega\pi}{\omega_s})e^{-j\pi\frac{\omega}{\omega_s}}$$

ただし， $\omega_s \equiv 2\pi / T$

$$\left|G_{oh}(j\omega)\right| = \frac{2}{\omega}\left|\sin(\frac{\omega\pi}{\omega_s})\right| = \frac{2\pi}{\omega_s}\frac{\left|\sin(\frac{\omega\pi}{\omega_s})\right|}{\frac{\omega\pi}{\omega_s}} = T\frac{\left|\sin(\frac{\omega\pi}{\omega_s})\right|}{\frac{\omega\pi}{\omega_s}}$$

$$\angle G_{oh}(j\omega) = \begin{cases} -\pi\dfrac{\omega}{\omega_s} & :(0<\omega<\omega_s) \\ -\pi\dfrac{\omega}{\omega_s}+n\pi & :(n\omega_s<\omega<(n+1)\omega_s) \end{cases}$$



零次ホールドのゲイン，位相特性（片対数グラフ上ではない）

零次ホールドは低域通過フィルタの一種で，図より不要な高域の信号をどのように減衰させるか理解できよう。直流に対するゲインはサンプリング周期 T になる。

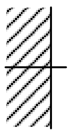
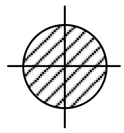
Z 変換表

$F(z) \equiv Z\{f(kT)\} \equiv Z(f(t)) \equiv Z(F(s))$ と書かれるが，$f(t)$ の z 変換を求めるには，数列 $\{f(kT)\}$ を求めて z 変換する。$F(s)$ の z 変換を求めるには，$f(t)$ を求め，次に $\{f(kT)\}$ を求め，それから z 変換を求めるのが本来の意味である。

時間関数 $f(t)$	$f(t)$ のラプラス変換 $F(s)$	$f(kT)$ の z 変換 $F(z)$	数列 $f(kT)$
		1	$1 : k = 0$ $0 : k \neq 0$
		z^{-n}	$1 : k = n$ $0 : k \neq n$
$u(t)$ ， 1	$\dfrac{1}{s}$	$\dfrac{z}{z-1}$	$1 : \text{all} \ \ k$
t	$\dfrac{1}{s^2}$	$\dfrac{Tz}{(z-1)^2}$	kT
		$\dfrac{z}{z-p}$	p^k
		$\dfrac{z}{(z-p)^2}$	$k\,p^{k-1}$
$\dfrac{t^2}{2}$	$\dfrac{1}{s^3}$	$\dfrac{T^2 z(z+1)}{2(z-1)^3}$	$\dfrac{(kT)^2}{2}$
e^{-at}	$\dfrac{1}{s+a}$	$\dfrac{z}{z-e^{-aT}}$	e^{-akT}
$1-e^{-at}$	$\dfrac{a}{s(s+a)}$	$\dfrac{(1-e^{-aT})z}{(z-1)(z-e^{-aT})}$	$1-e^{-akT}$
$t\,e^{-at}$	$\dfrac{1}{(s+a)^2}$	$\dfrac{T\,z\,e^{-aT}}{(z-e^{-aT})^2}$	kTe^{-akT}
$\sin\beta t$	$\dfrac{\beta}{s^2+\beta^2}$	$\dfrac{z\sin\beta T}{z^2-2z\cos\beta T+1}$	$\sin\beta kT$
$\cos\beta t$	$\dfrac{s}{s^2+\beta^2}$	$\dfrac{z(z-\cos\beta T)}{z^2-2z\cos\beta T+1}$	$\cos\beta kT$
$e^{-\alpha t}\sin\beta t$	$\dfrac{\beta}{(s+\alpha)^2+\beta^2}$	$\dfrac{z\,e^{-\alpha T}\sin\beta T}{z^2-2e^{-\alpha T}z\cos\beta T+e^{-2\alpha T}}$	$e^{-\alpha kT}\sin\beta kT$
$e^{-\alpha t}\cos\beta t$	$\dfrac{s+\alpha}{(s+\alpha)^2+\beta^2}$	$\dfrac{z(z-e^{-\alpha T}\cos\beta T)}{z^2-2e^{-\alpha T}z\cos\beta T+e^{-2\alpha T}}$	$e^{-\alpha kT}\cos\beta kT$

ただし，$k = 0, 1, 2, 3\cdots$

連続系の制御理論とディジタル制御理論の関連

	連続系	ディジタル制御系
変換による解法	ラプラス変換 ↓ ブロック図 ↓ 特性方程式の根 s （閉ループ伝達関数の分母 $=0$） ↓ 実部 安定条件：$R_e(s)<0$  根 s が左半平面内	z変換 ↓ ブロック図 ↓ 特性方程式の根 z （閉ループ伝達関数の分母 $=0$） ↓ 安定条件：$\lvert z\rvert<1$ 根 z が単位円内 
時間領域の解法	状態方程式（微分方程式） $\dfrac{d\boldsymbol{x}}{dt}=\boldsymbol{Ax}+\boldsymbol{Br}$ ← 指令値 ↓ 安定条件：$\boldsymbol{A}$ の固有値 s $R_e(s)<0$ 特性方程式の根と $\boldsymbol{A}$ の 固有値 s は一致する。	状態方程式（差分方程式） $\boldsymbol{x}(k+1)=\boldsymbol{Ax}(k)+\boldsymbol{Br}(k)$ ← 指令値 ↓ 安定条件：$\boldsymbol{A}$ の固有値 z $\lvert z\rvert<1$ 特性方程式の根と $\boldsymbol{A}$ の 固有値 z は一致する。
周波数特性	伝達関数 $G(s)$ $s=j\omega$ と置いて $G(j\omega)$ を計算して求める。	伝達関数 $G(z)$ $z=e^{j\omega T}$ と置いて $G(e^{j\omega T})$ を計算して求める。 T ： サンプリング周期

連続系については文献(19)参照

索　引