

図2.6-13 位置検出分解能の比較

### 第3章 速度応答を高速化する デッドビート制御の実現

### 第3章 速度応答を高速化するデッドビート制御の実現

#### §3-1 本章の課題と構成

前章では、PLL制御により定常的な電動機速度を非常に高精度に制御することが出来るが、過渡的な応答を高速化できないことを§2-5-4に示した。これに対して本章では、電動機速度の過渡的な応答を高速化するデッドビート制御系の実現について論じる。

デッドビート応答あるいは有限時間整定応答とは、離散時間制御系により実現可能な有限サンプル時間内に出力が整定するという高速な応答である。制御系の理想的な応答は、目標値の変化に誤差なく瞬時に追従し、また外乱に対しても出力が全く変動しないことである。しかし、このような応答は、物理的には実現不可能である。そこで、目標値及び外乱の変化に対して、有限時間、しかもできるだけ短時間で、制御誤差が0になるような制御が望まれる。このような応答はデッドビート応答あるいは有限時間整定応答と呼ばれている。

一般的には有限時間整定応答とデッドビート応答は区別されていないが、本文では説明の都合上、次のように区別することにする。

有限時間整定応答：出力が目標値に一致するまでの時間が有限である応答であり、特に整定途中の応答波形については限定しないもの。

デッドビート応答：有限時間整定応答の中でも、特に、ステップ状の目標値変化に対する出力応答に行き過ぎが生じない応答を指す。

応用の観点からは、出力応答の行き過ぎは好ましくないから、上に示したデッドビート応答が重要である。

本章の課題は、設定速度および負荷外乱のステップ状の変化に対して電動機速度をデッドビート応答させ過渡的な応答を高速化するデジタル制御系を実現することである。具体的には、次の①～③について詳細に述べる。

- ① 設定速度、あるいは、負荷外乱の変化に対してデッドビート応答させるために必要な制御器の構造を明らかにすること。
- ② 制御器のデジタル化（時間離散化）に伴う無駄時間を明らかにし、その補償法を提案すること。
- ③ 実験装置を製作し、デッドビート制御が可能であることを示し、提案する制御法の評価を行うこと。

これらを示すため、本章の構成は次のようになっている。

§3-2には、制御理論の分野における有限時間整定制御の研究の動向、および、有限時間整定制御の電動機駆動系への応用に関する研究の動向を述べる。

§3-3では、1入力1出力系の有限時間整定制御系の設計手順を示す。このため、まず、設計の基礎となる事項を定理の形でまとめ、その後で、伝達関数を利用した具体的な設計手順を示す。

§3-4では、設定速度および未知外乱のステップ状変化に対して、電動機速度のデッドビート応答を達成するための制御系の構造の候補として、PI制御およびI-P制御を取り上げて比較検討し、I-P制御の方が優れていることを示す。なお、ここでの比較検討は§3-4-1に示す仮定を設けることにより議論を単純化している。実際のシステムでは、無駄時間の補償が必要になるが、それについては§3-5と§3-6に2種類の対策を示す。

§3-5と§3-6では、実際にデッドビート制御を実現するために必要となる無駄時間の補償について2種類の方法を提案する。なお、補償すべき無駄時間は、§3-5-1に示すように、検出無駄時間と演算無駄時間の2つがある。

§3-5では、線形外挿 (§3-5-2) と出力フィードバック (§3-5-3) を組み合わせた方法を示す。ロータリエンコーダの出力パルスを計数すると電動機の平均速度が得られるが、1サンプリング周期の間に平均速度を2回検出し、そのデータを外挿することにより瞬時速度を求めることが出来る。また、演算処理に伴う無駄時間は、制御器の出力をフィードバックすることにより補償できる。この2つを組み合わせることによりデッドビート制御が出来ることをシミュレーションにより確認する。

§3-6では、予測型オブザーバを用いた方法を提案する。オブザーバを用いて平均速度から瞬時速度を推定し (§3-6-1)、さらに、オブザーバを利用して次回サンプリング時刻における瞬時速度を予測することにより演算無駄時間を補償する (§3-6-2) ののである。§3-6-3では、制御ゲインを計算し、デッドビート制御が可能であることをシミュレーションにより確認する。

§3-7では、さきに§3-5および§3-6に示した方法によりデッドビート制御が実現できることを実験により検証する。その結果、§3-6に示した予測型オブザーバを用いる方法の方が、電動機速度の振動が少ないことが明らかとなる。

§3-8では、量子化誤差を考慮した電動機速度誤差の解析・検討を行う。これまでは、有限時間整定制御系の実現方法に焦点を絞るため量子化誤差の影響

を無視してきた。しかし、デジタル制御系では、量子化誤差が必然的に発生し、これが定常的な精度に大きな影響を与えるから、本節で検討しておく。§ 3-8-1では、量子化誤差のあるデジタル制御系を解析し、§ 3-5の方法では不安定ブロックを含むため、§ 3-6の方法に比べ電動機速度振動が大きくなることを明らかにする。§ 3-8-2では、位置検出の量子化誤差を考慮したデジタルシミュレーション手法を示す。§ 3-8-3では、量子化誤差による電動機速度の誤差振幅を低減するため予測型オブザーバのゲインを可変にする方法を提案し、その有効性をシミュレーションにより確認する。

§ 3-9は本章のまとめであり、§ 3-10には本章の参考文献を示す。

### § 3-2 研究の動向

有限時間整定制御の研究動向については、特に制御理論の分野での動向についての解説がある[1,2]。これらによれば、制御理論における従来の研究のほとんどは、有限時間整定応答を問題としており、応答波形に行き過ぎがないこと、すなわち、前節で示したデッドビート性についてはあまり問題にされていなかった。応答波形を考慮した研究としては、わずかに、固有ベクトル配置により応答波形の調整を試みた Sebakhy と Abdel-Moneim の研究[3]、および、1入力1出力系においてサンプリング周期と行き過ぎ量の関係を調べた Mita の研究[4]がある程度である。

一方、有限時間整定制御の研究は、古くから行われており、Bergen と Ragazzini [5] のモデルマッチング法により補償器の伝達関数を決定する方法にまでさかのぼることができる。しかし、この方法では、極-零相殺の問題、あるいは、サンプリング周期内でのリップル（影の振動）などの問題が多く、その後、状態空間法に基づく Kalman の研究 [6,7,8] に発展した。1970年代には入って、Kalman の結果の多入出力系への拡張が試みられ、また、その双対であるデッドビートオブザーバの研究も進められた。これらの制御理論分野での動向については、解説[1,2]を参考にした。

これらの制御理論家の活発な研究にもかかわらず、有限時間整定制御の電動機駆動系への応用は、1980年代にはいるまで極めて少なかった。1980年代にはいると、制御演算を実現するマイクロプロセッサを利用できる環境が整ったため、有限時間整定制御の電動機駆動系への応用に関する研究が次第に行われるようになった。

深尾ら[9]は、無整流子電動機駆動用インバータの直流入力電流制御へ有限時間整定制御を応用した。用いた手法は、モデルマッチング法により極-零相殺を行うものである。

村田らは、[12]で制御対象のインパルス応答から有限時間整定応答を実現するために必要な制御入力を計算しておく方法を提案し、[13]ではそれを応用して2.2kW 誘導電動機速度を開ループバタン制御により有限整定させた。

Holtz ら[14]は、14kW直流電動機のサイリスタレオナード装置を利用し、電機子電流の有限時間整定制御を行った。この方法は、前向きループに挿入した補償要素により極-零相殺を行うものである。ただし、軽負荷時の電流断続モードを考慮して、補償要素を連続モードの時と切り替えるようにしていることと、トルク外乱に対してオフセットをなくすため電機子電流の積分制御を併用していることに特徴がある。

Depping ら[15,16]は、15W永久磁石界磁直流電動機速度の有限時間整定制御を行った。この方法も、前向きループに挿入したフィルタにより極-零相殺を行うものである。ただし、運転中の電動機パラメータ変動に対するため、速度制御と同時にパラメータ同定を行い、フィルタの定数を調整することに特徴がある。井上ら[17]も同様な制御方式を提案している。

これらは、いずれも前向きフィルタにより極-零相殺をしているものであるが、入力に飽和が生じると相殺されたモードが出力に現れかえって応答を遅くするという問題がある。すなわち、著者らは、同様なモデルマッチング法を使用して電動機速度の有限時間整定制御を行うためのフィルタを求めシミュレーションを行ったが、この方法では電動機駆動用電力変換器に飽和が生じると、非常に遅いモードの応答が出力に発生することを確認している[10]。著者らは、このように制御ループ内に飽和要素がある場合には、極-零相殺ではなく、状態フィードバックにより有限時間整定させるべきであると考え、オブザーバを応用した実現方法を提案した[11]。

有限時間整定制御系の具体的な設計法としては、たとえば、ゲイン変化法[18]がある。また、制御則を状態フィードバックに限定するならば、極配置法[19,20]を利用して制御系の極をz-平面の原点に移動させるような制御ゲインを計算することもできる。

デジタル制御器では、演算処理に要する時間が制御系の無駄時間として作用し安定性を劣化させるという問題があり、その補償が必要である。この無駄時間に対しては、本章に示すようにオブザーバによる予測が有効な対策である。予測

値を用いて無駄時間を補償するという考え方自体は、特別に新しいものではなく、1951年にKalmanにより、シュミレータを用いて補償を行うという方法が提案されている[第4章 34]。しかし、無駄時間の問題は、マイクロプロセッサを用いてより高速な制御系を設計しようとする事が多くなるに従い、再び、注目されるようになり、美多による解説[第4章 35]がある。

### § 3-3 有限時間整定制御系の設計手順

ここでは、1入力1出力系の有限時間整定制御系の設計手順を示す。このため、まず、設計の基礎となる事項を定理の形でまとめ、その後で、伝達関数を利用した具体的な設計手順を示す。直感的な理解のしやすい伝達関数を主に使うが、状態空間表現も一部使用する。伝達関数を利用する場合には、比較的低次系では数式処理がしやすいという利点があるが、制御ループ内で極-零相殺をすることもあるので注意しておかなければならない。

まず、設計の基礎となる事項を述べる。設定値または外乱から出力までのパルス伝達関数が、それぞれ、 $G_1(z)$  または  $G_2(z)$  であるとき、有限時間整定するために、各々の伝達関数は次の2つの定理に示すような条件を満たさなければならない。

【定理 3.3.1】 {設定値の変化に対する有限時間整定}

設定値から出力までの伝達関数を  $G_1(z)$  とする。設定値のステップ変化に対して出力が有限時間整定するための必要十分条件は、次の①、②が同時に成立することである。

①  $G_1(z)$  がある有限な整数  $k$  に対して  $z^{-1}$  の有限べき級数に展開できること：

$$G_1(z) = \sum_{j=0}^k a_j z^{-j} \quad (3.1)$$

② 次式が成り立つこと：

$$\sum_{j=0}^k a_j = 1 \quad (3.2)$$

□

【定理 3.3.2】 {外乱の変化に対する有限時間整定}

外乱から出力までの伝達関数を  $G_2(z)$  とする。外乱のステップ変化に対して出力が有限時間整定するための必要十分条件は、次の①、②が同時に成立することである。

①  $G_2(z)$  がある有限な整数  $k$  に対して  $z^{-1}$  の有限べき級数に展開できること：

$$G_2(z) = \sum_{j=0}^k b_j z^{-j} \quad (3.3)$$

② 次式が成り立つこと：

$$\sum_{j=0}^k b_j = 0 \quad (3.4)$$

□

(注1) 式(3.1)、(3.3)は、応答が有限時間内で納まることを保証している。

(注2) 式(3.2)は、 $k$ 回目のサンプリング以降に(出力) = (設定値)となることを保証している。

(注3) 式(3.4)は、 $k$ 回目のサンプリング以降に外乱による出力誤差が0になることを保証している。

上の2つの定理では、各々の伝達関数が  $z^{-1}$  の有限べき級数に展開できることが要求されているが、それが可能であるための条件はこれから示す【定理 3.3.3】で与えられる。それを示す前に、システムの状態空間表現から求められる伝達関数表現について注意しておく。

システムの状態方程式が

$$\mathbf{x}(k+1) = \mathbf{A} \mathbf{x}(k) + \mathbf{b} u(k) \quad (3.5)$$

$$y(k) = c \mathbf{x}(k) \quad (3.6)$$

の時、 $u$  から  $y$  への伝達関数  $G_3(z)$  は次式で与えられる。

$$G_3(z) = c(zI - \mathbf{A})^{-1} \mathbf{b} \quad (3.7)$$

ただし、本文では特別に、 $G_3(z)$  は、分子・分母に共通因子があっても通分をしていない形のものを表すことにする。これは、極-零相殺をする前の伝達関数を  $G_3(z)$  としていることに相当する。

上記のような伝達関数が、 $z^{-1}$  の有限べき級数に展開されるための条件は、次のように与えられる。

【定理 3.3.3】 {伝達関数の有限べき級数展開}

ある伝達関数  $G_3(z)$  が  $z^{-1}$  の有限べき級数に展開できるための十分条件は、次の①または②が成り立つことである。

① 伝達関数  $G_3(z)$  の 0 でない極は、零点と相殺する。

② 伝達関数  $G_3(z)$  の全ての極が 0 である。 □

(注 4) 上記で、条件①を満たす場合には、極-零相殺が起こり、不可制御なモードが生じる。一方、条件②を満たす場合は、状態フィードバックなどにより極を原点に配置することに相当する。後者では、不可制御な状態が残らなく、状態有限時間整定制御と呼ばれる。

以上で基礎事項の説明を終わり、以下には、伝達関数を使用した有限時間整定制御系の設計手順を示す。

《有限時間整定制御系の設計手順》

(STEP1) 制御対象の伝達関数  $G_p(z)$  を求める。

(STEP2) 適当な制御則を仮定して、その伝達関数  $G_c(z)$  を求める。

(STEP3)  $G_c(z)$  の中に共通な極と零があれば相殺する。(注 5 参照)

(STEP4) 制御系の閉ループ伝達関数  $G_f(z)$  を求める：

$$G_f(z) = \frac{G_c(z)G_p(z)}{(1 - G_c(z)G_p(z))} \quad (3.8)$$

(STEP5)  $G_f(z)$  の分母多項式中で、 $z$  の最高次の係数だけを残しほかの低次項の係数をすべて 0 にするように制御則のゲインを定める。(注 6 参照)

ここで、もし低次項の係数で 0 でないものが残るようなら、(STEP2) で仮定した制御則の構造では有限時間整定制御が不可能であるから、別な制御則を考えて (STEP2) からやり直す。 □

(注 5) (STEP3) は、 $G_c(z)$  の次数を下げてそれ以降の計算を簡単化するためのものである。また、このように制御器の内部で極-零相殺を行っても、制御計算の桁数を十分に大きくしておけば、相殺されたモードを励起しないから問題ない。

(注 6) ここで、 $G_f(z)$  の分子分母で極-零相殺をしてもよいが、それは避けた方がよい。その理由は、制御器と制御対象のつなぎの部分(検出データあるいは制御入力を受渡しの部分)に非線形性があると、極-零相殺が不完全となり、相殺され残ったモードの影響が出力に現れるからである。

### § 3-4 応答波形を考慮した制御系構造の決定

ここでは、設定速度および未知外乱のステップ状変化に対して、電動機速度のデッドビート応答を達成するための制御系の構造の候補として、PI 制御および I-P 制御を取り上げて比較検討し、I-P 制御の方が優れていることを示す。なお、ここでの比較検討は § 3-4-1 に示す仮定を設けることにより議論を

簡単化している。実際のシステムでは、無駄時間の補償が必要になるが、それについては § 3-5 と § 3-6 に 2 種類の対策を示す。

### § 3-4-1 検討のための仮定と電動機の離散時間モデル

議論を簡単にするため以下の仮定を設ける。

(A1) 制御対象である直流電動機の電機子電圧入力  $u(t)$  から電動機速度出力  $\omega_m(t)$  への伝達関数  $G_m(s)$  が次のように表わされるものとする。

$$G_m(s) = \frac{\omega_m(s)}{u(s)} = \frac{K_m}{T_m s + 1} \quad (4.1)$$

(A2) サンプリング周期を  $T_s$  とし、その間は  $u(t)$  が一定であるものとする。

(A3) 電動機パラメータ  $K_m$ ,  $T_m$  は正確に知られており、変動しないものとする。

(A4) 電動機速度は、直接検出できるものとする。

(A5) 制御器の動作遅れはないものとする。

#### <各仮定に関する注意>

(A1) この場合は直流機の電氣的時定数を無視している。厳密には、直流機の電機子インダクタンス、および、駆動用チョッパに挿入される直流リアクトルを考慮すれば、電動機の伝達関数は、電気系の遅れまで入れて二次系となる。しかし、電気系の遅れが、機械系の遅れおよび閉ループ系の応答時間に比べて十分小さければ、電気系の遅れは無視できるものと考えられる。その妥当性は、仮定 (A1) のもとに設計された制御系により実験的に有限時間整定制御が可能である事実 (後述の § 3-7-3) により確認される。

(A2) 電動機を駆動する PWM チョッパのキャリヤ周波数が制御のサンプリング周波数に比べて十分大きければ、チョッパはサンプリング周期の間一定の電圧を電動機に供給していると考えられる。このときチョッパは、零次ホールドと考えることが出来る。

(A3) 電動機パラメータが不正確であったり、運転中に変動することがある。このようなパラメータ変動の影響は、第 4 章で扱う。

(A4) 電動機速度が直接に検出できない場合については、§ 3-4 と § 3-5 で扱う。

(A5) 実際のシステムでは、制御演算に要する時間が動作遅れになる。その対策については、§ 3-5 と § 3-6 で述べる。

次に、直流電動機の離散時間モデルを示す。図 3.4-1(a) に、直流電動機に位置検出用のエンコーダを付加した系のブロック図を示す。(エンコーダは、あとで、検出器として利用するためここで考慮に置いておく。) 仮定 (A2) で、 $T_s$  の間は直流電動機への入力が一一定であるとしたから、図 3.4-1(a) から同図(b) の離散時間モデルを得る。直流電動機のパルス伝達関数は、

$$G_m(z) = \frac{Q}{z - P} \quad (4.2)$$

ただし、 $P$ ,  $Q$  および 同図(b) の  $R$ ,  $S$  は、

$$P = \exp(-T_s/T_m) \quad (4.3a)$$

$$Q = K_m(1 - P) \quad (4.3b)$$

$$R = T_m(1 - P) \quad (4.3c)$$

$$S = K_m(T_s - R) \quad (4.3d)$$

また、同図(b) の、 $\Delta\theta_m$  は電動機が  $T_s$  の間に回転する機械角度である。

### § 3-4-2 PI 制御と I-P 制御の比較

ここでは、設定速度、および、未知外乱のステップ状変化に対して、電動機速度の有限時間整定応答を達成するための制御系の構造の候補として、PI 制御および I-P 制御を取り上げて比較検討する。

[ 1 ] P I 制御

図3.4-2 に、P I 制御系のブロック図を示す。この図から、外乱および設定値のそれぞれの変化に対する各パルス伝達関数は、

$$\text{外乱: } G_{PI\tau}(z) = \frac{\omega_m(z)}{V_\tau} = \frac{Q(z-1)}{G_o(z)} \quad (4.4)$$

$$\text{設定値: } G_{PIr}(z) = \frac{\omega_m(z)}{\omega_r(z)} = \frac{K_i Qz + K_p Q(z-1)}{G_o(z)} \quad (4.5)$$

ただし、

$$G_o(z) = z^2 + [Q(K_i + K_p) - P - 1]z + P - K_p Q \quad (4.6)$$

上記のパルス伝達関数で与えられる系に対して、有限時間整定を行うためのゲイン  $K_i$ 、 $K_p$  の設定方法には、次の二通りがある。

[ 1-1 ] パルス伝達関数の極を  $z$  平面の原点に配置する

このためには、式(4.6)の  $z$  の係数と定数項を0にすればよいから、

$$Q(K_i + K_p) - P - 1 = 0 \quad (4.7.a)$$

$$P - K_p Q = 0 \quad (4.7.b)$$

とすれば、極が原点に配置される。そのためのゲインは式(4.7a-b)から

$$K_p = P/Q, \quad K_i = 1/Q \quad (4.8)$$

以下に実際に有限時間整定していることを確認する。式(4.8)を式(4.4),(4.5)にそれぞれ代入して

$$G_{PI\tau}(z) = \frac{Q(z-1)}{z^2} \quad (4.9)$$

$$G_{PIr}(z) = \frac{(1+P)z - P}{z^2} \quad (4.10)$$

式(4.9), (4.10)から、外乱および設定速度の単位ステップ変化に対する電動機速度の応答は

$$\text{外乱: } \omega_m(z) = \frac{Q(z-1)}{z^2} \cdot \frac{z}{z-1} = Qz^{-1} \quad (4.11)$$

$$\text{設定値: } \omega_m(z) = \frac{(1+P)z - P}{z^2} \cdot \frac{z}{z-1} = (1+P)z^{-1} + z^{-2} + z^{-3} \dots \quad (4.12)$$

この応答を図示すれば、図3.4-3 のようになる。同図から、電動機速度  $\omega_m$  は確かに有限時間整定をする。しかし、設定速度  $\omega_r$  のステップ状変化に対して、電動機速度  $\omega_m$  にオーバーシュートが発生する。(デッドビート応答ではない。)これは、電力変換器の容量として過大なものを必要とし、実現上好ましくない。(次に示す設計法によれば、設定値変更に対する応答にオーバーシュートのない有限時間整定をP I 制御により行うことができる。)

[ 1-2 ] パルス伝達関数において極・零相殺を行う

このためには、式(4.5)のパルス伝達関数  $G_{PIr}(z)$  が次のように、分子・分母に同じ因子を持つようにできればよい。

$$G_{PI\tau}(z) = \frac{(az - b)}{z(az - b)} \quad (4.13)$$

このためには、式(4.5)と(4.13)を比較して、

$$a = Q(K_i + K_p) = 1 \quad (4.14a)$$

$$b = K_p Q = Q(K_i + K_p) - P - 1 \quad (4.14b)$$

$$P - K_p Q = 0 \quad (4.14c)$$

であればよい。従って、制御ゲインは次のようになる。

$$K_p = P/Q, \quad K_i = (1 - P)/Q \quad (4.15)$$

この時、設定値の変化に対する伝達関数が式(4.13)となるから、確かに、設定値のステップ変化に対しては有限時間整定する。一方、外乱に対するパルス伝達関数は、式(4.15)を式(4.4)に代入して、

$$G_{PI\tau}(z) = \frac{Q(z - 1)}{z(z - P)} \quad (4.16)$$

式(4.16)から、ステップ状外乱変化に対しては  $\omega_m$  が有限時間整定しないことがわかる。この時の電動機速度の応答の例をを図3.4-4 に示す。

以上の結果より、PI制御によって有限時間整定制御を実現する場合には、二種類の制御ゲインを切り替えて使う必要があることがわかる。すなわち、外乱に対しては式(4.8)で、また、設定値変更に対しては式(4.15)で、それぞれ与えられる制御ゲインを切り替える必要がある。

## [2] I-P制御

I-P制御則を用いれば、制御ゲインの切り替えを行わなくても設定値および外乱のステップ変化に対して有限時間整定制御が可能であることが、以下の検討から分かる。

I-P制御系のブロック図は、図3.4-5 に示すようになる。同図から、外乱および設定値に対する電動機速度へのパルス伝達関数は

$$\text{外乱: } G_{I-P\tau}(z) = \frac{Q(z - 1)}{G_o(z)} \quad (4.17)$$

$$\text{設定値: } G_{I-P\tau}(z) = \frac{K_i Q z}{G_o(z)} \quad (4.18)$$

ただし、 $G_o(z)$  は式(4.6)で与えられる。既に示したように、 $G_o(z)$  の極をすべて原点に配置するためには、制御ゲインを式(4.8)のように与えればよい。このように制御ゲインを定めれば、外乱および設定値のステップ状変化に対する電動機速度  $\omega_m$  の応答は、

$$\text{外乱: } \omega_m(z) = G_{I-P\tau}(z) \frac{z}{z - 1} = Qz^{-1} \quad (4.19)$$

$$\text{設定値: } \omega_m(z) = G_{I-P\tau}(z) \frac{z}{z - 1} = \frac{K_i Q}{z - 1} \quad (4.20)$$

式(4.19)、(4.20)から、外乱、および、設定値のステップ状変化に対する電動機速度の応答は、図3.4-6 のようになる。

このことから、I-P制御則を用いることにより制御ゲインの切り替えをしなくても外乱および設定値変更に対して有限時間整定させることができることが分かる。しかも、設定値変更に対する電動機速度の応答にオーバーシュートが生じない。

参考のため、PI制御系とI-P制御系の応答の比較を図3.4-7 に示す。PI制御系の設定値に対するステップ応答波形は、I-P制御系のステップ応答波形

にPI制御系の負荷応答波形を加えたものになっている。

以上の〔1〕と〔2〕の結果を比較して、外乱および設定値のステップ状変化の両方を考慮するためには、I-P制御の方が優れていることが分かる。従って、これから後の検討においては、制御系の構造としてI-P制御を採用することにする。

### § 3-5 無駄時間の補償方法 I

#### (線形外挿と制御入力フィードバックを組み合わせた補償法)

前節(§ 3-4)に示した検討結果により、デッドビート制御を実現するための制御則としてI-P制御が適していることが明らかになった。しかし、実際にデッドビート制御を実現するためには、§ 3-4-1で設けた仮定(A4)、(A5)をどの様に実現するか、すなわち、無駄時間補償をどの様にして行うかを明らかにする必要がある。

この節と次節では、実際にデッドビート制御を実現するために必要となる無駄時間の補償方法について述べる。§ 3-5-1に示すように、補償すべき無駄時間には検出無駄時間と演算無駄時間の2つである。これらの無駄時間の補償方法として2種類の方法を提案する。ひとつは、本節に示す線形外挿(§ 3-5-2)と制御入力フィードバック(§ 3-5-3)を組み合わせたものである。もうひとつは、次節に示す予測型オブザーバを用いたものである。これら2種類の比較については、§ 3-7および§ 3-8-1に示す。

#### § 3-5-1 補償すべき無駄時間

補償すべき無駄時間には、前節の§ 3-4-1の仮定(A4)、(A5)にそれぞれ対応して、次の2つがある。

(A4) 電動機速度の検出に伴う無駄時間(検出無駄時間)

(A5) マイクロプロセッサの演算時間による無駄時間(演算無駄時間)

ここでは、それぞれを検出無駄時間、および、演算無駄時間と呼ぶことにする。各々の発生原因を以下に説明する。

#### 〔1〕 検出無駄時間

検出無駄時間は、実験装置において採用した電動機速度の検出方法によって生じるものである。すなわち、制御のデジタル化、検出器の高信頼度・高精度化(オフセット、ドリフトの除去)の観点から、電動機回転子位置の検出にロータリエンコーダを採用した。ロータリエンコーダは、電動機軸に直結された回転円板に設けられたスリットをLEDとフォト・トランジスタにより検出するものであり、本質的には、位置検出器である。実験装置では電動機速度を検出するため、サンプリング周期 $T_s$ の間に検出されるパルス数を計数するようにした。この方法で検出される速度は、図3.5-1に示すように、サンプリング周期の間の平均速度であり、過渡時は、サンプリング時刻での瞬時速度には一致しない。一方、デッドビート制御のためには電動機速度の瞬時値を検出しなければならない。同図からわかるように、電動機の時定数に比べサンプリング周期 $T_s$ が十分小さい場合は、時刻 $(k+1)T_s$ において検出される $T_s$ 間の平均速度は、時刻 $(k+0.5)T_s$ における瞬時速度にほぼ等しい。いい換えれば、平均速度は、瞬時速度に比べて約 $T_s \times 0.5$ の無駄時間を含んでいる。

従って、ロータリエンコーダを使用して検出される平均速度から、各サンプリング時刻における瞬時速度を推定する方法を考案する必要がある。

#### 〔2〕 演算無駄時間

マイクロプロセッサを制御器に使用する場合を考えると、マイクロプロセッサが検出データを読み込み、演算を完了して制御信号を出力するまでの時間を考慮しなければならない。一般に、マイクロプロセッサの処理速度は遅いため、制御プログラムの処理に要する時間は、制御のサンプリング周期に比べて無視できない大きさである。むしろ、制御演算に要する時間より若干長めにサンプリング周期を設定することが多い。この場合の制御の無駄時間は、1サンプリング周期に近いものとなる。

図3.5-2に、本章で考えている電動機速度のデッドビート制御系における、マイクロプロセッサへのデータ入出力のタイミングチャートを示す。単一サンプリング周期系として解析できるようにするため、検出データのマイクロプロセッサへの読み込み、および、電動機への制御入力の変更は、それぞれ同一のサンプリング時刻で行うようにしている。(ここで、マイクロプロセッサの演算処理完了と同時に制御入力を変化させるようにすることも可能ではある。しかし、その場合は、制御系が多重サンプリング周期系となり、解析・設計が面倒になる。)同図から、マイクロプロセッサが検出データを読み込んでから、制御入力に変化す

るまでの動作遅れ時間はちょうど1 サンプル周期  $T_s$  となる。

ゆえに、とくに高速な応答を実現しようとする時には、制御プログラムの処理に要する演算時間を無駄時間として考慮して、それを補償するように制御系を設計しておかなければならない。

図3.5-3 に、上記の2種類の無駄時間を考慮に入れた制御系のブロック図を示す。

### § 3-5-2 線形外挿による瞬時速度推定

ロータリエンコーダが1 サンプル周期  $T_s$  間に発生するパルス数を計数したデータから瞬時速度を線形外挿する方法を以下に示す。

図3.5-4 に示した説明図において、太実線は瞬時速度  $\omega_m$  の時間変化を示す。ここで、時刻  $k+1$  での瞬時速度  $\omega_m(k+1)$  を推定することを考える。まず、サンプル周期  $T_s$  を二等分して、前半の半周期間と後半の半周期間のそれぞれのパルス数を計数すれば、平均速度  $\overline{\omega_{m1}}$  および  $\overline{\omega_{m2}}$  を検出できる。ここで、サンプル周期の間は、電機子電圧が一定で、電動機の定数がサンプル周期  $T_s$  に比べて十分大きければ、 $T_s$  間の瞬時速度の変化はほぼ直線と考えられる。そこで瞬時速度  $\omega_m(k+1)$  の推定値  $\hat{\omega}_m(k+1)$  は、図3.5-4 を参考にして次式で与えられることがわかる。

$$\hat{\omega}_m(k+1) = \overline{\omega_{m2}} + \frac{\overline{\omega_{m2}} - \overline{\omega_{m1}}}{2} \quad (5.1)$$

このように線形外挿により瞬時速度の推定を行えば時刻  $k+1$  において、推定値  $\hat{\omega}_m(k+1)$  を得ることができるから、速度検出に伴う無駄時間を補償できる。(なお、式(5.1)の計算に要する時間は、厳密には速度検出に伴う無駄時間とするべきであるが、便宜上以下に述べる制御演算に要する無駄時間に含めて考えることにする。)

この方式は、著者らの発表文献[B-3]により初めて提案されたものであるが、大前ら[21]はこれを拡張して、速度を3回検出して2次式により瞬時速度を推定する方法を提案し、サイリスタ・レオナードの全デジタル速度制御装置に応用した。その結果、その方法が、設定速度のステップ応答のオーバーシュートを抑え

るために有効であることが確認されている。

### § 3-5-3 制御入力フィードバックによる演算無駄時間の補償

マイクロプロセッサの制御演算処理に要する時間は、図3.5-3 に示したように制御器の出力部に挿入された遅れ時間となる。以下にはこの演算処理に伴う無駄時間を状態フィードバックにより補償する方法を検討する。

可制御な系では、状態フィードバックにより、閉ループ系の固有値を任意に配置することができるが知られている。そこで、演算無駄時間を含む系において、制御対象への制御入力  $u$  も状態変数の一つと考えて、これをフィードバックすれば、すべての固有値を  $z$  平面の原点に配置でき、デッドビート制御が可能となると考えられる。このような考察に基づいて制御入力のフィードバックを行ったI-P制御系のブロック図を図3.5-5 に示す。同図において、破線で囲んだ部分が、制御演算による遅れとそれを補償するためのフィードバックゲイン  $K_1$  である。ただし、I-P制御における、積分ループへのフィードバック量は回転角度の増分  $\Delta\theta_m$  とし、内側の比例ループへは前の§ 3-5-2に示した瞬時速度の推定値  $\hat{\omega}_m$  をフィードバックするようにした。(積分ループに瞬時速度の推定値を用いない理由は、推定誤差により電動機速度のオフセット誤差が生じるのを避けるためである。)

さて、デッドビート制御に必要な制御ゲインは、制御系の固有値を  $z$  平面の原点に配置する方針のもとに、次のようにして求められる。まず、図3.5-5 において、 $\omega_m = \hat{\omega}_m$  と仮定して、ブロック図を書き直せば、図3.5-6 を得る。このブロック図から、系の状態方程式は、

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{b}\omega_r' \quad (5.2)$$

ただし、

$$\mathbf{x}(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} = \begin{bmatrix} \omega_m(k) \\ u(k) \\ x_3(k) \end{bmatrix} \quad (5.3a)$$

$$A = \begin{bmatrix} P & Q & 0 \\ -K_p & -K_1 & 1 \\ -K_i R & -K_i S & 1 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ K_i T_S \end{bmatrix} \quad (5.3b)$$

行列  $A$  の固有多項式は、

$$\det(zI - A) = z^3 + a_2 z^2 + a_1 z + a_0 \quad (5.4)$$

ただし、

$$a_2 = K_1 - P - 1 \quad (5.5a)$$

$$a_1 = P - K_1 P - K_1 + K_i S + Q K_p \quad (5.5b)$$

$$a_0 = K_1 P + K_i R Q - K_i P S - K_p Q \quad (5.5c)$$

式(5.5a-c)で、 $a_2 = a_1 = a_0 = 0$  とできれば、系の固有値が  $z$  平面の原点に配置できる。そこで、式(5.5a-c)で各右辺 = 0 とし、必要な制御ゲインが次のように決まる。

$$K_1 = P + 1 \quad (5.6a)$$

$$K_p = \frac{-P^3 T_S + R}{T_S Q(1-P)} \quad (5.6b)$$

$$K_i = \frac{1}{T_S Q} \quad (5.6c)$$

のように必要な制御ゲインを得る。式(5.6a-c) で与えられる制御ゲインを使ってシミュレーションした結果を図3.5-7 に示す。予想どおり、制御入力  $u$  をフィードバックすることにより、演算の無駄時間を補償してデッドビート制御ができることを確認できる。

以上により、線形外挿による瞬時速度の推定と制御入力フィードバックによる演算時間補償方法を組み合わせることにより、デッドビート制御を実現出来ることが明らかになった。

### § 3-6 無駄時間の補償方法 II

#### (予測型オブザーバによる無駄時間の補償)

ここでは、検出無駄時間と演算無駄時間を補償する方法として、前節とは別な方法、すなわち、予測型オブザーバを利用する方法を示す。すなわち、オブザーバを用いて平均速度から瞬時速度を推定し(§ 3-6-1)、さらに、オブザーバを利用して次回サンプリング時刻における瞬時速度を予測することにより演算無駄時間を補償する(§ 3-6-2)のである。

#### § 3-6-1 オブザーバによる電動機瞬時速度の推定

ここでは、オブザーバにより電動機瞬時速度を推定することにより検出に伴う無駄時間を補償する方法を示す。まず、オブザーバの設計法を準備してから、次に、電動機の瞬時速度を推定するデッドビートオブザーバを導入する。

#### [1] オブザーバの構成法

最初に、準備としてオブザーバの構成法を示しておく。制御対象は、1入力1出力で次のように表されるものとする。

$$\mathbf{x}(k+1) = A\mathbf{x}(k) + \mathbf{b}u(k) + Dv(k) \quad (6.1a)$$

$$y(k) = cx(k) \quad (6.1b)$$

ただし、式(6.1a)の右辺第3項の  $v$  は外乱を表すベクトルである。(一般に、オブザーバは、系が確定的であるという立場に立って設計されるから  $v=0$  として設計されることが多い。しかし、外乱に対する応答も高速化したいから  $v$  を考慮に入れておくことにする。) 式(6.1)の制御対象に対する同次元オブザーバは、次式で与えられる。

$$\hat{x}(k+1) = \hat{A}\hat{x}(k) + \hat{b}u(k) + f(y(k) - \hat{c}\hat{x}(k)) \quad (6.2)$$

図3.6-1 に、同次元オブザーバの構成を示す。同図から、オブザーバゲイン  $f$  は、出力誤差を減少させるための修正を行っていることがわかる。

ここで、 $x$  と  $\hat{x}$  の誤差ベクトル  $e$  を

$$e(k) = \hat{x}(k) - x(k) \quad (6.3)$$

と定義すれば、式(6.2)-(6.1)から

$$e(k+1) = (\hat{A} - f\hat{c})\hat{x}(k) + \hat{b}u(k) + fcy(k) - [Ax(k) + bu(k) + Dv(k)] \quad (6.4)$$

さらに、制御対象とオブザーバとの係数行列の誤差を考慮して、

$$\hat{A} = A + \Delta A \quad (6.5a)$$

$$\hat{b} = b + \Delta b \quad (6.5b)$$

$$\hat{c} = c + \Delta c \quad (6.5c)$$

とすれば、式(6.4), (6.5a-c)から

$$e(k+1) = [A - fc]e(k) + [(\Delta A - f\Delta c)\hat{x}(k) + \Delta bu(k)] - Dv(k) \quad (6.6)$$

式(6.6)右辺の第2項は、オブザーバと制御対象の係数行列の誤差によるもので、第3項は外乱によるものである。それぞれは、誤差ベクトル  $e$  に対する強制外力となる。ここで、係数誤差がなく、かつ、外乱もないとすれば、

$$\Delta A = 0, \quad \Delta b = 0, \quad \Delta c = 0$$

$$v(k) = 0$$

となり、式(6.6)は

$$e(k+1) = (A - fc)x(k) \quad (6.7)$$

となる。式(6.7)は自由系であり、行列  $A - fc$  が漸近安定であれば、 $e$  は 0 に収束する。特に、行列  $A - fc$  のすべての固有値が 0 であれば、任意の初期誤差に対して  $e(k)$  は、高々  $n$  サンプル目まで 0 とすることができる。(ただし、 $n$  は  $A$  の次数。) このようなオブザーバをとくに、デッドビートオブザーバと呼ぶ。

## [2] 電動機の瞬時速度を推定するデッドビートオブザーバの構成法

以上の準備のもとに、ロータリエンコーダ付き電動機に対する同次元オブザーバを求める。式(4.2), (6.2)を参考にしてオブザーバは次式で与えられる。

$$\hat{x}(k+1) = \hat{A}\hat{x}(k) + \hat{b}u(k) + f(y(k) - \hat{c}\hat{x}(k)) \quad (6.8a)$$

$$\hat{x}(k) = \begin{bmatrix} \Delta \hat{\theta}_m(k) \\ \hat{\omega}_m(k) \end{bmatrix}, \quad c = (1 \ 0) \quad (6.8b)$$

$$\hat{A} = \begin{bmatrix} 0 & \hat{R} \\ 0 & \hat{P} \end{bmatrix}, \quad \hat{b} = \begin{bmatrix} \hat{S} \\ \hat{Q} \end{bmatrix}, \quad \hat{f} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} \quad (6.8c)$$

次に、実際の電動機とオブザーバのパラメータ誤差がなく、すなわち、

$$\hat{P} = P, \quad \hat{Q} = Q, \quad \hat{R} = R, \quad \hat{S} = S$$

さらに、外乱も無視できると仮定して、デッドビートオブザーバとするために必要なオブザーバゲインを求める。このためには、式(6.8)の  $A-fc$  の固有値をすべてゼロにすればよい。 $A-fc$  の固有多項式は、

$$\det(zI - A + fc) = z^2 + a_1 z + a_0 \quad (6.9)$$

ただし、

$$a_1 = f_1 - P, \quad a_0 = f_2 R - f_1 P \quad (6.10)$$

上記で、 $a_1 = a_0 = 0$  であれば、2つの固有値を 0 にできる。従って、デッドビートオブザーバを得るための  $f$  は次のように求められる。

$$f = \begin{bmatrix} P \\ P^2/R \end{bmatrix} \quad (6.11)$$

式(6.8)、(6.11)から、デッドビートオブザーバを構成する演算式は、次のようになる。

$$\Delta \hat{\theta}_m(k+1) = R \hat{\omega}_m(k) + S u(k) + P [\Delta \theta_m(k) - \Delta \hat{\theta}_m(k)] \quad (6.12a)$$

$$\hat{\omega}_m(k+1) = P \hat{\omega}_m(k) + Q u(k) + \frac{P^2}{R} [\Delta \theta_m(k) - \Delta \hat{\theta}_m(k)] \quad (6.12b)$$

次に、上のオブザーバの計算時間を短くする方法を示す。式(6.12)を計算するためには、乗算を6回行う必要がある。(ただし、係数  $P, Q, R, S$  などはあらかじめ計算して置く。)これに対して、式(6.12)を次のように変形すれば、乗算が5回となり、演算時間を短縮できる。

$$\hat{\omega}_m^*(k) = \hat{\omega}_m(k) + \frac{P}{R} [\Delta \theta_m(k) - \Delta \hat{\theta}_m(k)] \quad (6.13a)$$

$$\Delta \hat{\theta}_m(k+1) = R \hat{\omega}_m^*(k) + S u(k) \quad (6.13b)$$

$$\hat{\omega}_m(k+1) = P \hat{\omega}_m^*(k) + Q u(k) \quad (6.13c)$$

式(6.12)に対応するオブザーバの構成を図3.6-2に、また、式(6.13)に対応するオブザーバの構成を図3.6-3に示す。

上に示したデッドビートオブザーバを用いて電動機の瞬時速度を推定すれば、検出に伴う無駄時間を補償できる。

### § 3-6-2 予測型オブザーバによるマイクロプロセッサの演算時間の補償

ここでは、前項に示したオブザーバを利用して、制御器の演算時間を補償する方法を考える。

演算に伴う無駄時間は、時刻  $k$  におけるデータに基づいて計算された制御入力を時刻  $k+1$  で出力するために生じ、ちょうど1サンプリング周期の遅れとなる。これに対して、オブザーバを使えば実際の時刻  $k+1$  になる以前に、時刻  $k+1$  での状態量の予測値を計算しておくことができる。そこで、この予測値に基づい

て、時刻  $k+1$  に出力すべき制御入力  $u(k+1)$  をあらかじめ計算して置くことができる。このようにすることにより、演算に伴う無駄時間を、オブザーバの予測動作によってキャンセルすることができる。以下では、このように、状態量を予測するオブザーバを予測型オブザーバと呼ぶことにする。

実際に、演算無駄時間が補償されることは、図3.6-4 に示すブロック図の等価変換によって確認できる。同図(a)には、予測型オブザーバと制御無駄時間を隔に示してある。この図(a)は、フィードバック信号の引き出し点を変えて、同図(b)に変換できる。図(b)の  $z$  と制御無駄時間  $z^{-1}$  をキャンセルさせれば図(c)を得る。

以上で、予測型オブザーバを使用することにより、検出および演算に伴う2種類の無駄時間を同時に補償できることが明らかになった。

### § 3-6-3 制御ゲインの計算

ここでは、I-P制御と予測型オブザーバを組み合わせた制御器によりデッドビート制御を行うための制御ゲイン  $K_i, K_p$  を計算する。

制御ゲインを計算するためには、図3.6-4(c)において、オブザーバを介さずに  $\omega_m(k)$  などが直接検出できるものと考えた図3.6-5 の系について考えればよい[2]。さらに、同図から、系の状態方程式は

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{b}\omega_r(k) \quad (6.14a)$$

$$\mathbf{x}(k) = \begin{bmatrix} \omega_m(k) \\ \Delta\theta_0(k) \end{bmatrix} \quad \mathbf{A} = \begin{bmatrix} P-K_pQ & K_iQ \\ SK_p-R & 1-K_iS \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 0 \\ T_s \end{bmatrix} \quad (6.14b)$$

式(6.14)から行列  $\mathbf{A}$  の固有多項式は

$$\det(z\mathbf{I}-\mathbf{A}) = z^2 + a_1z + a_0 \quad (6.15a)$$

$$a_1 = K_pQ + K_iS - P - 1 \quad (6.15b)$$

$$a_0 = P - K_pQ + K_i(QR-PS) \quad (6.15c)$$

式(6.15)で、 $a_1 = a_0 = 0$  とすれば、制御系の固有値を  $z$  平面の原点に配置できる。そのための制御ゲイン  $K_i, K_p$  は、式(6.15)から

$$K_p = \frac{-T_sP^2 + R}{T_sQ(1-P)^2} \quad K_i = \frac{1}{T_sQ} \quad (6.16)$$

と求められる。

以上に導出した式(6.13)の予測型オブザーバと、式(6.16)の制御ゲイン  $K_i, K_p$  を組み合わせた制御系の応答のシミュレーション結果を図3.6-6 に示す。同図から、デッドビート制御が達成されることを確認できる。

### § 3-7 実験による検証

#### § 3-7-1 実験装置の構成

図3.7-1 に、実験装置のハードウェア構成図を示す。表3.7-1 に実験装置のパラメータを示す。

直流電動機およびロータリエンコーダは、第2章で示したものと同じであるが、電力変換器は、高速応答を得るためにトランジスタチョッパに変更した。トランジスタチョッパはパワートランジスタ4個をブリッジ接続したもので、出力電圧(直流電動機の電機子電圧)の制御は、三角波比較によるPWM制御を行っている。PWMのキャリア周波数は、1kHzである。

マイクロプロセッサは、CPU Z-80 (2.5MHz) であり、浮動小数点演算を行うため演算専用LSI AM9511 を付加してある。直流電動機の回転子位置の検出は、ロータリエンコーダの出力パルス(600パルス/回転、デューティ50%)の立ち上がり立ち下がりカウンタにより計数して行う。これは、1/1200回転の分解能で検出するのと同値である。マイクロプロセッサのデータ入出力を含む制御のタイミングはすでに示した図3.5-2 のようになっている。すなわち、プログ

ラマブルタイマ (Z-80CTC) がサンプリング周期ごとに発生するの基準パルスにより、まず、データ入出力用ラッチが動作する。しかる後に、マイクロプロセッサの割り込み処理ルーチンが起動して、順次、入力ラッチから位置データを読み込み、制御演算を行い、出力ラッチへトランジスタチョッパの通流角指令値を出力する。このように、入出力ラッチを設けたことにより、制御系は一定サンプリング周期の基準パルスに同期した離散時間動作を行い、検出データの大きさの違いによりマイクロプロセッサの内部演算処理時間が変化しても制御のタイミングは一定となる。

### § 3-7-2 サンプリング周期の決定

離散時間制御系の設計では、まず、サンプリング周期を決定しなければならない。サンプリング周期の大きさは、制御系の性能に大きな影響を与えるから、その選定は、重要である。以下には、実験システムのサンプリング周期の決定にあたり考慮した事項をまとめておく。

まず、離散時間制御系の安定性の面からは、サンプリング周期が短い方が連続系に近づき好ましいと考えられる。しかるに、電動機速度の有限時間整定制御システムにおいてはサンプリング周期を短くするという事は、それだけ大きな制御入力を電動機に短時間に加えなければならないことになる。電力変換器のピーク容量の面からは、サンプリング周期が長い方が好ましいと思われる。以下には、サンプリング周期と制御入力のピーク値の関係を検討する。

例として、次の一時遅れ系の制御を考える。

$$\frac{d}{dt}x(t) = -x(t) + u(t) \quad (7.1)$$

制御入力として、時刻  $t=0$  大きさ  $u$  のステップ入力に加わったとすれば状態  $x(t)$  は

$$x(t) = e^{-t}x(0) + (1 - e^{-t})u \quad (7.2)$$

ここで、初期値を  $x(0) = 0$  として、時刻  $t = t_r$  で、 $x(t_r) = 1$  となるような制

御入力の大きさを  $u_r$  とすれば

$$1 = \left[ 1 - e^{-t_r} \right] u_r \quad (7.3)$$

式(7.3) から、 $u_r$  と  $t_r$  の関係を計算した結果を図3.7-2 に示す。同図から、 $t_r < 0.1$  では、 $u_r = 1/t_r$  で近似できることがわかる。ここで、式(7.1) の系は、直流ゲインが 1 であるから、時刻  $t_r$  以後  $x(t) = 1$  を保つには、 $u(t) = 1$  ( $t \geq t_r$ ) となればよい。これに対して、整定時間  $t_r$  小さくするには、過渡的に大きな制御入力  $u_r$  を必要とする。逆に、変換器のピーク容量を小さく抑えるためには、到達時間  $t_r$  (上の例では、サンプリング周期に等しい) を大きくすることが望ましい。

さらに、サンプリング周期を小さくできない理由としては、速度検出の誤差を小さくして定常状態での速度制度を向上させたいという要求が上げられる。このことに関連して、以下には、サンプリング周期と速度検出誤差の関係を検討する。

ロータリエンコーダの出力パルスをサンプリング周期のあいだ計数して電動機速度を検出する場合の、速度検出誤差の最大値を求める。ロータリエンコーダの一回転当たりのパルス数を  $N_p$ 、サンプリング周期を  $T_s$  とすれば、1 サンプリング周期間の計数で最大  $1/N_p$  回転の計測誤差があるから、

$$\text{〔最大速度検出誤差〕} = 60/N_p T_s \text{ [rpm]} \quad (7.4)$$

となる。実験システムでは、 $N_p = 1200$  [パルス/回転] であるから、 $T_s = 25$  [ms] とすれば

$$\text{〔最大速度検出誤差〕} = 60/(1200 \times 0.025) = 2 \text{ [rpm]} \quad (7.5)$$

となる。これは、設定速度 500 rpm とする時、検出精度 0.4% であり、精度的にみてもほぼ妥当であると考えられる。またこの時、電動機の時定数が

$T_m = 95$  [ms] であるから、 $T_s/T_m = 0.263$  となり、図3.7-2 から、 $u_r = 4.32$  である。すなわち、定常的に必要な制御入力  $u_r$  の 4.32 倍の制御入力を過渡的に必要とする。これは、実験室レベルでは実現可能な値である。

さらに、マイクロプロセッサの演算能力の点からは、サンプリング周期が 25 ms ならば、その間に必要な演算処理を完了することが、経験的にみて可能である。

以上の事前検討により、以下に示す有限時間整定制御の実験におけるサンプリング周期を 25 ms と決定した。

### § 3-7-3 制御応答の実験結果

実験は、§ 3-5 と § 3-6 に示したむだ時間を補償する次の二つの方法に対しておこなった。

- 補償方法Ⅰ — 線形外挿と制御入力フィードバックの組み合わせ (§ 3-5)
- 保証方法Ⅱ — 予測型オブザーバ (§ 3-6)

まず、補償方法Ⅰの制御系全体のブロック図は、すでに、図3.5-5 に示した通りである。マイクロプロセッサの制御プログラムのフローチャートを図3.7-3 に示す。制御ゲインは、式(5.6a-c) と表3.7-1 のパラメータから次のようになる。

$$K_1 = 1.7686 \quad (7.6a)$$

$$K_p = 2.809 \text{ [V/rad/s]} \quad (7.6b)$$

$$K_i = 61.142 \text{ [V/rad]} \quad (7.6c)$$

設定速度のステップ変化に対する電動機速度とチョッパの通流角指令信号の応答の実験結果を図3.7-4 に示す。電動機速度は、一応は有限時間整定応答しているが、定常状態で振動的になっている。

次に、予測型オブザーバでむだ時間を補償する保証方法Ⅱに対する実験結果を

示す。制御系のブロック図はすでに図3.6-4 に示した通りである。制御プログラムのフローチャートを図3.7-5 に示す。制御ゲインは式(6.16) と表3.7-1 の制御パラメータから次のように決定される。

$$K_p = 1.906 \text{ [V/rad/s]} \quad (7.7a)$$

$$K_i = 61.142 \text{ [V/rad]} \quad (7.7b)$$

$$F = 34.967 \text{ [1/s]} \quad (7.7c)$$

図3.7-6 に、設定速度のステップ変化に対する応答の実験結果を示す。補償方法Ⅰの場合(図3.7-4) に比べて、電動機速度の振動が小さい。この理由は、§ 3-8-1 で検討する。

図3.7-7 に、外乱のステップ変化に対する応答を示す。4 サンプリング周期で設定速度に回復している。ただし、4 サンプリング周期め以降に若干の過渡応答がみられる。この原因は、負荷外乱のステップ変化が完全ではなかったことと、負荷変化のタイミングがサンプリング時刻に同期していなかったためであると思われる。すなわち、負荷変動の実験は、電動機軸に直結した負荷用発電機の電機子負荷抵抗の一部を手動スイッチにより短絡して行った。この方法では、電動機の負荷トルクが完全にはステップ状に変化しないし、また、手動スイッチ投入のタイミングとサンプリング時刻に完全に同期させることは困難である。このために、図3.7-7 の実験結果では、図3.6-6 に示したシミュレーションのような理想的状態とは異なり、若干の過渡応答が残るものと考えられる。

### § 3-8 量子化誤差の検討

本節では、量子化誤差を考慮した電動機速度誤差の解析・検討を行う。これまで、デッドビート制御系の実現方法に焦点を絞るため量子化誤差の影響を無視してきた。しかし、デジタル制御系では、量子化誤差が必然的に発生し、これが定常的な精度に大きな影響を与えるから、本節で検討しておく。

マイクロプロセッサを用いた電動機速度制御系において、速度精度に影響を与える量子化誤差には次の2つがある。

(1) 演算量子化誤差：マイクロプロセッサの内部演算でのデータのけた落ちである。

(2) 検出量子化誤差：電動機位置信号をロータリエンコードを用いてデジタル的に検出する際に発生する。

演算量子化誤差の影響を § 3-8-1 で検討する。§ § 3-8-2 では、検出量子化誤差を考慮したデジタルシミュレーション手法を示す。§ § 3-8-3 には、速度誤差を減少するための一つの方法として可変ゲインオブザーバを提案する。

#### § § 3-8-1 内部演算でのデータのけた落ちを考慮した無駄時間補償方法の比較

補償方法 I により無駄時間を補償した系の実験結果図 3.7-4 において、速度および制御入力が増加した原因を考察する。この制御系に対応するブロック図は図 3.5-5 である。同図において破線で囲んだ部分が演算無駄時間を補償する部分である。ここで、ゲイン  $K_1$  は式(7.6a) で与えられ、大きさが 1 以上である。すなわち、破線で囲んだブロック単独では、不安定である。ただし、例外的に、このブロックへの入力  $u_1(k)$  と出力  $u(k)$  が次式を満たす限りにおいては、不安定モードの影響が外部に現われない。

$$u_1(k) = (K_1 + 1)u(k) \quad (8.1)$$

もし、デジタル制御系の内部演算が無限の精度で行われているとすれば、定常状態において式(8.1) が満たされるように外側のループが動作するように制御系は設計されている。しかし、実際のデジタル制御系では、マイクロプロセッサの内部演算に伴うデータのけた落ちのために、式(8.1) が定常状態でも満たされないことがある。このような場合には、 $u(k)$  の振幅が増大して不安定モードの影響が外部に現われてくる。しかし、その振動が発散するかといえば、そうではない。その理由は、もともと制御系全体としては安定であるように設計されているから、振動の振幅が演算量子化誤差（データのけた落ち）に比べて大きくなると、破線で囲んだブロックの不安定モードによる振動が、閉ループ全体としての制御により抑制されるようになる。そのために、不安定モードによる振動の振

幅が一定値以上には増加しない。以上の考察結果は、図 3.7-4 の実験結果にみられるような制御入力  $u(k)$  の振動の振幅は増加するものの、ある一定値で抑えられているという事実をよく説明する。

以上の検討から、内部演算にデータのけた落ちがあるデジタル制御系では、単独では不安定であるような制御ブロックの使用は、避けたほうがよいといえる。

補償方法 I に比べ、予測型オブザーバを用いた補償方法 II では、制御無駄時間の補償のために不安定ブロックを使用していない。ゆえに、速度制御誤差が少ない。このことは、図 3.7-4 と図 3.7-6 の実験結果でも明らかであるが、さらに、図 3.8-1 に示すデジタルシミュレーション結果によっても確認される。

#### § § 3-8-2 位置検出の量子化誤差を考慮したデジタルシミュレーション手法

ロータリエンコードを用いた位置検出法の検出分解能は、ロータリエンコードのパルス間隔で決定される。図 3.8-2 にロータリエンコードによる回転位置検出方法の説明図を示す。ここで、回転子が時刻  $k$  から  $k+1$  の間に機械角  $\Delta\theta(k)$  だけ回転したとすれば、その期間に検出されるパルス数をカウントすることにより回転角度  $\Delta\theta(k)$  がデジタル的に検出される。回転角度  $\Delta\theta(k)$  は、図 3.8-2 を参考にして次式で与えられる。

$$\Delta\theta_m^*(k) = \left\{ 1 + \text{INT} \left[ \frac{\Delta\theta_m(k) - \Delta\theta_0(k)}{\theta_p} \right] \right\} \theta_p \quad (8.2)$$

ただし、 $\text{INT}[\dots]$  は整数化を示す。また、

$\theta_0(k)$  : 時刻  $k$  での位置誤差  
 $\theta_p$  : ロータリエンコードのパルス間隔

なお、ロータリエンコード 1 回転当たりのパルス数を  $N_p$  とすれば

$$\theta_p = \frac{2\pi}{N_p} \quad [\text{rad}] \quad (8.3)$$

実験装置では、 $N_p = 1200$  である。さらに、次の時刻  $k+1$  での位置誤差  $\theta_0(k+1)$  は、図3.8-2を参考にして、

$$\theta_0(k+1) = \Delta\theta_m^*(k) - [\Delta\theta_m(k) - \theta_0(k)] \quad (8.4)$$

図3.8-3に位置信号のデジタル検出に伴う角度検出誤差を考慮したデジタルシミュレーションのフローチャートを示す。

このデジタルシミュレーションは、すでに示した図3.8-1、および、以下に示す速度誤差の解析に利用されている。

### § 3-8-3 可変ゲインオブザーバによる速度制御誤差の低減

予測型オブザーバを使用した場合の位置検出の量子化誤差による速度誤差の振幅は、図3.8-1から、約 13rpm である。これは、速度の検出誤差 2rpm (cf. 式(7.5)) に比べて大きい。以下では、この速度誤差を、予測型オブザーバのゲイン  $F$  を可変にすることにより低減する方法について検討する。

まず、位置検出の量子化を考慮した制御系のブロック図を図3.8-4に示す。同図の量子化器は、制御系に対するノイズ源として考えることができる。この量子化ノイズが、オブザーバゲイン  $F$  を介してフィードバックされることにより電動機速度に誤差振動が発生する。

ここで、ゲイン  $F$  は、オブザーバ自身を有限時間整定させるために導入されたものであり、その値は、

$$F = \bar{P}/\bar{R} \quad (8.5)$$

である。ゲイン  $F$  を式(8.5)にした場合の図3.8-4の系の設定速度  $\omega_r$ 、および、負荷外乱  $V_a$  のステップ変化に対する応答のシミュレーション結果を図3.8-5(a)に示す。定常状態でも、速度  $\omega_m$ 、および、制御入力  $u$  が振動的である。これに対して、 $F=0$  とした場合のシミュレーション結果が図3.8-5(b)である。 $F=0$  とすることにより、量子化ノイズのフィードバック量が少なくなったから、定常状態での  $\omega_m$ 、 $u$  の振動が減少している。しかるに、負荷外乱への応答が非常に遅くなっている。この理由は、 $F=0$  としたために、オブザーバ

の収束が遅くなったためである。すなわち、 $F=0$  とすると、オブザーバの推定値  $\Delta\theta^0$  と、実際の電動機の回転角  $\Delta\theta_m$  の差をフィードバックするループがなくなるために、オブザーバの収束が遅くなる。

そこで、この対策として、負荷が急変した場合には、 $F=0$  から式(8.5)の値に戻してやる方法が考えられる。(この切り替えは、たとえば、速度  $\omega_m$  の変化により負荷の急変を検出できるから、実現可能である。)このような対策を行った制御系の応答のシミュレーション結果が図3.8-6である。同図より、無負荷状態で負荷が投入された場合 ( $-V_t: 0 \rightarrow 20V$ ) は、うまくゆくが、すでに負荷が入っている状態でさらに負荷が追加された場合には、電動機速度に好ましくない応答が生じる。この理由は、次のように考えられる。

これまで考えてきた予測型オブザーバでは、外乱を無視してきた。これに対して、外乱を考慮したオブザーバの誤差ベクトル  $e$  は既に示した式(6.6)から

$$e(k+1) = [A - fc]e(k) + [(\Delta A - f\Delta c)\hat{x}(k) + \Delta bu(k)] - Dv(k) \quad (8.6)$$

上式で、右辺第3項が負荷外乱によるものである。上式からわかるように、負荷外乱が入るとオブザーバの推定誤差  $e$  は0に収束しない。このことは、図3.8-6で負荷が入った場合に、次式で定義される位置推定誤差  $\Delta\theta_d$

$$\Delta\theta_d = \Delta\theta_m - \Delta\theta^0 \quad (8.7)$$

が、0から次第にずれてゆくことにより確認できる。ここで、もし  $\Delta\theta_d = 0$  でない時に、オブザーバゲイン  $F$  を急変すると、図3.8-4の  $F$  の出力部に不必要な信号が発生する。この信号は、積分および、比例制御ループを介してフィードバックされ、図3.8-6に示したような好ましくない速度応答を誘発する。

そこで、負荷が入った状態でも、位置推定誤差  $\Delta\theta_d$  を0に保つために、オブザーバに新しい制御ブロックを追加することを考える。そのような方法の一つは、オブザーバに負荷外乱の大きさを補償するような制御ブロックを追加してやることである。図3.8-7にそのような対策を施した制御系のブロック図を示す。同図において、破線で囲んだ部分が、負荷補償ブロックである。このブロック

は、オブザーバの位置推定誤差  $\Delta\theta_d$  が 0 になるまで、それを積分し、負荷外乱の大きさを推定するものである。図3.8-7 の系の応答のシミュレーション結果を図3.8-8 に示す。同図から明らかなように、負荷投入時も、オブザーバの位置推定誤差  $\Delta\theta_d$  が 0 になるように制御されており、負荷急変時に好ましくない速度応答も発生していない。さらに、図3.8-5(a) に比較して、定常状態での  $\omega_m$ 、 $u$  の振動も、オブザーバゲイン  $F$  を変化することにより小さくなっていることが確認できる。

以上により、オブザーバゲインを可変にして速度誤差の振動を低減する方法を示すことができた。参考として、図3.8-8 の系の応答をシミュレーションするためのプログラムのフローチャートを図3.8-9 に示しておく。

### § 3-9 本章のまとめ

本章では、電動機速度の過渡的な応答を高速化することを目的とし、設定速度および負荷外乱のステップ状の変化に対して電動機速度をデッドビート応答させ過渡的な応答を高速化するデジタル制御系の実現について論じた。

その成果をまとめると以下ようになる。

(1) 設定速度および未知外乱のステップ状変化に対して、電動機速度のデッドビート応答を達成するための制御系の構造の候補として、PI制御およびI-P制御を取り上げて比較検討し、I-P制御の方が優れていることを示した。すなわち、§ 3-4において、外乱および設定値変更に対して電動機速度の応答にオーバーシュートが生じないようにして有限時間整定させるためには、PI制御則では制御ゲインの切り替えが必要であるのに対して、I-P制御則では制御ゲインの切り替えが不要であることが分かった。

(2) 制御器のデジタル化(時間離散化)に伴う無駄時間を示し、その補償法を提案した。すなわち、§ § 3-5-1で検出無駄時間と演算無駄時間の2つの無駄時間を補償する必要があることを示し、2種類の補償法を§ 3-5および§ 3-6で提案した。§ 3-5では、線形外挿(§ § 3-5-2)と出力フィードバック(§ § 3-5-3)を組み合わせた方法を提案し、これによりデッドビート制御が出来ることをシミュレーションにより

確認した。一方、§ 3-6では、予測型オブザーバを用いた方法を提案し、これによってもデッドビート制御が可能なることをシミュレーションにより確認した。

(3) 実験装置を製作し、デッドビート制御が可能であることを示し、提案する制御法の評価を行った。すなわち、§ 3-7では、製作した実験装置の構成を示し、さきに§ 3-5および§ 3-6に示した2種類の方法によりデッドビート制御が実現できることを実験により確認した。実験結果を比較すると、§ 3-6に示した予測型オブザーバを用いる方法の方が§ 3-5の方法に比べ、電動機速度の振動が少ないことを示した。その理由を検討するため§ § 3-8-1では量子化誤差のあるデジタル制御系を解析し、§ 3-5の方法では不安定ブロックを含むため、§ 3-6の方法に比べ電動機速度の振動が大きくなることが分かった。

(4) 量子化誤差による電動機速度の誤差振幅を低減する方法を提案した。すなわち、§ § 3-8-3では、量子化誤差による電動機速度の誤差振幅を低減するため予測型オブザーバのゲインを可変にする方法を検討し、その有効性をシミュレーションにより確認した。

なお、実験を行った経験によると本章に示したデッドビート制御系は電動機の定数(たとえば、 $K_m$ )の変動に非常に敏感であり、わずかな定数変動があっても制御不能になることが分かった。本制御系のロバスト性を高めるための検討が必要であるが、それについては第4章で述べることにする。

### § 3-10 本章の参考文献

- [1] 今井:「デッドビート制御」、計測と制御、Vol.22, No.7, pp.606-613, 1983.7
- [2] J.O'Reilly: "The Discrete Linear Time Invariant Time-Optimal Control Problem -- An Overview", Automatica, Vol.17, No.2, pp.363-370, 1981.2
- [3] O.A.Sebakhy, T.M. Abdel-Moneim: "State Regulation in Linear Discrete-Time Systems in Minimum Time", IEEE Trans., Vol.AC-24, No., pp.84-88, 1979
- [4] T.Mita: "A Relation between Overshoot and Sampling Period in Sampled Data Feedback Control Systems", IEEE Trans., Vol.AC-25, No., pp.603-604,

- [5] A.Bergan, J. Ragazzini: "Sampled-Data Processing Technique for Feedback Control Systems", AIEE Trans., Vol.73-Pt.2, No., pp.236-247, 1954
- [6] R.E.Kalman, J.E. Bertram: "A Unified Approach to the Theory of Sampling Systems", J. Farankin Inst., No.267, pp.405-436, 1959
- [7] R.E.Kalman, J.E. Bertram: "General Synthesis Procedure for Computer Control of Single and Multiloop Linear Systems", AIEE Trans., Vol.77-Pt.2, pp.602-609, 1959
- [8] R.E.Kalman: "On the General Theory of Control Systems", 1st IFAC Cong., No.4, pp.2020-2030, 1960
- [9] T.Fukao, A. Morita, K. Nishida: "Minimum Time Settling Control Scheme for a Line Commutated Converter Output Current and a System Controller using a Microprocessor", IEEE IECI'80, pp.92-97, 1980.11
- [10] 原島、近藤: 「制御信号の飽和を考慮した電動機速度の最短 時間制御の検討」、第23回自動制御連合講演会、No.3027, 1980.11
- [11] 原島、近藤: 「マイクロプロセッサを用いた電動機速度の最適制御装置の検討」、電気学会半導体電力変換研究会資料、No.SPC-81-7, 1981.2
- [12] 村田: 「インパルス応答を用いた有限整定応答制御装置の設計」、電気学会論文誌、Vol.C-102, No.3, pp.67-74, 1982.3
- [13] 村田、原田、相良: 「電流形インバータ駆動誘導電動機の有限整定パターン速度制御」、電気学会論文誌、Vol.B-102, No.4, pp.197-202, 1982.4
- [14] J.Holtz, U. Schwellenberg: "A New Fast-Response Current Control Scheme for Line Controlled Converter", IEEE IAS ISPC'82, pp.175-183, 1982
- [15] F.Depping, M. Voits: "Microcomputer-Based Parameter-Adaptive Speed Control with Deadbeat Response Algorithm for an Electrical Drive", ETG-Facheberichte 11, pp.127-134, 1982.10
- [16] F.Depping, M. Voits: "Automatic Selection of Control Algorithm for an Electrical Drive with Microprocessor-Based Speed Control", IFAC Symposium "Control in Power Electronics and Electrical Drives", pp.507-514, 1982
- [17] 井上、前納: 「マイコンによる直流電動機の高精度速度制御」、電気学会情報処理研究会資料、No.IP-82-38, 1982.10
- [18] J.T.Tou (中村ほか訳): 「現代制御理論」、コロナ社、pp.150-160
- [19] T.Kailath: "Linear Systems", Prentice-Hall, pp.202-204
- [20] 増淵: 「システム制御」、コロナ社、pp.167-178, 1987.11
- [21] 松田、大前、高橋、大橋: 「電動機瞬時速度のデジタル予測方式」、電気学会 昭59全国大会、No.534, 1984.3

## 第3章の図・表のリスト

§ 3-1 ~ § 3-3

なし

§ 3-4-1

図3.4-1 ロータリエンコーダ付き直流電動機のブロック図

§ 3-4-2

図3.4-2 P I 制御系のブロック図

図3.4-3 P I 制御系のステップ応答 (制御ゲインが式(4.8) のとき)

図3.4-4 P I 制御系のステップ応答 (制御ゲインが式(4.15) のとき)

図3.4-5 I - P 制御系のブロック図

図3.4-6 I - P 制御系のステップ応答

図3.4-7 P I 制御系と I - P 制御系の応答の比較

§ § 3-5-1

図3.5-1 ロータリエンコーダを用いた電動機速度検出の模式図

図3.5-2 マイクロプロセッサへのデータ入出力のタイミングチャート

図3.5-3 無駄時間を考慮にいたした制御系のブロック図

§ § 3-5-2

図3.5-4 線形外挿による瞬時速度推定の説明図

§ § 3-5-3

図3.5-5 補償方法 I による I - P 制御系のブロック図

図3.5-6  $\omega_m = \hat{\omega}_m$  のとき図3.5-5 と等価なブロック図

図3.5-7 補償方法 I によるシステムの制御応答

(シミュレーション結果)

§ § 3-6-1

図3.6-1 同一次元オブザーバの構成

図3.6-2 式(6.12) に対応するオブザーバの構成

図3.6-3 式(6.13) に対応するオブザーバの構成

§ § 3-6-2

図3.6-4 予測型オブザーバによる演算無駄時間の補償を

説明するブロック図

§ § 3-6-3

図3.6-5 制御ゲイン式(6.16) を計算するためのブロック図



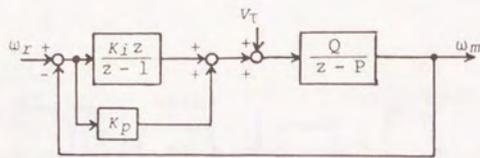


図3.4-2 P I 制御系のブロック図

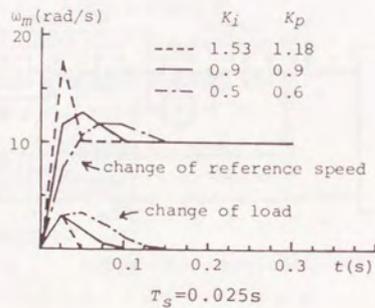


図3.4-3 P I 制御系のステップ応答 (制御ゲインが式(4.8) のとき)

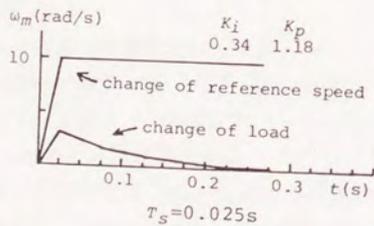


図3.4-4 P I 制御系のステップ応答 (制御ゲインが式(4.15) のとき)

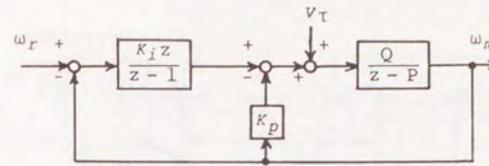


図3.4-5 I - P 制御系のブロック図

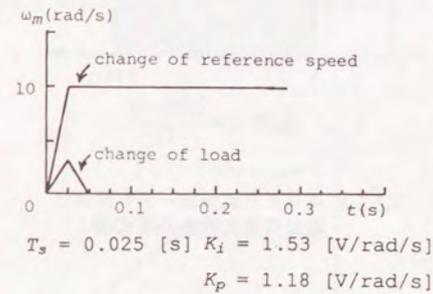
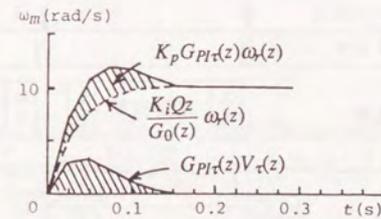


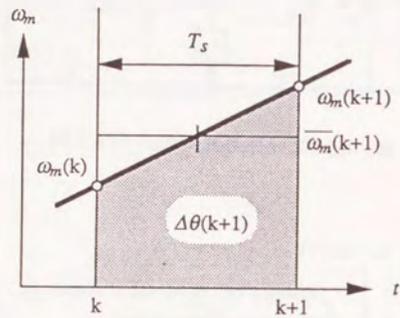
図3.4-6 I - P 制御系のステップ応答



$$\text{PI 制御} \quad \frac{\omega_m(z)}{\omega(z)} = \frac{K_i Q z}{G_0(z)} + K_p G_{PI}(z)$$

$$\text{I-P 制御} \quad \frac{\omega_m(z)}{\omega(z)} = \frac{K_i Q z}{G_0(z)}$$

図3.4-7 P I 制御系と I - P 制御系の応答の比較



$$\bar{\omega}_m(k+1) = \frac{\Delta\theta(k+1)}{T_s}$$

図3.5-1 ロータリエンコーダを用いた電動機速度検出の模式図

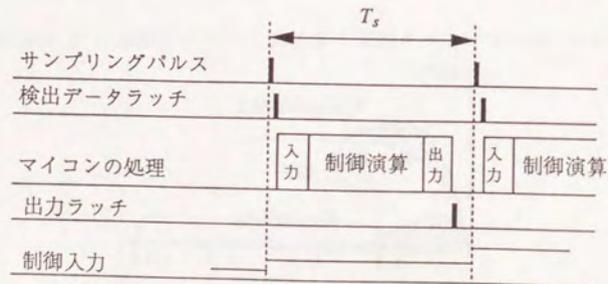


図3.5-2 マイクロプロセッサへのデータ入出力のタイミングチャート

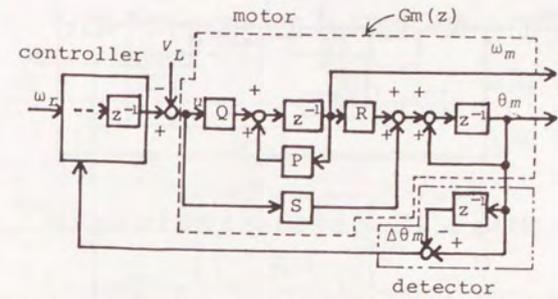


図3.5-3 無駄時間を考慮にいれた制御系のブロック図

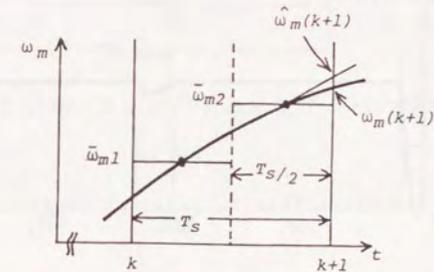


図3.5-4 線形外挿による瞬時速度推定の説明図

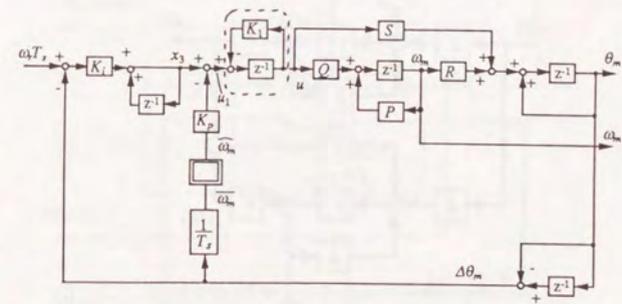


図3.5-5 補償方法IによるI-P制御系のブロック図

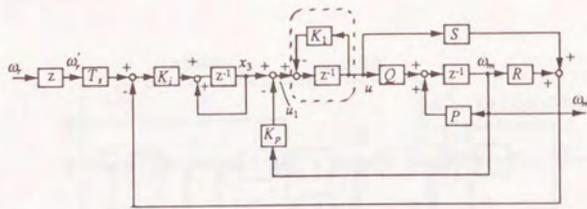


図3.5-6  $\omega_m = \bar{\omega}_m$  のとき図3.5-5 と等価なブロック図

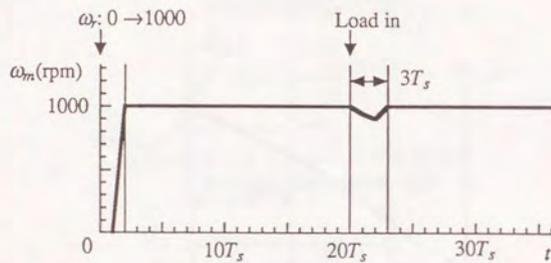


図3.5-7 補償方法Iによるシステムの制御応答 (シミュレーション結果)

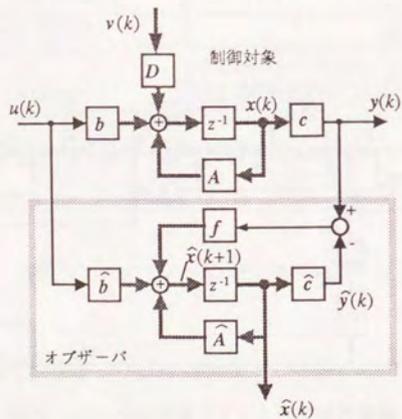


図3.6-1 同一次元オブザーバの構成

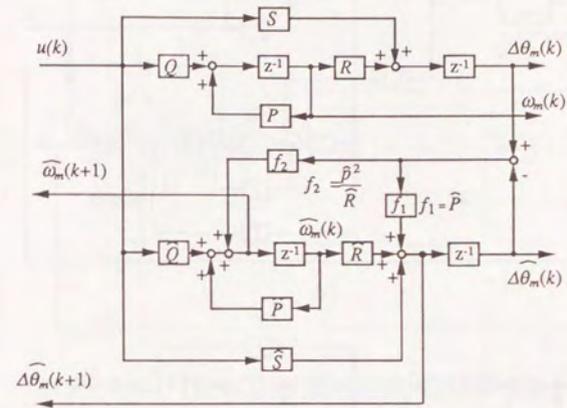


図3.6-2 式(6.12)に対応するオブザーバの構成

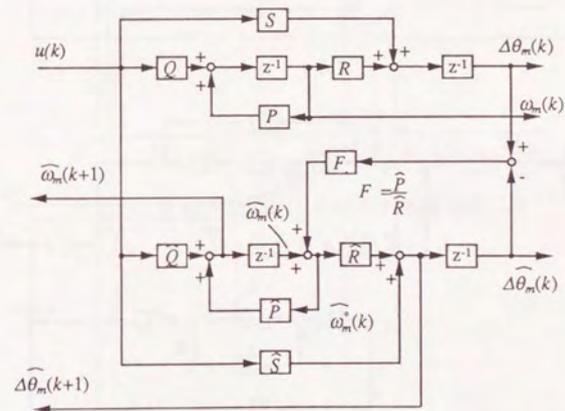
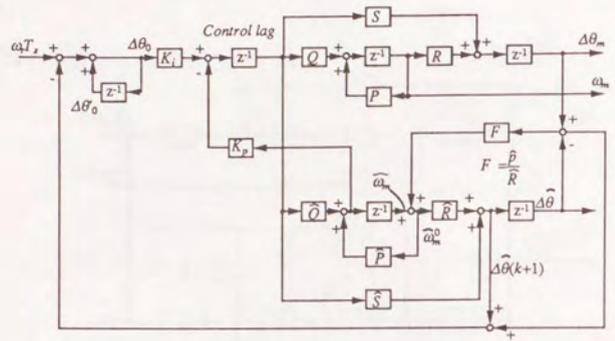
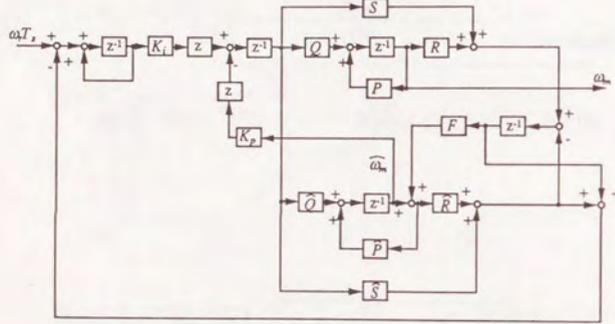


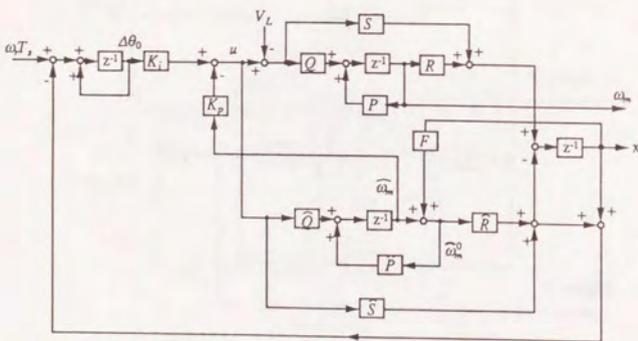
図3.6-3 式(6.13)に対応するオブザーバの構成



(a)



(b)



(c)

図3.6-4 予測型オブザーバによる演算無駄時間の補償を説明するブロック図

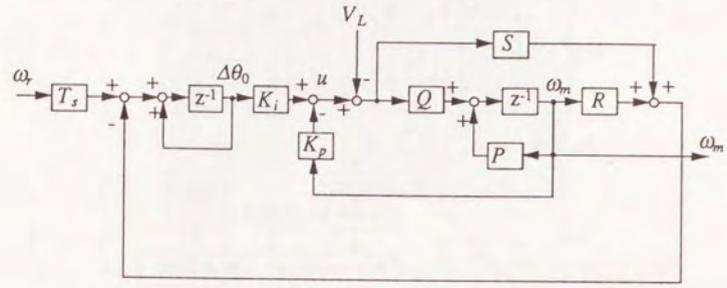


図3.6-5 制御ゲイン式(6.16)を計算するためのブロック図

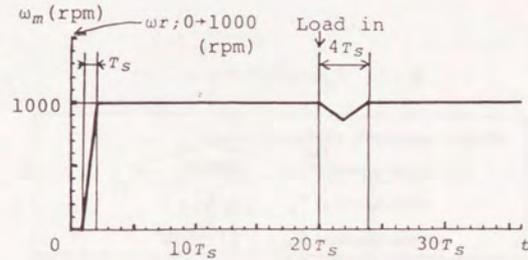


図3.6-6 補償方法IIによるシステムの制御応答 (シミュレーション結果)

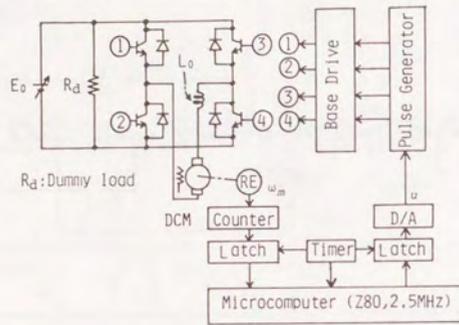


図3.7-1 実験装置の構成図

表3.7-1 実験装置のパラメータ

Motor: separately-excited dc motor	
rated power:	2.2kW
time constant $T_m$ :	0.095 s
gain constant $K_m$ :	27 rpm/V
Control sampling period $T_s$ :	0.025 s
Rotary encoder:	1200 pulse/rev
PWM chopper frequency:	1 kHz
DC choke $L_0$ :	1 mH
DC power supply $E_0$ :	0 - 100 V (50A max)

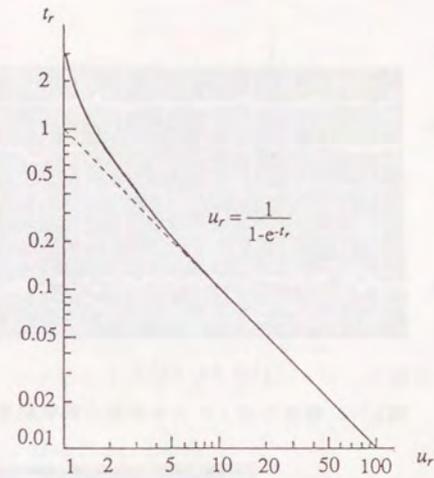


図3.7-2 式(7.3)から求めた  $u_r$  と  $t_r$  の関係

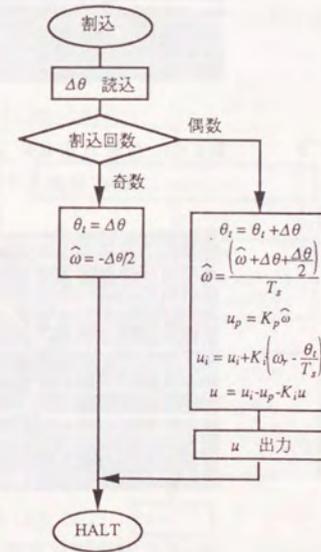


図3.7-3 補償方法Iの制御プログラム

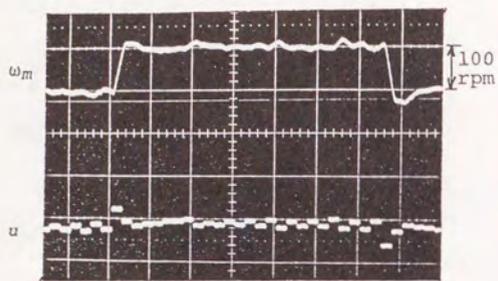


図3.7-4 補償方法 I による制御の実験結果

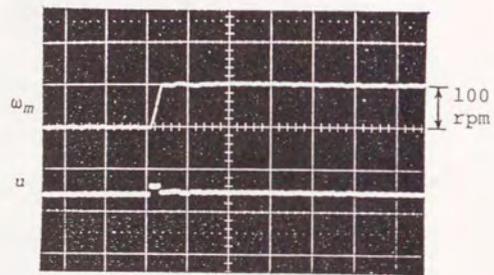


図3.7-6 補償方法 II による制御の実験結果  
(設定速度ステップ変化)

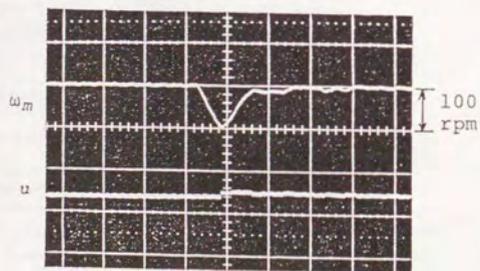


図3.7-7 補償方法 II による制御の実験結果  
(外乱のステップ変化)

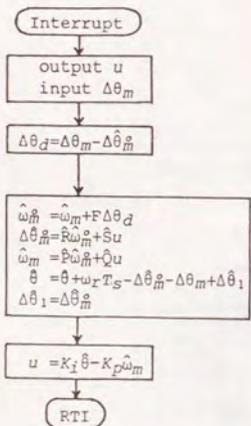


図3.7-5 補償方法 II の制御  
プログラムのフローチャート

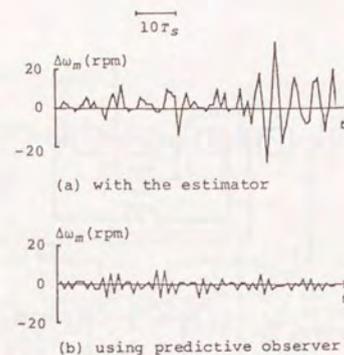


図3.8-1 けた落ちを考慮したシミュレーション結果

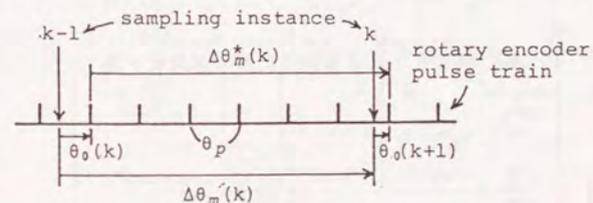


図3.8-2 ロータリエンコーダを用いた回転子位置検出の説明図

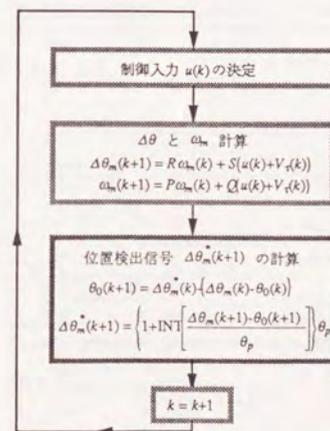


図3.8-3 検出量子化を考慮したシミュレーション  
プログラムのフローチャート

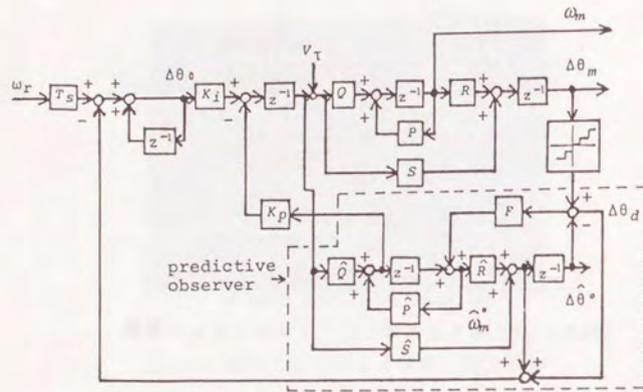
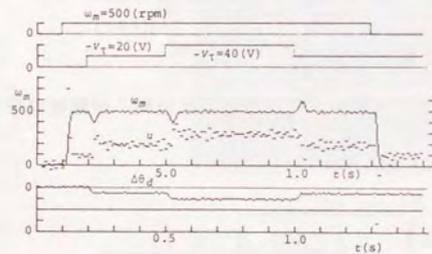
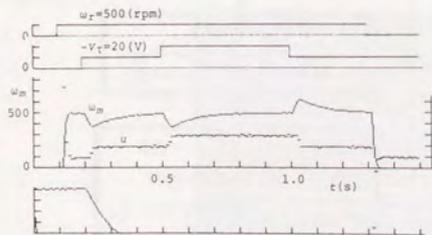


図3.8-4 検出量子化を考慮した系のブロック図



(a)  $F \neq 0$



(b)  $F = 0$

図3.8-5 シミュレーション結果

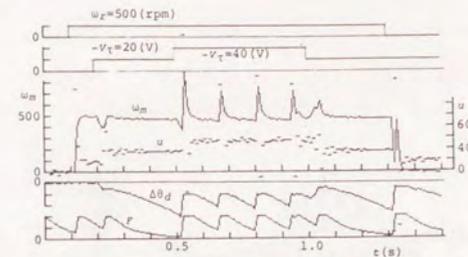


図3.8-6 シミュレーション結果 ( F 可変 )

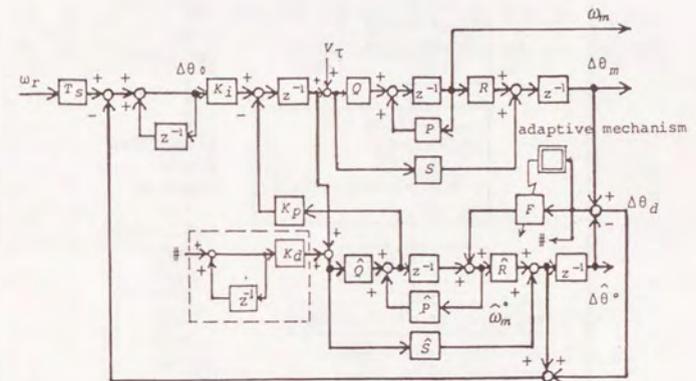


図3.8-7  $\Delta\theta_d = 0$  とするためのブロック図

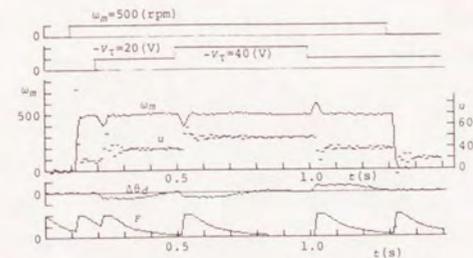


図3.8-8 シミュレーション結果 ( 図3.8-7 の系の応答 )

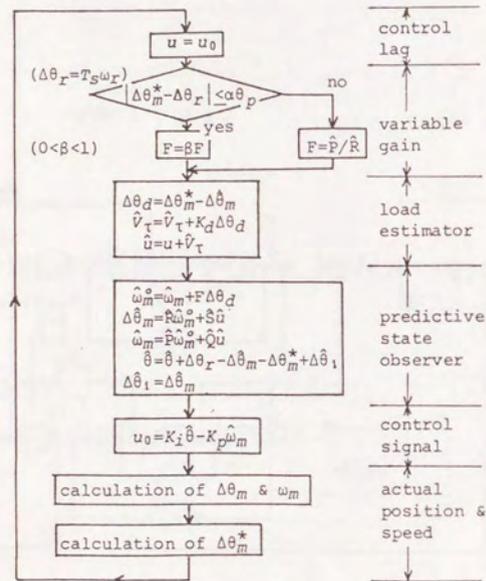


図3.8-9 シミュレーションのためのフローチャート

#### 第4章

### パラメータ感度低減を目的とした サンプリング周期短縮効果の評価

## 第4章 パラメータ感度低減を目的としたサンプリング周期短縮効果の評価

### §4-1 本章の課題と構成

実用的な制御系では、制御対象のパラメータが多少変動しても制御特性が変化しにくいことが要求される。本章では、電動機速度のデジタル制御系のパラメータ変動に対する感度を低減するため、サンプリング周期を短縮することの効果の評価をする。

デジタル制御系では、サンプリング周期が短いほどフィードバックによる修正が頻繁に行われる。従って、サンプリング周期を短くすることによりパラメータ感度を低減できると予想される。このため、本章では、サンプリング周期を短くすることにより、パラメータ感度を小さく抑えられるかどうかを具体的な例により検討する。ただし、パラメータ感度を小さくできたとしても、制御精度あるいは応答速度が劣化したのでは意味がない。

本章の課題は、サンプリング周期を短くすることにより電動機速度のデジタル制御系のパラメータ感度を改善出来るかどうかを明らかにすること、さらに、制御精度および応答速度を悪くすることなくパラメータ感度を改善できるかどうかを明らかにすることである。このように、デジタル制御系の制御精度・応答速度・パラメータ感度の3つを総合的に解析したものは、著者の知る限りでは、これまでにはない。

これらを明らかにするため、本章の構成は次のようになっている。

§4-2から§4-4までは評価を行うための準備である。

§4-2では、パラメータ感度、サンプリング周期、量子化誤差についてのこれまでの研究の動向を紹介する。なお、パラメータ感度・サンプリング周期・量子化誤差を個々に扱った研究はあるが、3つを総合的に論じた研究はこれまでにない。

§4-3および§4-4では、それぞれ、パラメータ感度および量子化誤差の計算方法を準備する。従来からは、伝達関数から求める方法が知られていたが、ここでは、それを状態方程式を与えられれば数値計算できるように整理しなおす。なお、第2章では、量子化誤差によるリミットサイクルを解析し速度制御精度を評価したが、この方法は比較的低次元系にしか適用できない。本章では、高次元系にも適用できる評価方法を検討する。

以上の準備の基に、§4-5および§4-6では、具体的な制御系についてサンプリング周期を短くした場合に、パラメータ感度および量子化による速度制御誤差がどのように変化するかを評価する。この2つの節が、従来にはない新しい結果である。

§4-5では、第3章に示したデッドビート応答をする系のサンプリング周期を短くした場合に、パラメータ感度と速度誤差がどのようになるかを計算する。その結果、サンプリング周期を短くすれば、それに比例して応答が速くなる反面、パラメータ感度が増加し、速度誤差も増加すること示す。

§4-6では、デッドビート応答という条件をはずし、応答時間を一定するようにしてサンプリング周期を短くしたときのパラメータ感度および速度誤差を計算し、実験結果により制御応答の比較を行う。その結果、サンプリング周期を短くしても応答時間を一定に保つようになれば、パラメータ感度と速度誤差の大きさを小さく抑えることが出来ることを明らかにする。

§4-7では、本章で得られた結果をまとめる。

§4-8には、本章の参考文献を示す。

### §4-2 研究の動向

ここでは、フィードバックの効果、制御系の感度、サンプリング周期、および、量子化誤差の評価の各々についての研究の動向を紹介する。

#### §§4-2-1 制御対象の曖昧さとフィードバック

実用的な制御系は、制御対象の曖昧さを許容できるものでなければならない。すなわち、制御対象のパラメータが多少変動したり、また、外乱が入っても、制御特性が変化しにくいことが要求される。

制御系の設計は制御対象の数学的モデルに基づいて行われることが多いが、その数学的モデルは必ずしも正確ではなく、数学的モデルの不完全さを許容して所要の制御目的を達成するように制御系を構成する必要がある。数学的モデルが不確かさを含む原因としては、次のものが考えられている[1]。

- (1) 制御対象のパラメータの測定が正確でない。
- (2) 運転条件の変化により制御対象の特性が変動する。

- (3) 外乱や雑音に関する事前データの不完全さ。
- (4) 設計を容易にするため、数学モデルの低次元化あるいは簡略化を行う。
- (5) 制御系の非線形性の影響。

これらのモデルの不完全さの問題は、単に同定技術を駆使してモデルを精密にするだけでは解決されない。その理由は、たとえ精度のよいモデルが得られたとしても、それが複雑であり過ぎれば、制御系をいたずらに複雑化するだけで、制御性能はそれほど改善されないという結果に陥るからである。むしろ、数学的モデルの不完全さがある程度許容し、所要の制御目的を達成するように制御系を構成する必要がある。

制御対象の曖昧さを許容するように制御系を設計するという考え方自身は、特別に新しいものではない。たとえば、文献[1]には、次の指摘がされている。「モデルの不確かさを克服することは、昔から制御の基本テーマの一つであった。もともとフィードバック自身、ホロヴィッツ[2]の言葉を借りれば、“無知に対処する手段”の一つであり、ボードのフィードバック理論[3]にはフィードバックのこの側面が強調されている。」

制御対象の曖昧さを許容する設計法の例としては、文献[4]、[5]がある。これらにおいては、制御系の応答が参照モデルに一致するように設計が行われる。特に、その参照モデルそのものが、適当なところで高次項を打ち切っても制御系の応答があまり変化しないように選ばれているところに特徴がある。

フィードバック制御では、制御結果（制御誤差）をみてから対策（制御入力）の修正を行う。これにより、制御対象に関する事前知識の曖昧さを許容しその影響が出力に現れにくくしている。つまり、制御対象に外乱が入ったりパラメータ変動があっても、制御系の定常的および過渡的な制御特性が変化しにくくしている。その反面、フィードバック制御系では対策が後手に回ることになり、安定性の問題が生じる。

ともあれ、「フィードバックは制御対象に関する知識の不完全さに対処するほとんどもただ一つの方策であり、同時に知識の不完全さを補い得るきわめて有効な方策でもある。」[6]

以上のように、フィードバックの主要な効果は、制御対象の曖昧さが制御系全体の特性に及ぼす影響を抑制することであると考えられている。

#### § 4-2-2 制御系の感度

ここでは、文献[7]に基づいて制御系の感度について簡単に紹介した後で、研究の動向を述べる。

制御系の感度とは、制御対象のに関する知識の曖昧さが制御系の特性に与える影響を測るための尺度である[7]。すなわち、各種の曖昧さを制御系に対する外乱と総称すれば、外乱  $\Delta D$  が加えられたときの制御系の特性変動を  $\Delta C$  とすれば、制御系の感度  $S$  は、次のように定義される。

$$S = \lim_{\Delta D \rightarrow 0} \frac{\Delta C}{\Delta D} \quad (2.1)$$

感度  $S$  が小さいほど、外乱が制御特性に与える影響が小さいわけであるから、フィードバック制御として望ましいといえる。ここで、 $\Delta D$  および  $\Delta C$  としてどのようなものに着目するかによって、次のように各種の感度が考えられる。

- <1> 比較感度（パラメータ感度）：制御対象のパラメータの変化率に対する制御系の伝達特性の変化率の比として定義される。
- <2> 動的感度：パラメータ、初期値、システム方程式の階数などの変化に対して、制御系の解軌道のずれを問題とするもの。
- <3> 極感度：パラメータ変動に対する、制御系の極の変化を問題とするもの。
- <4> 評価関数感度：最適制御系においてパラメータ変動に対する評価関数の値の変化を問題とするもの。

これらの詳細な定義および計算のしかたに関しては、文献[7]を参照されたい。

つぎに、制御系の感度についての研究の動向を述べる。制御系の感度は、古典フィードバック理論を正当化するために Bode[3]により導入されたことを受けて、Horowitz[2]は、感度の概念を積極的に導入してフィードバック制御系の設計論を展開した。すなわち、まず、パラメータ変動に対する感度が制御系の一巡伝達関数（開ループ伝達特性）と密接な関係にあることを示した。従って、開ループ伝達特性と感度の両方の設計仕様を満たすためには、制御系の構造をいわゆる2自由度制御系にしなければならないことを明らかにした。

この感度の概念を多変数系に拡張し、開ループ系および閉ループ系のパラメータ変動の影響を直接比較できるようにした、いわゆる、比較感度を定義したのは、Perkins および Cluz である [8, 9]。

感度に関する研究を年代的にみると、1960年代から1970年代初期に多くの発表がなされた [7, 10-16]。特に、文献サーベイとしては、[10] は1964年までの157件の論文を、また、[16] は1965年から1970年までの134件の論文を、リストアップしている。

このように、この時期に、制御系の感度が注目された理由は、最適制御ならびに適応制御との関連においてであるとされている。最適制御と感度の関連を、井上[14] は、次のように説明している。『最適制御論では、制御対象の特性が厳密に正確に数学モデル化されて与えられるものと仮定し、そのモデルに対して最適制御を求めようとするものである。ところが、我々はこの仮定が実際にそぐわないことを知っている。従って、最適制御理論が発展し始めた初期の段階でこのことが指摘され、いわゆる評価関数感度解析に関する研究が始められた。・・・途中略・・・これに対して、ほぼ時を同じくして、解析という立場からではなく、もっと積極的に最適制御をシンセシスする最初の段階で感度を考慮に入れるべきだという考え方（途中略）が現れ、感度を考慮した最適制御のシンセシスという新しいジャンルが確立されるようになった。』

一方、適応制御と感度との関連は、文献[10]の中で、次のように説明されている。『Interest in the problems connected with the study of parameter variation arose with the appearance of adaptive systems, more precisely, as a reaction to their appearance. The question naturally was raised as to what extent were the complications in the system cause by the introduction of adaptation justified, and as to what were the main capabilities of networks of the usual type in the presence of parameter variations.』

しかしながら、感度を用いた適応制御の研究はあまり成功しなかったように思われる。その理由は、文献[16]のなかの次の文章に説明されている。『・・・自由討論の部では、まず Tsytkin 教授から、・・・感度研究の一つの側面に関して悲観的な見解が出された。すなわち、感度法によって適応制御系を構成する場合、感度モデルをつくる必要があるが、この感度モデルは制御対象に構造を知らなければ作ることができない。一方、問題は構造に関する知識なしでよい制御を

行うことである。また、感度モデルは制御対象と同様の構造を持つから、複雑な制御対象の場合にはもう一つの複雑なシステムを作らなければならないことになる。これは大変に費用のかかることではなかろうか。したがって、結論として、感度を用いずに適応制御系を構成すべきではなかろうか。』

さて、制御系の感度を小さく抑えるには、制御系の構造をどのようにすればよいのであろうか。この問いに対する一つの答えとしては、受動的適応制御系と呼ばれる制御系構成法が提案されている[17-22]。この制御系では、制御系をあらかじめ特別な構造にしておくことにより、能動的調整機構を持たなくてもパラメータ感度を抑制できるようにしてある。受動的適応制御系は、制御系構造が比較的簡単で、頻繁なパラメータ変動に対しても適応調整のための時間遅れがなく、補償効果を得られることを特長とする。この受動的適応制御系の構造を電動機の位置制御あるいは速度制御に応用して、パラメータ変動あるいは負荷外乱に対して強い制御系を実現したという報告もある[23, 24]。

以上は主に連続時間制御系の感度に関するものであった。以下には、離散時間制御（デジタル制御）の感度に関する研究を簡単に紹介する。

デジタル制御系ではサンプリング周期毎にしか制御入力の変更が行われなから、連続時間制御系に比べて、パラメータ変動の影響を抑制するというフィードバックの効果が制限される。デジタル制御系のパラメータ感度のある値以下に小さくはできない原因は、デジタル制御器の無駄時間であるといわれている[25, 26]。このことは、次のように説明できる。すなわち、デジタル制御器ではデータを取り込んでから演算処理をしてその結果を出力するまでに時間的な遅れがある。たとえば、データの入力と出力を同時刻に行う場合には、少なくとも1サンプリング周期の無駄時間が存在する。制御系の無駄時間は、余分な位相遅れをもたらす。このため、閉ループ系を安定に保つために開ループゲインの大きさが制限される。一方、開ループゲインが大きいほど、パラメータ感度は小さくできるという関係がある。結局、デジタル制御系の無駄時間により開ループゲインが制限されるため、パラメータ感度を小さくできないことになる。

デジタル制御系のパラメータ感度を抑える方法としては、次の2つがある。すなわち、サンプリング周期を短くするもの[25]と、連続系マイナーループによるもの[26]である。前者は、無駄時間そのものが小さくできるものであるから、パラメータ感度を小さく抑えることができると考えられている。一方、後者は、デ

デジタル制御ループの内側に連続系のマイナーループを設ける。これにより、プラントとマイナーループからなる連続系部分の位相遅れを小さくする。すると、デジタル補償器により低周波数領域での開ループゲインを大きくできるようになるから、パラメータ感度を小さく抑えることができる。

#### § 4-2-3 サンプルング周期

デジタル制御系の設計においてサンプルング周期の選定は重要であるが、現在のところ、一般的なサンプルング周期決定手法は確立されていないようである。ケースバイケースにより、これまでの経験よるか、あるいは、シュミレーションにより確認しながらサンプルング周期を決定している状況にある。

まず、サンプルング周期選定に当たって考慮にいれておくべき項目を、文献[27]を参考にまとめておく。

- ① 制御系に要求される性能、特に、応答仕様
- ② 制御対象の動特性、および、制御対象のモデルが正確である周波数範囲
- ③ 外乱の性質、特に、その周波数特性
- ④ センサと検出ノイズなどの特性
- ⑤ 操作器の動特性および操作量の飽和
- ⑥ 制御方式の種類（処理内容の複雑さ）
- ⑦ コントローラの処理能力（演算速度、ビット数、メモリ容量）

たとえば、①の制御系に要求される応答性に着目した場合には、閉ループ制御系の周波数バンド幅の10倍程度のサンプルング周波数にするればよいとされている[28-30]。また、②および③の制御対象および外乱の動特性に着目した場合も、これらの主要時定数の1/10程度のサンプルング周期にすればよいとされている[31]。また、サンプルング周期を大きく選び過ぎて失敗する例が多いから、可能な限り短く選ぶ方が無難であるといわれている[32]。

必要なサンプルング周期の長さは、デジタル制御器の設計法にも関係する。連続系をプラントとするデジタル制御系の設計法としては、大別して2通りのものがある[33]。第一の方法は、連続なプラントの入力に0次ホールドを付加したものを適当なサンプルング周期で離散時間モデル化し、これに基づいてディジ

タル制御系を設計するものである。第二の方法は、まず連続な制御系を設計し、その制御器の部分を離散時間化するものである。離散化の方法としては、インパルス応答不変近似、ホールド等価近似、後退差分近似、双一次近似、周波数歪を補正した双一次変換などがある。（[29]の pp.52-58, [30]の pp.38-69, および[31]を参照のこと。）上記の二つの方法を比較すれば、第1の方法のほうが、比較的長いサンプルング周期でも安定な制御系を設計できる。また、デッドビート制御は第一の方法でしか設計できない。

サンプルング周期に関連しては、デジタル制御器での演算処理に要する時間が制御系の無駄時間として作用し、安定性を劣化させるという問題がある。この無駄時間の長さが、閉ループ制御系に要求される応答時間に比べて無視できなくなると、制御系の過渡応答および安定性に大きな影響を及ぼす。この無駄時間に対しては、第3章に示したようにオブザーバによる予測が有効な対策である。予測値を用いて無駄時間を補償するという考え方自体は、特別に新しいものではなく、1951年にKalmanにより、シュミレータを用いて補償を行うという方法が提案されている[34]。しかし、無駄時間問題は、マイクロプロセッサを用いてより高速な制御系を設計しようとすることが多くなるに従い、再び、注目されるようになった[35]。

#### § 4-2-4 量子化誤差

デジタル制御器の内部で扱えるデータの語長は有限であるから、必然的に量子化誤差の問題が発生する。デジタル制御系の量子化誤差の問題は、量子化されるデータの違により、次の2つに分類できる。

- (1) 信号の量子化誤差：信号入力のためのA/D変換、および、マイクロプロセッサ内部での計算結果の丸めなどにより生じる。これは、制御系の精度を劣化させる。
- (2) 制御ゲインの量子化誤差：設計計算で求められた制御ゲインの値を、有限語調で表現するために生じる。これは、制御系の伝達特性を変化させる。制御ゲインの誤差は、さきに示したパラメータ変化として扱うこともできる。

デジタル系における量子化の問題は、制御の分野よりも、デジタル信号処

理の分野、特にデジタルフィルタに関連して、多くの研究がなされている[36-42]。また、最近デジタル制御関係の成書が多く出版されているが、量子化問題を扱っているものは少ない。比較的ページ数を割いているものとしては、[43-45]がある。特に[45]は、量子化問題について80ページ以上をあてており、サーベイも行き届いている。同書によれば、信号の量子化に対する接近法として次の3つがある。

- (A) 量子化誤差の影響(精度の劣化)の大きさを評価するもの
  - (A1) 確定的方法
  - (A2) 統計的方法
- (B) 量子化誤差に伴う精度の劣化を抑制するもの
- (C) 量子化によるリミットサイクルが発生しないようにするもの

但し、(C)については、デジタルフィルタなどには有効であろうが、フィードバック制御系に対しては、制約条件がきびしすぎて適用が困難であろうとしている[45]。そこで、(A)と(B)に関する文献を、著者なりにまとめておく。

- (A) 量子化誤差の影響の大きさを評価するもの
  - (A1) 確定的方法

信号の量子化に起因するリミットサイクルを正確に評価する最も有力な方法は、デジタルシミュレーションである。比較的単純なリミットサイクルに対しては、§2-6に示したようにモード分解して解析的に求めることができる。これについては、文献[46]が参考になる。また、図式解法[47]もあるが、最近では、パーソナルコンピュータによりシミュレーションするほうが効率的であろう。

一方、リミットサイクルの大きさだけを評価しようとするものもある。すなわち、BI-BO安定に基づく最悪ケースの評価[48]、 $z$ 変換の最終値定理を使った定常状態の誤差評価[49]、および、リアプノフの直接法による評価[50,51,37]などである。これらのうち、最初のもの[48]が、数値計算に関しては最も安定な評価が行えるようである。(これに関しては後の節で詳しく示す。)これに対して、 $z$ 変換最終値定理によるものは、リミットサイクルのように誤差が定常値に落ち着かない場合には、必ずしも誤差振幅の上限を与えない([43]のp.176脚注参照)。また、リアプノフの方法では、シャープなリアプノフ関数を見つける一般的な方

法がないという難点がある([45]のp.136参照)。

#### (A2) 統計的方法

この方法では、量子化誤差を白色ノイズと仮定する。このとき、 $i$ 番目の量子化器の影響による出力の分散は、次式で求められる。

$$\sigma_i^2 = \frac{1}{2\pi f} \frac{\Delta_i^2}{12} \oint H_i(z) H_i(z^{-1}) z^{-1} dt \quad (2.2)$$

ただし、

- $\Delta_i$  :  $i$ 番目の量子化器の量子化幅
- $H_i(z)$  :  $i$ 番目の量子化器から出力までのパルス伝達関数
- $\sigma_i^2$  :  $i$ 番目の量子化器の影響による出力の分散

式(2.2)の計算は、留数計算よってもできるが、逐次計算法[52]の方が簡便である。あるいは、パルス伝達関数を經由しないで、状態方程式から直接時間領域で計算することもできる([29]のpp.169-172を参照：後の節で詳しく示す)。

- (B) リミットサイクルの振幅を抑制するもの

これは、ディザ信号を加えることにより、量子化誤差を白色化して、リミットサイクルの最大振幅を抑制するものである[53-55]。

以上、量子化問題の扱い方を種々述べたが、これらの内(A1)に属するBI-BOに基づく最悪ケースの評価、および、(A2)に属する白色ノイズモデルに基づく時間領域での評価については§4-4で詳しく述べる。

### § 4-3 パラメータ感度とその評価方法

本節では、デジタル制御系のパラメータ感度を数値計算する方法を紹介する。まず、§ 4-3-1では、Horowitz [2] に従ってパラメータ感度の計算式を開ループ伝達関数を使って示す。§ 4-3-2では、パラメータ感度と外乱・検出ノイズ抑制効果には密接な関係があり、片方を良くすると他方が悪くなるというよく知られた関係を説明する。ところで、解析すべき制御系が多重ループをもつと § 4-3-1 に示した開ループ伝達関数の解析式を求めることが煩雑となる。そこで、§ 4-3-3では、ファデーブの方法を利用してパラメータ感度を数値計算する方法を示す。

#### § 4-3-1 パラメータ変動に対する伝達関数の感度

ここでは、Horowitz (文献[2]の第3章を参照) に従って、パラメータ感度を導入する。以下には、簡単のため1入力1出力系について示すが、多入力多出力系への拡張もされている[1], [56]。

制御系の中のある要素  $k$  が  $\Delta k$  だけ変動したときに、制御系の入出力特性  $T$  が  $\Delta T$  だけ変化したとする。このとき、パラメータ感度  $S$  は、次式で定義される。

$$S = \frac{\Delta T / (T + \Delta T)}{\Delta k / (k + \Delta k)} \quad (3.1)$$

パラメータ感度  $S$  が小さいということは、制御系の中の要素の変動にもかかわらず入出力特性が変化が少ないことを意味し、制御系として望ましい特性である。

パラメータ感度の具体的な計算式を以下に示す。図4.3-1(a) に示す制御系において、要素  $k$  が  $\Delta k$  だけ変動した場合の入力  $I$  から出力  $O$  への伝達特性の変化を計算する。このために、まづ、次の4つの伝達特性を定義しておく。

- $t_{oi}$  :  $k=0$  のときの  $I \rightarrow O$  への伝達特性
- $t_{os}$  :  $k=0$  のときの  $S \rightarrow O$  への伝達特性
- $t_{ci}$  :  $k=0$  のときの  $I \rightarrow C$  への伝達特性
- $t_{cs}$  :  $k=0$  のときの  $S \rightarrow O$  への伝達特性

このとき、制御系の信号線図は、図4.3-1(b) のようになる。同図から、入力  $I$  から出力  $O$  への伝達特性  $T$  は、

$$T = t_{oi} + \frac{k t_{ci} t_{os}}{1 - k t_{cs}} \quad (3.2)$$

さて、 $k$  が  $k_0$  から  $k_f$  に変動したとき、 $T$  が  $T_0$  から  $T_f$  に変化したとする。すなわち、

$$k_f = k_0 + \Delta k \quad (3.3)$$

$$T_f = T_0 + \Delta T \quad (3.4)$$

ただし、 $t_{oi}$ 、 $t_{os}$ 、 $t_{ci}$ 、 $t_{cs}$  は、 $k$  が変化しても変わらないものと仮定する。このとき、 $T_0$  および  $T_f$  は、式(3.2)を参考にして、それぞれ、

$$T_0 = t_{oi} + \frac{k_0 t_{ci} t_{os}}{1 - k_0 t_{cs}} \quad (3.5)$$

$$T_f = t_{oi} + \frac{k_f t_{ci} t_{os}}{1 - k_f t_{cs}} \quad (3.6)$$

一方、式(3.4)から、

$$\frac{\Delta T}{T_f} = \frac{T_f - T_0}{T_f} \quad (3.7)$$

式(3.7)の右辺に、式(3.5)と(3.6)を代入し、式(3.3)を用いて整理すれば、

$$\frac{\Delta T}{T_f} = \frac{1}{1 - k_0 t_{cs}} \cdot \frac{T_f - t_{oi}}{T_f} \cdot \frac{\Delta k}{k_f} \quad (3.8)$$

式(3.1)と(3.8)から、 $k$  の変動に対するパラメータ感度  $S$  は、

$$S = \frac{1}{1 - k_0 t_{cs}} \left(1 - \frac{t_{oi}}{T_f}\right) \quad (3.9)$$

ここで、要素  $k$  の回りの開ループ伝達特性は、

$$L_0 = -k_0 t_{cs} \quad (3.10)$$

であるから、これを使って式(3.9)は次のように変形される。

$$S = \frac{1}{1 + L_0} \left(1 - \frac{t_{oi}}{T_f}\right) \quad (3.11)$$

さらに、 $t_{oi} = 0$  となることが多く、その場合には、式(3.11)が次のように簡単化される。

$$S = \frac{1}{1 + L_0} \quad (3.12)$$

以上により、パラメータ感度の計算式は、式(3.9)、(3.11)、(3.12) などで与えられる。

次に、パラメータ感度を小さく抑えるための条件について検討する。それには、式(3.11)または(3.12)から、 $|1 + L_0|$  を大きくすれば十分である。簡単のため連続系を例にとって、パラメータ感度の大きさが 1 以下となる条件を説明する。連続系の開ループ伝達関数  $L_0(s)$  のナイキスト軌跡は、図4.3-2 のようになる。

図において、 $|1 + L_0(j\omega)|$  は点  $(-1, 0)$  から  $L_0(j\omega)$  へのベクトルの大きさである。したがって、式(3.12)で与えられるパラメータ感度が 1 以下となる条件は、図4.3-2 において、 $L_0(j\omega)$  のナイキスト軌跡が点  $(-1, 0)$  を中心とする単位円の外側にあることである。同様に、離散系においても、パラメータ感度の大きさが 1 以下となる条件は、開ループ伝達関数のナイキスト軌跡が点  $(-1, 0)$  を中心とする単位円の外側にあることである。なお、最適レギュレータは、円条件により全周波数領域で上の条件を満たすことが知られており、パラメータ感度の大きさが 1 以下である。

#### § 4-3-2 パラメータ感度と外乱および検出ノイズ抑制効果

ここでは、パラメータ感度と外乱および検出ノイズ抑制効果の密接な関係について説明しておく。図4.3-3 に示す制御系において、設定値  $r$ 、外乱  $d$ 、検出ノイズ  $n$  の各々から出力  $y$  への伝達関数は、それぞれ、

$$G_r = \frac{y}{r} = \frac{PC}{1+L} \quad (3.13)$$

$$G_d = \frac{y}{d} = \frac{P}{1+L} \quad (3.14)$$

$$G_n = \frac{y}{n} = \frac{L}{1+L} \quad (3.15)$$

ただし、

$$L = PCH \quad (3.16)$$

一方、プラント  $P$  の変動に対するパラメータ感度は、式(3.12)より、

$$S = \frac{1}{1+L} \quad (3.17)$$

以上に式から、次のことがわかる。

まづ、パラメータ感度と外乱抑圧特性の関係は、式(3.14)と(3.17)から、

$$S = \frac{G_d}{P} \quad (3.18)$$

この式は、外乱  $d$  が出力  $y$  に与える影響を小さくすること、すなわち、 $G_d$  を小さくするとパラメータ感度  $S$  も小さくなることを意味する。

次に、パラメータ感度と検出ノイズ抑制特性の関係は、式(3.15)と(3.17)から、

$$S + G_n = 1 \quad (3.19)$$

この式から、パラメータ感度  $S$  を小さくすると、逆に  $G_n$  が 1 に近づくことになる。このことは、検出ノイズが低減されずに出力に現れることを意味する。従って、パラメータ感度の仕様と検出ノイズ抑制に関する仕様とは相反するものであることがわかる。従って、全周波数領域にわたり、両方の仕様を満足することができない場合がある。このような場合には、周波数帯域を分割して満足させるべき仕様を変えることで対応する。たとえば、低周波数領域では、外乱抑圧特性（パラメータ感度）が重視されるのでこれを満足するようにし、一方、高周波数領域では検出ノイズが大きくなりやすいから検出ノイズ抑制に関する仕様を満足するように設計を行う。

#### § 4-3-3 パラメータ感度の数値計算法

パラメータ感度の計算式は、すでに § 4-3-1 の式(3.11)または(3.12)として示した。しかし、解析すべき制御系が多重ループになると、開ループ伝達関数  $L_0$  などの解析式を求めることがかなり煩雑となる。そこで、以下には、パラメータ感度の数値計算法を示す。

まづ、ファデーブの方法を利用した伝達関数の数値計算法を準備する。次の状態方程式で与えられる時間離散システムを考える。

$$x(k+1) = Ax(k) + bu(k) \quad (3.20a)$$

$$y(k) = cx(k) \quad (3.20b)$$

このシステムの入力  $u$  から出力  $y$  への伝達関数は、

$$G(z) = c \frac{\text{adj}(zI-A)}{\det(zI-A)} b \quad (3.21)$$

ここで、 $A$  次数を  $n$  とすれば、

$$\det(zI-A) = z^n + \alpha_{n-1} z^{n-1} + \dots + \alpha_1 z + \alpha_0 \quad (3.22)$$

$$\text{adj}(zI-A) = B_{n-1} z^{n-1} + \dots + B_1 z + B_0 \quad (3.23)$$

これらの式の係数、 $\alpha_i$ 、 $B_i$  はファデーブの方法により計算できる。

【ファデーブの方法】

for  $i = 1$  to  $n$

$$X = AB_{n-i} : (B_{n-1} = I)$$

$$\alpha_{n-1} = -\text{trace}(X)/i \quad (3.24)$$

$$B_{n-i-1} = X + \alpha_{n-i} I$$

next  $i$

ただし、 $i=n$  のとき  $B_{-1}$  は 0 となる（これは計算のチェックに利用できる）。

以上の準備のもとに、図4.3-4 に示す系のパラメータ感度の数値計算法を示す。同図において、 $k$  が変動する要素である。 $k$  の変動に対する入出力伝達関数  $T$  のパラメータ感度は、式(3.11)で与えられるから、

$$S = \frac{1}{1+L_0} \left(1 - \frac{t_0 i}{T_f}\right) \quad (3.25)$$

上式の  $L_0$ ,  $t_{oi}$ ,  $T_f$  はそれぞれ次のようにして計算できる。

#### $L_0$ の計算

図4.3-4 において、 $a'-a''$  間を開放したときの状態方程式

$$\mathbf{x}(k+1) = \tilde{\mathbf{A}} \mathbf{x}(k) + \tilde{\mathbf{b}} \alpha''(k) \quad (3.26a)$$

$$\alpha'(k) = \tilde{\mathbf{c}} \mathbf{x}(k) \quad (3.26b)$$

を求める。次に、 $a''$  から  $a'$  への開ループ伝達関数  $L_0$

$$L_0(z) = -\tilde{\mathbf{c}}(z\mathbf{I} - \tilde{\mathbf{A}})^{-1}\tilde{\mathbf{b}} \quad (3.27)$$

をさきに示したファディープの方法で計算する。

#### $T_f$ の計算

図4.3-4 において、 $a'-a''$  間を閉じたときの状態方程式

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{b}u(k) \quad (3.28a)$$

$$y(k) = \mathbf{c}\mathbf{x}(k) \quad (3.28b)$$

を求める。次に、 $u$  から  $y$  への開ループ伝達関数  $T_f$

$$T_f(z) = \mathbf{c}(z\mathbf{I} - \mathbf{A})^{-1}\mathbf{b} \quad (3.29)$$

をさきに示したファディープの方法で計算する。

#### $t_{oi}$ の計算

$t_{oi}$  は、 $k=0$  としたときの  $u$  から  $y$  への伝達関数であるから、

$$t_{oi}(z) = \mathbf{c}(z\mathbf{I} - \tilde{\mathbf{A}})^{-1}\tilde{\mathbf{b}} \quad (3.30)$$

をファディープの方法で計算すればよい。

【注意】 上記において、多くの場合  $t_{oi} = 0$  となる。また、それがブロック図を見るだけで明らかになることがあり、その場合には、 $L_0$  の計算だけでよい。 $(T_f$  の計算は不要。)

さて、パラメータ感度  $S(z)$  は、上記の方法で求めた  $L_0$ ,  $T_f$ ,  $t_{oi}$  を式(3.25)に代入して、

$$S(z) = \frac{1}{1+L_0(z)} \left[ 1 - \frac{t_{oi}(z)}{T_f(z)} \right] \quad (3.31)$$

また、角周波数  $\omega$  における感度  $S$  を求めるには、サンプリング周期を  $T$  として式(3.31)の  $z$  に

$$z = e^{j\omega T} \quad (3.32)$$

を代入すればよい。但し、式(3.32)において

$$e^{j\left(\frac{2\pi}{T} + \omega\right)T} = e^{j\omega T} \quad (3.33)$$

であるから、感度  $S$  はサンプリング角周波数  $2\pi/T$  毎に同じ値となる。従って、感度  $S(\omega)$  の計算は、 $\omega = 0 \sim 2\pi/T$  の範囲だけ行えば十分である。

#### § 4-4 量子化誤差の影響の評価方法

本節では、デジタル制御系の量子化器により発生する出力の制御誤差を具体的に計算する方法を紹介する。§ 4-4-1 では、量子化誤差の影響を評価するために必要となる状態方程式を導出する。この状態方程式を用いて、§ 4-4-2 では、量子化誤差が常にその最大値をとった場合の出力の制御誤差を計算する方法を示す。しかし、量子化誤差が常にその最大値を取るとは限らない。そこで、§ 4-4-3 では、量子化誤差が白色ノイズであると仮定した場合の出力誤差の標準偏差を計算する方法を示す。

##### § 4-4-1 量子化器を含むデジタル系のモデリング

ここでは、量子化器が存在するために発生する状態変数の誤差が満たすべき状態方程式を導出する。この状態方程式は、この後で量子化誤差の影響を評価するために必要となる。

まづ、量子化器には、図4.4-1 に示すような打ち切りおよび丸めの2通りがある。同図において、入力  $x$  が量子化幅  $\Delta$  で量子化された時の出力を  $x_q$  としている。また、図中の破線は  $\Delta = 0$  の場合の入出力関係を示す。ここで、量子化により生じる誤差を  $\varepsilon$  と表し、次式で定義する。

$$\varepsilon = x_q - x \quad (4.1)$$

図4.4-1 から明らかなように  $\varepsilon$  の大きさは、次のように制限される。

$$\text{打ち切り} \quad 0 \leq \varepsilon \leq \Delta \quad (4.2)$$

$$\text{丸め} \quad -\Delta/2 \leq \varepsilon \leq +\Delta/2 \quad (4.3)$$

次に、量子化器を含む系において、量子化誤差が従うべき状態方程式を導出する。このため、まづ、量子化操作のない（すなわち、量子化幅が無限小の）理想系の状態方程式が次のように与えられるものとする。

$$x(k+1) = Ax(k) + br(k) \quad (4.4a)$$

$$y(k) = cx(k) \quad (4.4b)$$

これに対して、量子化器を含む系の状態方程式は次のようになる。

$$x_q(k+1) = Ax_q(k) + br(k) + \tilde{d}u(k) \quad (4.5a)$$

$$y_q(k) = cx_q(k) \quad (4.5b)$$

ただし、 $x_q(k)$  は量子化のある場合の状態変数であり、 $\tilde{d}u(k)$  は式(4.5)を式(4.4)に形式的に合わせて整理するために導入された量子化誤差による外乱項である。なお、この外乱項の大きさは、状態変数線図における量子化器の位置および接続の様子から決定でき、その上限を抑えることが出来る。

さて、式(4.5)から式(4.4)を引いて式(4.1)を用いて整理すれば、

$$\varepsilon(k+1) = A\varepsilon(k) + \tilde{d}u(k) \quad (4.6)$$

$$\Delta y(k) = y_q(k) - y(k) = c\varepsilon(k) \quad (4.7)$$

以上により、量子化誤差  $\varepsilon(k)$  が従うべき状態方程式は式(4.6)で与えられ、量子化による出力誤差は式(4.7)で与えられることが解った。

##### § 4-4-2 最悪ケースの誤差振幅の評価方法

ここでは、量子化による誤差振幅を BI-BO（有界入力-有界出力）安定に基づいて評価する方法を示す。なお、この方法は、Bartran [48] によって最初にあたえられた。

これより以下本項においてノルム  $\|\dots\|$  は、すべて最大値ノルムを表すことにする。このため、本項に示す評価では、量子化誤差が常にその最大値をとった場合の出力誤差を求めることになる。このことは、最悪ケースを想定して誤差振幅を評価することを意味する。

量子化誤差によって生じる状態の誤差  $\varepsilon(k)$  と出力の誤差  $\Delta y(k)$  は、さきに示した式(4.6)、(4.7)を満たす。この式で表されるシステムに対して、次の2つの仮定をする。

① 式(4.6) で与えられるシステムは、BI-BO 安定である。すなわち、

$$\sum_{l=0}^{\infty} \|cA^l \tilde{b}\| \leq M \quad (4.8)$$

② 量子化に伴う外乱項の  $u(k)$  は有界である。すなわち、

$$\|u(k)\| \leq u_M \quad (4.9)$$

以上の仮定のもとに、式(4.6) の解は、初期値を  $\varepsilon(0) = \varepsilon_0$  とすれば、

$$\varepsilon(k) = A^k \varepsilon_0 + \sum_{j=0}^{k-1} A^{k-1-j} \tilde{b} u(j) \quad (4.10)$$

これと式(4.7) から、出力誤差  $\Delta y(k)$  は、

$$\Delta y(k) = cA^k \varepsilon_0 + \sum_{j=0}^{k-1} cA^{k-1-j} \tilde{b} u(j) \quad (4.11)$$

上式の両辺のノルムをとれば、

$$\|\Delta y(k)\| = \|cA^k \varepsilon_0\| + \left\| \sum_{j=0}^{k-1} cA^{k-1-j} \tilde{b} u(j) \right\| \quad (4.12)$$

上式から、最終的な出力誤差  $\Delta y(\infty)$  の上限を次のようにして求めることが出来る。上式の右辺第1項は、行列  $A$  が漸近安定、すなわち、

$$\lim_{k \rightarrow \infty} \|A^k\| = 0 \quad (4.13)$$

であれば、最終的な出力誤差の評価をするときは無視してよい。このとき、式(4.12) から次式が成り立つ。

$$\begin{aligned} \|\Delta y(k)\| &= \left\| \sum_{j=0}^{k-1} cA^{k-1-j} \tilde{b} u(j) \right\| \\ &\leq \sum_{j=0}^{k-1} \|cA^{k-1-j} \tilde{b}\| \cdot \|u(j)\| \end{aligned} \quad (4.14)$$

上式と仮定②、すなわち、式(4.9) から、

$$\|\Delta y(k)\| \leq \left[ \sum_{j=0}^{k-1} \|cA^{k-1-j} \tilde{b}\| \right] u_M \quad (4.15)$$

さらに、上式の右辺の  $[\dots]$  の中身は式(4.8) で抑えられるから、

$$\|\Delta y(k)\| \leq M \cdot u_M \quad (4.16)$$

以上により、出力誤差振幅の最大値は  $M \cdot u_M$  で抑えられることが分かった。なお、 $M$  の値は、式(4.8) の左辺の和を  $\|\cdot\|$  が充分小さくなる項までで打ち切って近似的に計算できる。

§ 4-4-3 白色ノイズモデルに基づく誤差の評価方法

ここでは、量子化誤差が白色ノイズであると仮定して、出力誤差の大きさを時間領域で評価する方法を示す。なお、周波数領域での計算法もあるが、時間領域での計算法の方が簡便である。それは、時間領域での計算法では、システムの状態方程式から伝達関数を求めなくてよいからである。

量子化誤差が従うべき状態方程式と出力方程式は、すでに式(4.6), (4.7) に示した。ここで、量子化による外乱項の  $u(k)$  は図4.4-2 に示すような確率密度分布をもつ白色ノイズであると仮定する。図中  $\Delta_i$  は、 $i$  番目の量子化器の量子化幅である。このとき  $u(k)$  の共分散は次のようになる。

$$\text{cov}(u(k), u(l)) = \begin{cases} Q & (k=l) \\ 0 & (k \neq l) \end{cases} \quad (4.17)$$

$$Q = \text{diag} \left[ \sigma_1^2, \sigma_2^2, \dots \right] \quad (4.18)$$

$$\sigma_i^2 = \frac{\Delta_i^2}{12} \quad (4.19)$$

一方、出力誤差  $\Delta y(k)$  の共分散は、次のように与えられる（文献[27] の pp.169-172 参照）。

$$E[\Delta y(k), \Delta y(k+l)] = cV(A^t)^l c^t \quad (4.20)$$

ただし、 $V$  は次のリアプノフ方程式の解である。

$$AVA^t - V = -\tilde{b}Q\tilde{b}^t \quad (4.21)$$

このリアプノフ方程式の解は、次式で与えられる。

$$V = \sum_{i=0}^{\infty} A^i \tilde{b} Q \tilde{b}^t (A^t)^i \quad (4.22)$$

上式で  $V$  の値を実際に計算するときには、右辺の級数和を適当な項で打ち切って近似すればよい。

以上のようにして、 $V$  が計算できれば、出力誤差  $\Delta y(k)$  の標準偏差  $\sigma_y$  は、式(4.20) で  $l=0$  とすることにより、

$$\sigma_y = \sqrt{cVc^t} \quad (4.23)$$

のように計算することが出来る。

§ 4-5 デッドビート制御系のパラメータ感度と速度誤差の評価

本節では、第3章に示した予測型オブザーバを利用したデッドビート制御系のサンプリング周期を小さくしたとき、パラメータ感度および速度誤差がどのように変化するかを解析する。

§ 4-5-1 予測型オブザーバを用いたデッドビート制御系

ここでは、§ 3-6 で示した式のうちの後の解析が必要となる式を整理しておく。

デッドビート制御系のブロック図は、図4.5-1 のようになる。以下では、状態変数  $x_1 \sim x_4$  は、同図に示したようにとるものとする。§ 3-6 の結果により、デッドビート制御に必要な制御ゲインは、次のようになる。

$$K_i = \frac{1}{K_m T_s (1-P)} \quad (5.1)$$

$$K_p = \frac{-T_s P^2 + T_m (1-P)}{K_m T_s (1-P)^2} \quad (5.2)$$

$$F = P/R \quad (5.3)$$

ただし、

$$P = \exp(-T_s/T_m) \quad (5.4)$$

$$Q = K_m (1-P) \quad (5.5)$$

$$R = T_m (1-P) \quad (5.6)$$

$$S = K_m (T_s - R) \quad (5.7)$$

サンプリング周期  $T_s$  とデッドビート制御に必要な制御ゲインの関係は図4.5-2のようになる。

#### § 4-5-2 パラメータ感度の評価

ここでは、サンプリング周期を変えた場合のデッドビート制御系のパラメータ感度を計算する。

まず、パラメータ感度の計算に必要となる状態方程式を導出しておく。電動機のゲイン定数  $K_m$  が変動した場合のパラメータ感度を計算する場合を考える。式(5.4)-(5.7)で、 $Q$  と  $S$  が  $K_m$  に比例して変化するから、図4.5-1において  $a$  点でループを切断した場合の状態方程式を求めればよい。その結果は、次のようになる。

$$x(k+1) = A_a x(k) + b_a a''(k) \quad (5.8)$$

$$a'(k) = c_a x(k) \quad (5.9)$$

ただし、

$$A_a = \begin{bmatrix} P & 0 & 0 & 0 \\ R & -\bar{R}F & -\bar{R} + \bar{S}K_p & -\bar{S}K_i \\ 0 & \hat{P}F & \hat{P} - \hat{Q}K_p & \hat{Q}K_i \\ 0 & -1 - \bar{R}F & -\bar{R} + \bar{S}K_p & 1 - \bar{S}K_i \end{bmatrix} \quad (5.10)$$

$$b_a = \begin{bmatrix} Q \\ S \\ 0 \\ 0 \end{bmatrix} \quad (5.11)$$

$$c_a = [0, 0, -K_p, +K_i] \quad (5.12)$$

ここで、 $a$  点を切断した場合には、 $\omega_r$  から  $\omega_m$  への直達経路がないからパラメータ感度は、式(3.12)から

$$S(z) = \frac{1}{1 + L_0(z)} \quad (5.13)$$

ただし、ループ伝達関数  $L_0(z)$  は、

$$L_0(z) = -\frac{a'(z)}{a''(z)} = -c_a [zI - A_a]^{-1} b_a \quad (5.14)$$

以上の式を使って計算したパラメータ感度  $S(z)$  の周波数特性を図4.5-3 にしめす。同図からデッドビート応答で最も重要な周波数帯域（サンプリング周波数の  $1/4$  あたり）の感度が1よりかなり大きくなっている。しかも、この傾向はサンプリング周波数が小さいほど顕著である。このことからデッドビート制御系で電動機のゲイン定数  $K_m$  が変動したとき、サンプリング周期が短いほど応答波形の乱れが大きくなることが分かる。

### § 4-5-3 量子化誤差による速度制御誤差の評価

ここでは、位置検出にロータリエンコーダを用いた場合の量子化誤差が電動機速度  $\omega_m$  に与える影響を § 4-4 に示した量子化誤差の評価法により評価する。

まず、§ 4-4 に示した量子化誤差の評価法を適用するために必要な状態方程式を導く。このためには、図4.5-1 において  $v$  を入力として  $\omega_m$  を出力とする系を考えればよいから、必要な状態方程式は、

$$\mathbf{x} = A_n \mathbf{x}(k) + \mathbf{b}_n v(k) \quad (5.15)$$

$$y(k) = \mathbf{c}_n \mathbf{x}(k) \quad (5.16)$$

ただし、

$$A_n = \begin{pmatrix} P & 0 & QK_p & QK_i \\ R & -\hat{R}F & -\hat{R} - (S - \hat{S})K_p & (S - \hat{S})K_i \\ 0 & \hat{P}F & \hat{P} - \hat{Q}K_p & \hat{Q}K_i \\ 0 & -1 - \hat{R}F & -\hat{R} + \hat{S}K_p & 1 - \hat{S}K_i \end{pmatrix} \quad (5.17)$$

$$\mathbf{b}_n = (0, 1, 0, 0)^t \quad (5.18)$$

$$\mathbf{c}_n = (1, 0, 0, 0) \quad (5.19)$$

上式を使えば、§ 4-4 に示した方法により、速度誤差の上限と標準偏差を計算できる。

図4.5-4 にサンプリング周期を変えた場合の速度誤差の計算結果を示す。なお、図中には参考のためシミュレーションにより求めた速度誤差の最大値と標準偏差も併せて示した。同図から、どの曲線も、サンプリング周期にほぼ逆比例して速度誤差が増加することを示している。

以上、本節では、デッドビート応答をする系のサンプリング周期を変えた場合のパラメータ感度と速度誤差を解析した。デッドビート応答をする系では、サンプリング周期を短くすれば、それに比例して応答が速くなる。しかし、その反面、サンプリング周期を短くすればするほど、速度誤差が増加し、パラメータ感度も増加することが分かった。従って、デッドビート制御系では、必要以上にサンプリング周期を短くしない方がよいといえる。

### § 4-6 応答時間を合わせた制御系のパラメータ感度と速度誤差の評価

本節では、応答時間を一定に保ったままサンプリング周期を短くすれば、パラメータ感度を小さくでき、速度誤差の増加も抑えることが出来ること示す。

#### § 4-6-1 応答時間を合わせるための制御ゲインの計算方法

ここでは、サンプリング周期を短くしても電動機速度の応答時間が変わらないようにするために必要な制御ゲインの計算法を示し、実際のデータによる計算結果を示す。

図4.6-1 に本節で制御ゲインの計算法を示すために使用する電動機速度制御系のブロック図を示す。予測型オブザーバを用いた電動機速度制御系は、等価的に図4.6-1 の様になる。ただし、同図では予測型オブザーバの代わりに電動機速度  $\omega_m$  が直接検出可能であり、かつ、制御演算の無駄時間もないとしている。予測型オブザーバを用いた場合でも、パラメータ誤差がなければ、 $\omega_m$  の変化に対する  $\omega_m$  の応答は、図4.6-1 と同じである。また、予測型オブザーバを隣に書か

ない図4.6-1の方がシステムが簡略化され、以下に示す制御ゲインの計算が容易になる。

サンプリング周期が異なる場合でも応答波形をほぼ同じにするために、次の方法を用いる。

サンプリング周期の異なる2つの系において、各々の系の  $z$  平面上での極を  $z = \exp(T_S \cdot s)$  でラプラス平面に写像し、ラプラス平面での主要極（実軸に近いもの）を一致させる。

このようにすれば、サンプリング周期の異なる2つの制御系の応答波形がほぼ同じになる。その理由を以下に説明する。

$z$  平面上の複素極をラプラス平面に写像すると、図4.6-2に示すように、サンプリング角周波数  $2\pi/T_S$  を周期とする無限個の極が生じる。しかし、サンプリング周期の異なる制御系の無限個の極の位置を、制御ゲインの調整だけで一致させることは不可能である。そこで、ラプラス平面上の実軸に近い極、すなわち、主要極（低周波数領域における系の応答を代表している極）から順次合わせるように制御ゲインを決める。実軸から離れた残りの極は合わないが、応答波形はあまり違わない。それは、制御系の周波数特性は、一般に高周波数領域では小さくなるように設計されることが多いからである。従って、ラプラス平面上の主要極を合わせておけば、サンプリング周期の異なる2つの制御系の周波数特性の低周波領域がほぼ等しくなり、時間応答波形の主要部分が合うように出来る。

この方法を利用した制御ゲイン計算の手順は次のようになる。

[サンプリング周期の異なる2つの制御系の制御ゲインの計算手順]

**step1** 基準となる応答波形をもつ離散時間制御系のサンプリング周期  $T_S^a$  と制御ゲインを与える。

**step2** 基準となる制御系の  $z$  平面極をラプラス平面に写像する。（後で示す例では、式(6.2)~(6.4)がこれに相当する。）

**step3** 新しいサンプリング周期  $T_S^b$  を持つ制御系のラプラス平面上での主要極が(step2)で求めたものと一致するように制御ゲインを計算する。（後で示す例では、式(6.8)がこれに相当する。）

以下ではこの手順により、図4.6-1に示した構造をもつ系の制御ゲインを計算する。

**step1** 基準系のサンプリング周期と制御ゲインを  $T_S, K_i, K_p$  とする。このとき、図4.6-1の系のパルス伝達関数を求めると次のようになる。

$$\frac{\omega_m(z)}{\omega_r(z)} = \frac{K_i T_S Q z}{z^2 + bz + cz} \quad (6.1)$$

ただし、

$$b = [K_i S + K_p Q - P - 1] \quad (6.2a)$$

$$c = [K_i(Q \cdot R - P \cdot S) - K_p \cdot Q + P] \quad (6.2b)$$

**step2** 基準となる制御系の  $z$  平面極をラプラス平面に写像する。式(6.1)の極を  $z_r \pm jz_i$  とすれば、

$$z_r = -b/2 \quad (6.3a)$$

$$z_i = \frac{1}{2} \sqrt{4c - b^2} \quad (6.3b)$$

これを  $z = \exp(T_S \cdot s)$  でラプラス平面に移した極を  $s_r \pm js_i$  とすれば、

$$s_r = \frac{1}{T_S} \ln [z_r^2 + z_i^2] \quad (6.4a)$$

$$s_i = \frac{1}{T_S} \tan^{-1} \left[ \frac{z_i}{z_r} \right] \quad (6.4b)$$

step3 上とは異なるサンプリング周期  $T_S^b$  もつ系を考えて、上記とは逆に、ラプラス平面の主要極を  $z$  平面に移すことを考える。ただし、サンプリング周期が違うから写像  $z = \exp(T_S^b \cdot s)$  を使って、ラプラス平面極  $s_r \pm js_i$  を  $z$  平面に移す。その結果を、 $z_r^b \pm z_i^b$  とすれば、次のようになる。

$$z_r^b = \exp(T_S^b \cdot s_r) \cos(T_S^b \cdot s_i) \quad (6.5a)$$

$$z_i^b = \exp(T_S^b \cdot s_r) \sin(T_S^b \cdot s_i) \quad (6.5b)$$

一方、サンプリング周期が  $T_S^b$  で、構造が図4.6-1 の系の極は、式(6.3)と同様になり、それらを各パラメータに ' を付けて区別するようにすれば、

$$z_r' = -b'/2 \quad (6.6a)$$

$$z_i' = \frac{1}{2} \sqrt{4c'^2 - b'^2} \quad (6.6b)$$

ただし、

$$b' = [K_i' S' + K_p' Q' - P' - 1] \quad (6.7a)$$

$$c' = [K_i' (Q' \cdot R' - P' \cdot S') - K_p' \cdot Q' + P'] \quad (6.7b)$$

式(6.5) と式(6.6) を等しくするような制御ゲイン  $K_i'$ ,  $K_p'$  をもとめると、

$$K_i' = \frac{\exp(2T_S^b \cdot s_r) - 2\exp(T_S^b \cdot s_r) \cos(T_S^b \cdot s_i) + 1}{Q' \cdot T_S^b} \quad (6.8a)$$

$$K_p' = \frac{P' + 1 - 2\exp(T_S^b \cdot s_r) \cos(T_S^b \cdot s_i) - S' K_i'}{Q'} \quad (6.8b)$$

以上により、異なるサンプリング周期でも時間応答波形をほぼ等しくする制御ゲインを求めるひとつの方法を導出した。この方法は多くの場合にうまくゆく。ただし、基準応答が有限時間整定応答であるときには、次のような変更が必要である。

基準応答が有限時間整定応答である場合には、その系の極は  $z$  平面の原点にある ( $z_r = z_i = 0$ )。よって、式(6.4a)が使えなくなる。これを回避するため  $z$  平面上の極を原点からわずかに離れるように制御ゲインを次のように変更をする。

$$K_i^d = \alpha K_i^d \quad (6.9a)$$

$$K_p^d = \alpha K_p^d \quad (6.9b)$$

ただし、 $K_i^d$ ,  $K_p^d$  は、有限時間整定するための制御ゲインであり、図4.6-1 の系では、次のようになる。

$$K_i^d = \frac{1}{T_S \cdot Q} \quad (6.10a)$$

$$K_p^d = \frac{-T_S \cdot P^2 + R}{T_S \cdot Q \cdot (1 - P)} \quad (6.10b)$$

上のようにしても、 $\alpha$  が 1 に非常に近ければ、制御系の応答はほとんど有限時間整定応答に近くなる。しかも、基準となる制御系の極が原点から離れるから、前に示した制御ゲインの計算法を使うことが出来る。

以上の計算法を使って実際に計算した結果を示す。なお、基準応答は、サンプリング周期 25ms のデッドビート応答である。図4.6-3 には、 $\alpha$  が 1 に近い場合の  $\omega_m$  のステップ応答の計算結果を示す。同図から、時間応答波形の変化が小さいことが分かる。図4.6-4 は、 $\alpha$  を変化させた場合の  $z$  平面極と、対応するラプラス平面極の動きを示す。図4.6-5 には、 $\alpha = 0.99$  のときの、制御ゲインの計算結果を示す。

#### § 4-6-2 パラメータ感度の評価

ここでは、応答時間をほぼ一定に保ったままサンプリング周期を短くした場合のパラメータ感度を評価し、サンプリング周期が短いほどパラメータ感度を小さく抑えることが出来ることを示す。なお、制御ゲインは前項に示したものをを用いる。パラメータ感度の計算式は § 4-5-2 に示したものと同一である。

図4.6-6 には、電動機のゲイン定数  $K_m$  の変化を想定したパラメータ感度の計算結果を示す。なお、制御ゲインには図4.6-5 に示したものをを使っているから、サンプリング周期を変えた場合でも、電動機速度の立ち上がり時間はおよそ 25ms である。また、このときのラプラス平面上の主要極は、

$$-97.37 \pm 58.79j \quad (\alpha = 0.99) \quad (6.11)$$

であった。このことから、この制御系のカットオフ周波数は、58.79 rad/s である。

フィードバック制御系としては、カットオフ周波数以下のパラメータ感度を 1 以下に抑えることが望まれる。そこで、カットオフ周波数における感度の値を、サンプリング周期を変えて計算した結果が、図4.6-7 である。同図から、パラメータ感度が 1 以下に抑えられるのは、サンプリング周期が約 6ms 以下のときである。しかし、2ms よりさらに小さくしても、パラメータ感度抑制効果は小さいことが分かる。従って、この例では、サンプリング周期は 2ms 程度にすることが、パラメータ感度抑制の観点からは望ましいといえる。

以上に示した数値例から、応答時間を一定にしてサンプリング周期を短くすれば、パラメータ感度を小さく抑えられることが分かる。

#### § 4-6-3 量子化誤差による速度制御誤差の評価

ここでは、ロータリエンコードの量子化誤差による電動機速度の制御誤差の大きさを評価し、応答時間を一定に保つことにより、速度制御誤差の増大をある程度抑制できることを示す。量子化誤差の計算式は § 4-5-3 と同一である。

図4.6-8 は、サンプリング周期を変えた場合の電動機速度の制御誤差の数値計算結果を比較したものである。なお、速度制御誤差の上限および標準偏差の計算値のほかに、§ 3-8-2 に示したデジタルシミュレーションにより求めた値も示してある。同図から、サンプリング周期を短くしても応答時間 25ms の系の速度制御誤差は、デッドビート系に比べて大きくならないことが分かる。この理由は、図4.6-5 に示したようにデッドビート系に比べて制御ゲインが小さく量子化誤差の増幅率が小さいためであると考えられる。さらに、応答時間 25ms の系のシミュレーション結果によると、サンプリング周期を短くすると速度制御誤差はむしろ減少する傾向にあることが分かる。

なお、図4.6-8 から分かるように、速度制御誤差の解析値とシミュレーション結果は、必ずしもよく一致しているわけではない。例えば、応答時間 25ms の系の速度誤差の上限値に関しては、解析値がシミュレーション結果よりもかなり大きくなっている。§ 4-4-2 に示した BI-BO 安定に基づく計算方法では常に最大の量子化誤差が加えられると仮定しているが、そのようなケースはむしろ希であり、上限の速度誤差が必ずしも発生するわけではない。また、白色ノイズモデルに基づく標準偏差に関しては、デッドビート制御系の場合にシミュレーション結果との差が大きい。従って、より正確に制御誤差を評価する手法の検討が今後必要である。

#### § 4-6-4 制御応答の実験結果による比較

ここでは、電動機速度の過渡応答のシミュレーション結果および実験結果を示す。これにより、応答時間を保ってサンプリング周期を短くすれば電動機のゲイン定数が変動した場合でも過渡応答波形の乱れが少ないことを示す。なお、実験装置は、§ 3-7-1 に示したもので、ハードウェア構成と定数は、それぞれ、図3.7-1 と表3.7-1 に示してある。

図4.6-9 は、電動機のゲイン定数  $K_m$  が変動したときの設定速度および外乱のステップ変化に対する電動機速度  $\omega_m$  の応答のシミュレーション結果であ

る。制御ゲインは、§ 4-6-1 に示したとおりである。同図(a)  $T_s=0.025s$  に比べて、同図(b)  $T_s=0.01s$  の方が  $K_m$  の変動に対して速度の応答波形の乱れが少ないことが分かる。なお、参考として、瞬時速度の検出ができて演算遅れもない理想的な系の応答のシミュレーション結果を図4.6-10 に示しておく。

図4.6-11 は、サンプリング周期が 25ms および 10ms の制御系の応答の実験結果である。制御ゲインは § 4-6-1 に示したように決めた。サンプリング周期が異なっても、応答時間がほぼ同じであることが分かる。

図4.6-12 には、電動機駆動用チョップの直流電源電圧  $E_0$  を変化したときの電動機速度の応答を比較したものである。なお、このチョップでは電流制御をしておらず、電圧増幅器として機能させているから、 $E_0$  を変化することは電動機のゲイン定数  $K_m$  が変化することに相当する。図4.6-10 の中段 ( $E_0=55V$ ) が設計状態であり、上段と下段は  $E_0$  を  $\pm 18\%$  変化させた場合である。サンプリング周期が 10ms の方が 25ms に比べて電動機速度の応答波形の乱れが少ないことが分かる。また、この結果は、図4.6-9 のシミュレーション結果ともよくあっている。なお、使用したマイクロコンピュータ (Z-80, 2.5MHz, AM9511付き) の演算速度の制約により、予測型オブザーバの演算を行うときはサンプリング周期を 10ms 以下に出来なかった。

以上により、応答時間を保ってサンプリング周期を短くすれば、電動機のゲイン定数が変動しても応答波形の乱れが少ないことが分かる。

#### § 4-7 本章のまとめ

本章では、電動機のデジタル速度制御系のサンプリング周期を短くすることにより、パラメータ感度を低減できるかどうか、さらに、その場合の速度精度および応答がどのようになるかについて解析・評価を行った。その結果をまとめると以下ようになる。

- (1) 状態方程式からパラメータ感度および量子化誤差による制御誤差を数値計算する方法を、それぞれ、§ 4-3 および § 4-4 に整理して示した。従来の伝達関数に基づく計算法に比べれば、状態方程式に基づくため制御系のループが複雑であっても計算機プログラム化が容易であるという特長が

ある。しかし、量子化誤差による制御誤差の数値計算法については、§ 4-6-3 に述べたようにシミュレーション結果との差が依然大きいいため、今後も検討が必要である。

- (2) デッドビート制御系のパラメータ感度および速度誤差の評価を行った。その結果、サンプリング周期を短くすると応答が速くなる反面、パラメータ感度および速度誤差が増加することが分かった (§ 4-5-2~3)。よって、デッドビート制御系では、必要以上にサンプリング周期を短くしない方がよいと考える。
- (3) 応答時間を保ったままサンプリング周期を短くした系のパラメータ感度および速度制御誤差を評価した (§ 4-6-2~3)。その結果、応答時間を変えないようにしてサンプリング周期を短くすれば、パラメータ感度を低減でき、かつ、速度制御誤差の増加も抑えられることが分かった。また、実験結果により、応答時間が同じならばサンプリング周期が短い系の方が、定数が変動しても過渡応答波形の乱れが少ないことが分かった (§ 4-6-4)。

<<付記>> 本章の解析では、電動機の電気的時定数を無視している。これは、本研究を行った当時の 8bit マイクロプロセッサによるサンプリング周期が 10ms 以上であったため、これに比べて電気的時定数は小さいと無視したものである。

#### § 4-8 本章の参考文献

- [1] 木村：「解説：多変数システム理論—設計論の話題」、システムと制御、Vol.26, No.3, pp.145-153, 1982.3
- [2] I.M. Horowitz: "Synthesis of Feedback Systems", Academic Press, 1963
- [3] H.W. Bode: "Network Analysis and Feedback Amplifier Design", Van Northland, 1945 (喜安訳：「回路網と帰還の理論」、岩波書店、1955)
- [4] 北森：「PID、I-PD制御からの発展の道」、システムと制御、Vol.27, No.5, pp.287-294, 1983.5

- [5] 木村、当房：「部分的モデルマッチングによるデジタル制御系の設計」、計測自動制御学会論文集、Vol.18, No.8, pp.777-783, 1982.7
- [6] 計測自動制御学会編：「自動制御ハンドブック 基礎編」、オーム社、1983.10 (p.465)
- [7] 杉浦、鈴木：「感度解析とそれを考慮した自動制御系の最適設計」、計測と制御、Vol.8, No.7, pp.443-57, 1969.7
- [8] J.B. Cruz, W.R. Perkins: "A New Approach to the Sensitivity Problem in Multivariable Feedback System Design", IEEE Trans., Vol.AC-9, pp.216-223, 1964.7
- [9] W.R. Perkins, J.B. Cruz: "The Parameter Variation Problem in State Feedback Control Systems", Trans. ASME, J. Basic Eng., Ser.D, Vol.87, pp.120-124, 1965.3
- [10] P.V. Kokotovic, R.S. Rutman: "Sensitivity of Automatic Control System --Survey--", Automation and Remote Control, Vol.26, No.4, pp.727-749, 1965
- [11] 小野、井上：「制御系における感度解析」、制御工学、Vol.13, No.3, pp.167-178, 1969.3
- [12] 杉浦：「感度の考慮による最適自動制御系設計への一つの道程」、制御工学、Vol.13, No.3, pp.179-190, 1963.3
- [13] 高松、橋本：「プロセスシステムの最適化と感度解析」、計測と制御、Vol.9, No.6, pp.426-438, 1970.6
- [14] 井上：「感度を考慮した最適制御系のシンセシス」、計測と制御、Vol.9, No.8, pp.565-579, 1970.8
- [15] 杉浦：「感度解析とそれを考慮した最適自動制御系の研究」、計測と制御、Vol.10, No.1, pp.1-16, 1971.1
- [16] 井上、橋本：「システム感度の研究に関する最近の動向」、システムと制御、Vol.15, No.8, pp.645-658, 1971.8
- [17] P.E. Fleischer: "Optimum Design of Passive-Adaptive, Linear Feedback Systems with Varying Plants", IRE Trans. on Auto. Contr., Vol.AC-7, No.2, pp.117-128, 1962.3
- [18] 奥山：「線形制御系パラメータ感度の評価と受動的適応制御系の設計」、計測と制御、Vol.3, No.10, pp.759-767, 1964.10
- [19] 西村、藤井、井上：「受動的適応制御の設計」、制御工学、Vol.10, No.6, pp.304-311, 1966.6
- [20] 西村、藤井、井上：「受動的適応制御の設計II」、制御工学、Vol.11, No.8, pp.411-420, 1967.8
- [21] 西村、藤井、井上、尾関：「受動的適応制御の設計III」、制御工学、Vol.12, No.3, pp.128-136, 1968.3
- [22] 井上、杉本：「モデル規範型制御系のパラメータ感度とその設計法」、制御工学、Vol.13, No.9, pp.580-587, 1969.9
- [23] 玉置、大西、宮地：「受動的適応制御によるDCサーボモータの低感度ロ

- バスト制御」、昭和60年電気学会全国大会、No.600, 1985.4
- [24] 黒沢、細田：「シュミレータ追従付速度制御」、昭和60年電気学会全国大会、No.604, 1985.4
- [25] I.M. Horowitz: "The Sensitivity in Sampled-Data Feedback System", IRE Trans. on Auto. Contr., Vol.AC-6, No.2, pp.251-259, 1961.9
- [26] D.P. Lindorff: "Sensitivity in Sampled-Data System", IEEE Trans., Vol.AC-8, No.2, pp.120-125, 1963.4
- [27] R. Iserman: "Digital Control Systems", Springer-Verlag, pp.105-109, 1981
- [28] 山下他：「デジタルプロセス制御」、コロナ社、pp.149-151, 1969.3
- [29] 木村：「デジタル信号処理と制御」、昭晃堂、p.281, 1982.9
- [30] P. Katz: "Digital Control Using Microprocessor", Prentice Hall International, p.236, 1981
- [31] 森下編：「デジタル計装制御」、計測自動制御学会、pp.82-83, 1983.1
- [32] 真鍋：「デジタル制御、実際面における問題点と展望」、計測と制御、Vol., No., pp.594-598, 1983.7
- [33] 伊藤：「デジタル制御理論の展開」、計測と制御、Vol.22, No.7, pp.587-593, 1983.7
- [34] R.E. Kalman, J.E. Bertarm: "General Synthesis Procedure for Computer Control of Single-loop and Multiloop Linear System (An Optimal Sampling System)", AIEE Trans. Pt.2 Application and Industry, Vol.77, pp.602, 1951.1
- [35] 美多：「演算時間を考慮したデジタル制御系の設計」、計測と制御、Vol.22, No.7, pp.614-619, 1983.7
- [36] B. Liu: "Effect of Quantization on the Accuracy of Digital Filters --A Review", IEEE Trans. on CT, Vol.CT-18, No.6, pp.670-677, 1971
- [37] S.P. Parker, S.F. Hess: "Limit Cycle Oscillation in Digital Filters", IEEE Trans. on CT, Vol.CT-18, No.6, pp.687-697, 1971
- [38] A.V. Oppenheim, R.W. Schaffer: "Digital Signal Processing", Prentice-Hall, 1975
- [39] T.A.C.M. Clasen, W.F.G. Meckengrauke, J.B.H. Peek: "Effects of Quantization and Overflow in Recursive Digital Filters", IEEE Trans., Vol.ASSP-24, No.6, pp.517-529, 1976
- [40] 青山：「語長制限されたデジタルフィルタの解析と最適設計法」、日経エレクトロニクス、No.8-8, pp.100-134; No.8-22, pp.125-136; No.9-5, pp.137-144, 1977
- [41] 太田、樋口：「状態方程式に基づくデジタルフィルタの構成理論」、計測と制御、Vol.18, No.2, pp.180-187, 1979.2
- [42] 石井：「デジタルフィルタ(小特集)」、電気学会雑誌、Vol.100, No.2, pp.113-117, 1980.2
- [43] G.F. Franklin, I.D. Powell (羽根田訳)：「ダイナミックシステムのデ

ジタル制御」、森北出版、1985.5

- [44] P. Katz: [30]
- [45] P. Moroney: "Issues in the Implementation of Digital Feedback Compensators", MIT Press, 1983
- [46] 高橋(北森訳):「制御と力学系」、コロナ社、pp.117-119
- [47] B.C. Kuo: "Digital Control System", Holt-Saunders International Editions, pp.245-255, 1980
- [48] J.E. Bertram: "The Effect of Quantization in Sampled-Feedback Systems", AIEE Trans. Pt.2 Application and Industry, Vol.76, pp.177-182, 1958.9
- [49] J.B. Slaughter: "Quantization Errors in Digital Control Systems", IEEE Trans., Vol.AC-9, pp.70-74, 1964.1
- [50] G.W. Jhonson: "Upper Bound on Dynamic Quantization Error in Digital Control Systems via the Direct Method of Liapunov", IEEE Trans., Vol.AC-10, No.4, pp.439-448, 1965.10
- [51] G.N.T.Lack: "Comment on Upper Bound on ..... (上の討論)", IEEE Trans., Vol.AC-11, No.2, pp.331-334, 1966.4
- [52] K.J. Astrom (中村他訳):「確率制御理論入門」、コロナ社、pp.130-143, 1975
- [53] R.B. Kieburztz: "An Experimental Study of Roundoff Effects in a Tenth-Order Recursive Digital Filter", IEEE Trans., Vol.COM-21, No.6, pp.757-763, 1973
- [54] R.B. Kieburztz, V.B. Lawrence, K.V. Mina: "Control of Limit Cycle in Recursive Digital Filters by Randomized Quantization", IEEE Trans., Vol.CAS-24, No.6, pp.291-299, 1977.6
- [55] M. Buttner: "Elimination of Limit Cycle in Digital Filters with Very Low Increase in the Quantization Noise", IEEE Trans., Vol.CAS-24, No.6, pp.300-304, 1977.6
- [56] M.G. Safonov, A.L. Laub, G.L. Hartmann: "Feedback Properties of Multivariable Systems -- The Role and Use of the Return Difference", IEEE Trans., Vol.AC-26, No.1, pp.47-65, 1981.2

#### 第4章の図・表のリスト

§ 4-1 ~ § 4-2

なし

§ § 4-3-1

図4.3-1 変動要素  $k$  を含むシステムの信号線図

図4.3-2 ナイキスト線図

§ § 4-3-2

図4.3-3 ノイズを考慮したフィードバック系

§ § 4-3-3

図4.3-4 変動要素  $k$  を含む高次システム

§ § 4-4-1

図4.4-1 量子化器の入出力と量子化誤差

§ § 4-4-2

なし

§ § 4-4-3

図4.4-2 白色ノイズの確率密度

§ § 4-5-1

図4.5-1 予測型オブザーバを用いたデッドビート制御系のブロック図

図4.5-2 デッドビート制御系のゲイン

§ § 4-5-2

図4.5-3 デッドビート制御系のパラメータ感度

§ § 4-5-3

図4.5-4 デッドビート制御系の速度誤差

§ § 4-6-1

図4.6-1 予測型オブザーバを用いた系の等価ブロック図

図4.6-2 サンプリング周期が異なる系の極配置手順

図4.6-3  $\alpha = 0.99$  のときの応答

図4.6-4  $\alpha:1 \rightarrow 0$  としたときの極の動き

図4.6-5 応答時間 25ms の系とデッドビート制御系の制御ゲインの比較

§ § 4-6-2

図4.6-6 応答時間 25ms の系のパラメータ感度

図4.6-7 応答時間 25ms の系のカットオフ周波数における感度

§ § 4 - 6 - 3

図4.6-8 応答時間 25ms の系とデッドビート制御系の速度誤差の比較

§ § 4 - 6 - 4

図4.6-9  $K_m$  変動時の予測型オブザーバを用いた系の過渡応答

図4.6-10  $K_m$  変動時の理想的 I - P 制御系の過渡応答

図4.6-11 設計状態での制御応答

図4.6-12 電源電圧を変化させたときの制御応答の比較

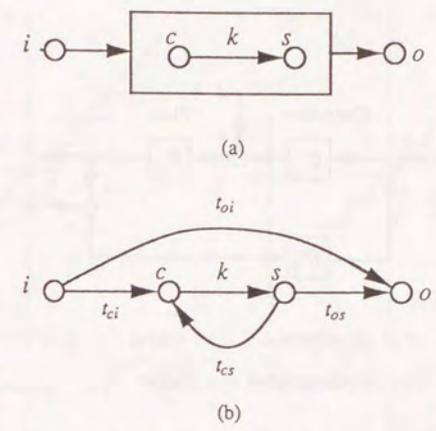


図4.3-1 変動要素  $k$  含むシステムの信号線図

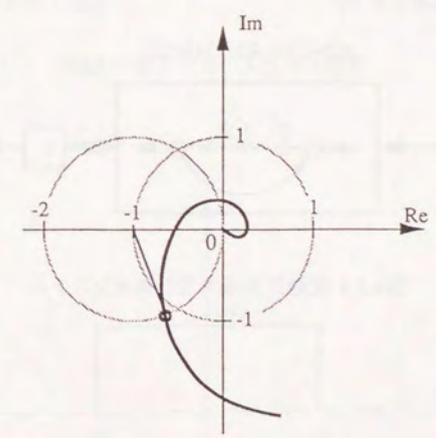
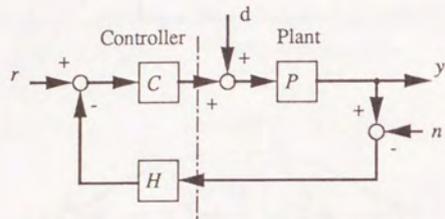


図4.3-2 ナイキスト線図



$r$ : reference     $y$ : output  
 $d$ : disturbance     $n$ : noise

図4.3-3 ノイズを考慮したフィードバック系

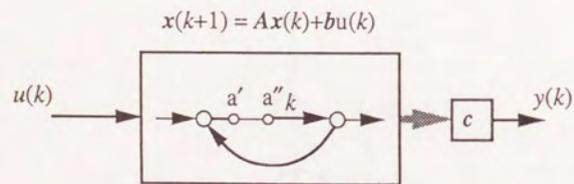


図4.3-4 変動要素  $k$  を含む高次システム

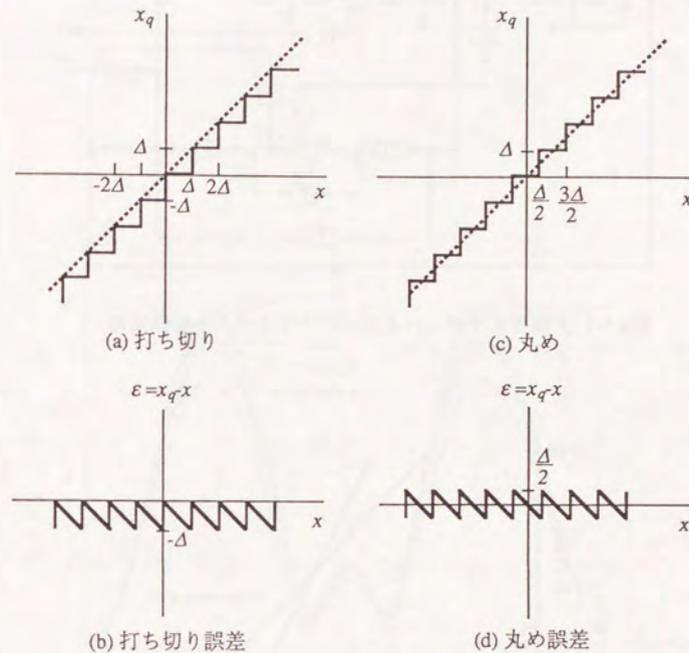


図4.4-1 量子化器の入出力と誤差

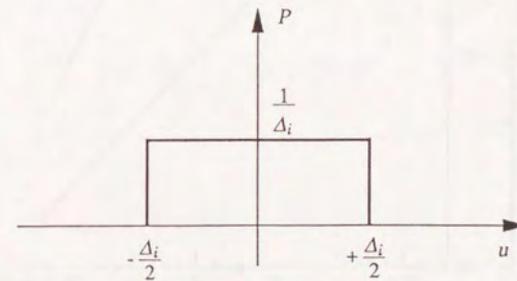


図4.4-2 白色ノイズの確率密度

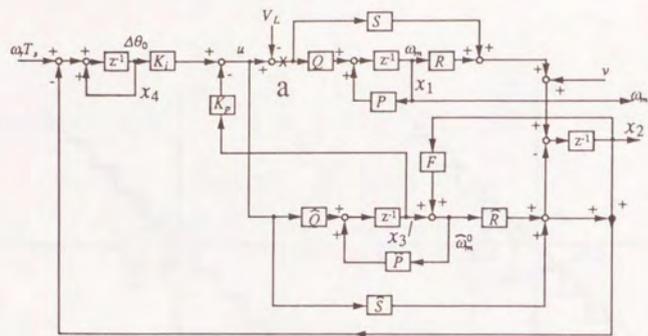


図4.5-1 予測型オブザーバを用いたデッドビート制御系のブロック図

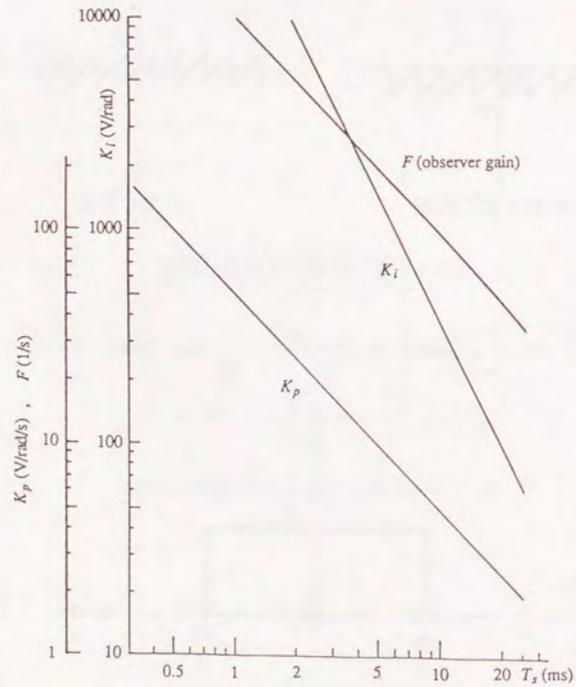


図4.5-2 デッドビート制御系のゲイン

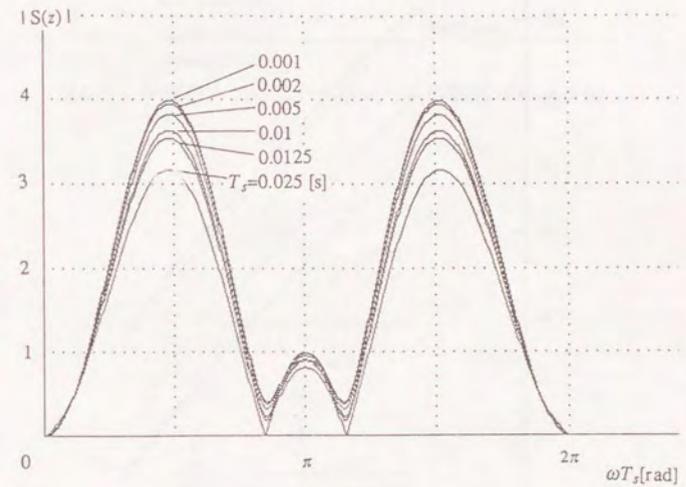


図4.5-3 デッドビート制御系のパラメータ感度  
( $K_m$ の変動)  $K_m=2.83$ [rad/s]  $T_m=0.095$  [s]

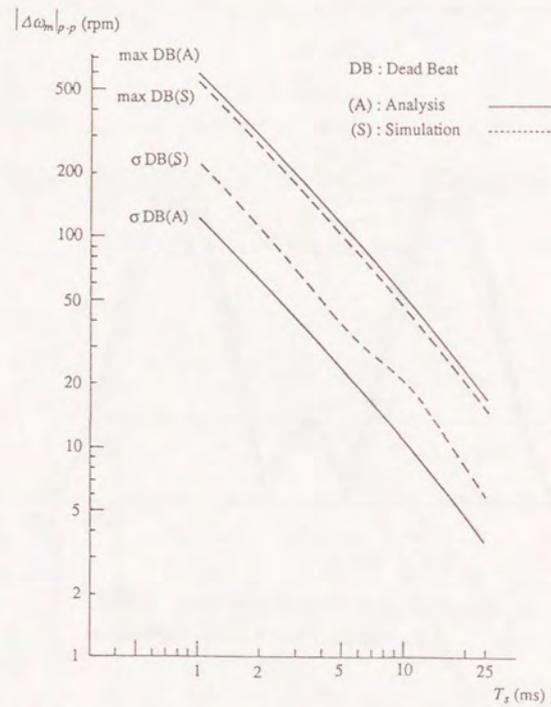


図4.5-4 デッドビート制御系の速度誤差

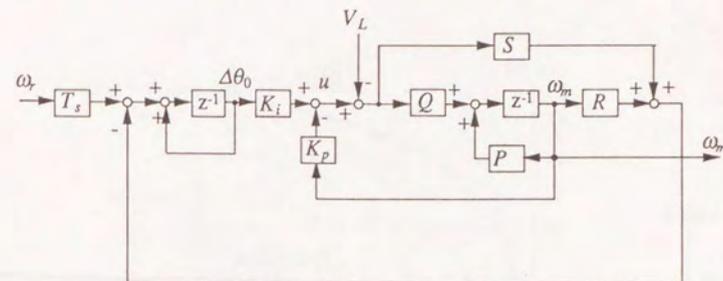


図4.6-1 予測型オブザーバを用いた系の等価ブロック図

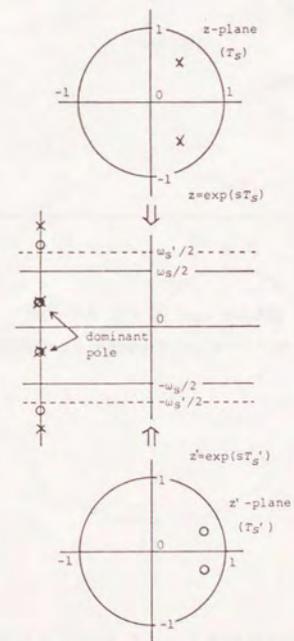


図4.6-2 サンプルング周期が異なる系の極配置手順

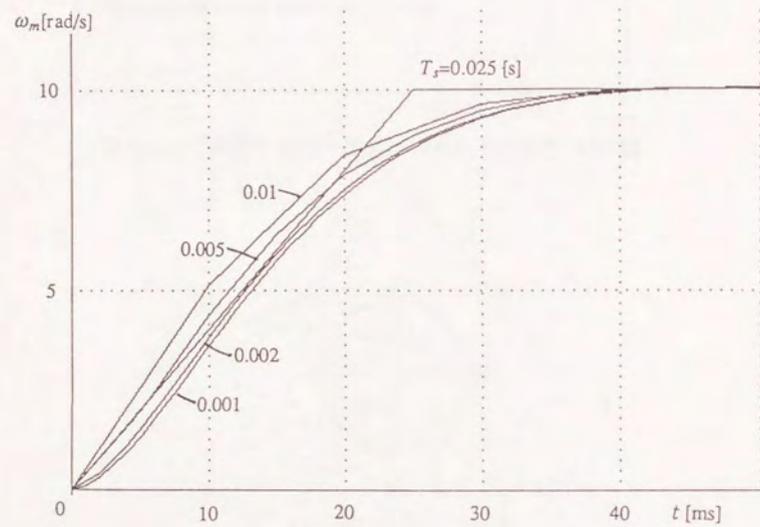


図4.6-3  $\alpha=0.99$  のときの応答  
 $K_m=2.83[\text{rad/s}]$   $T_m=0.095[\text{s}]$

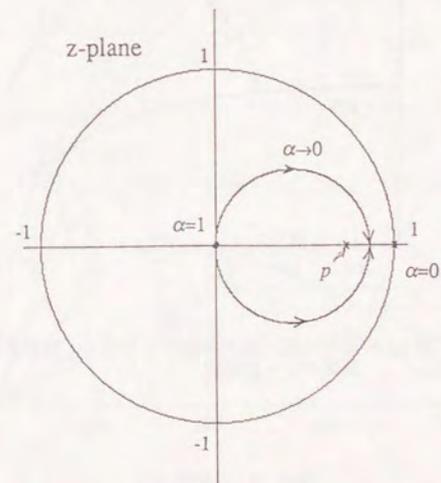
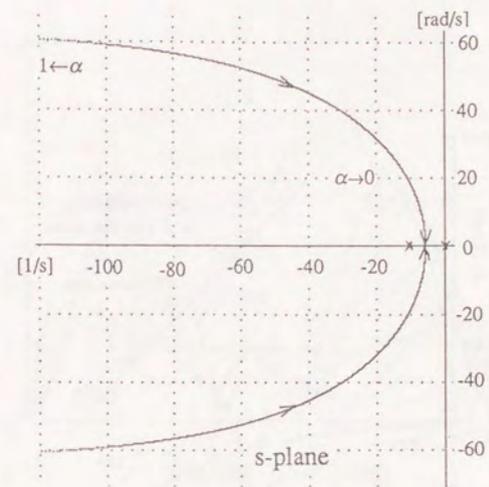


図4.6-4  $\alpha:1 \rightarrow 0$  としたときの極の動き  
 $T_m=0.095[\text{s}]$   $T_s=0.025[\text{s}]$

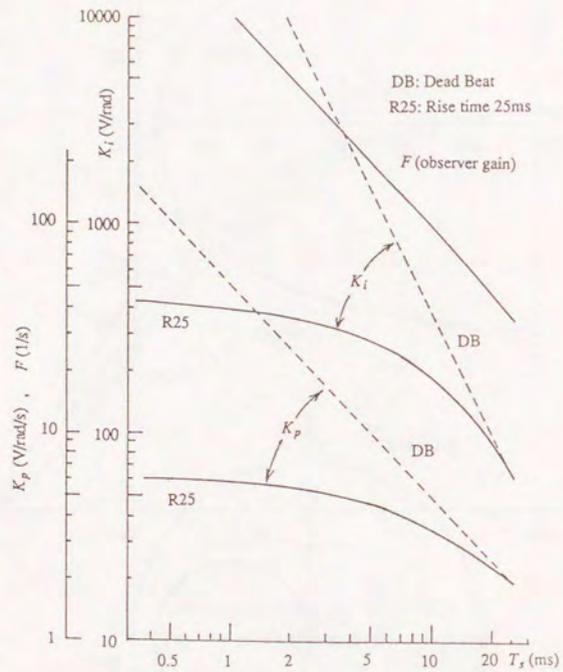
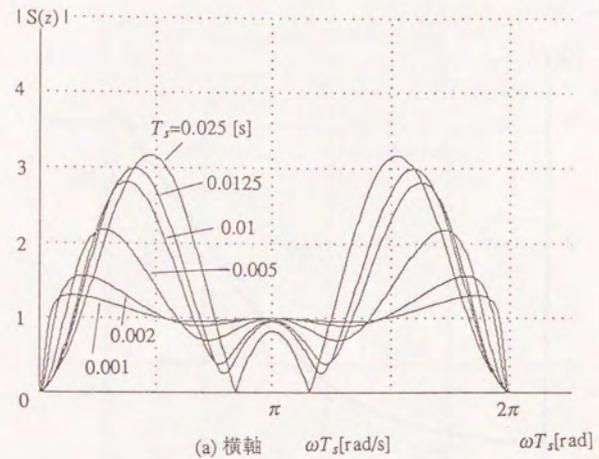
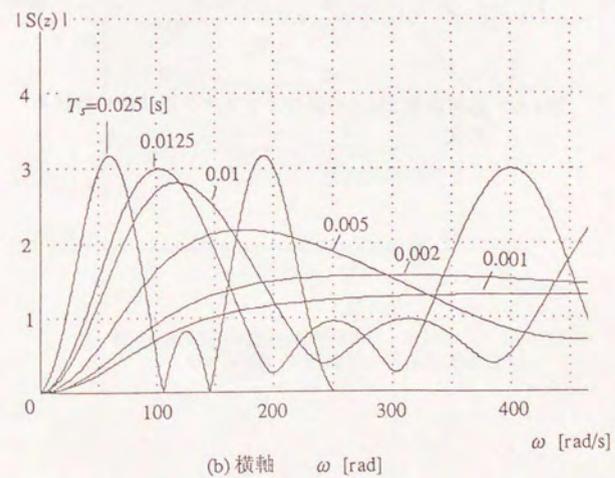


図4.6-5 応答時間25msの系とデッドビート制御系の制御ゲインの比較



(a) 横軸  $\omega T_s$  [rad/s]



(b) 横軸  $\omega$  [rad/s]

図4.6-6 応答時間 25ms の系のパラメータ感度  
( $K_m$ の変動)  $K_m=2.83$ [rad/s]  $T_m=0.095$  [s]

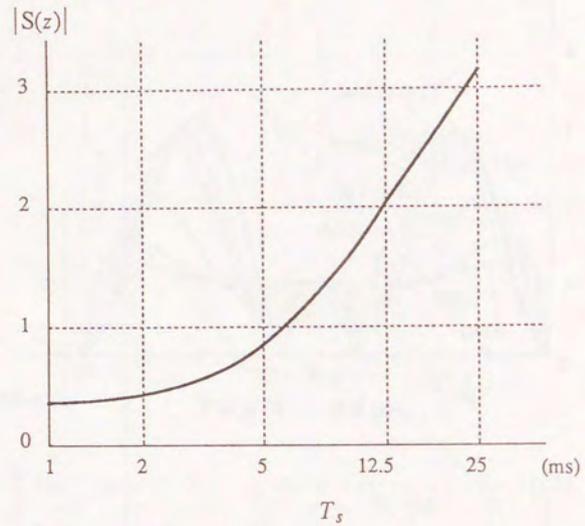


図4.6-7 応答時間 25ms の系のカットオフ周波数における感度

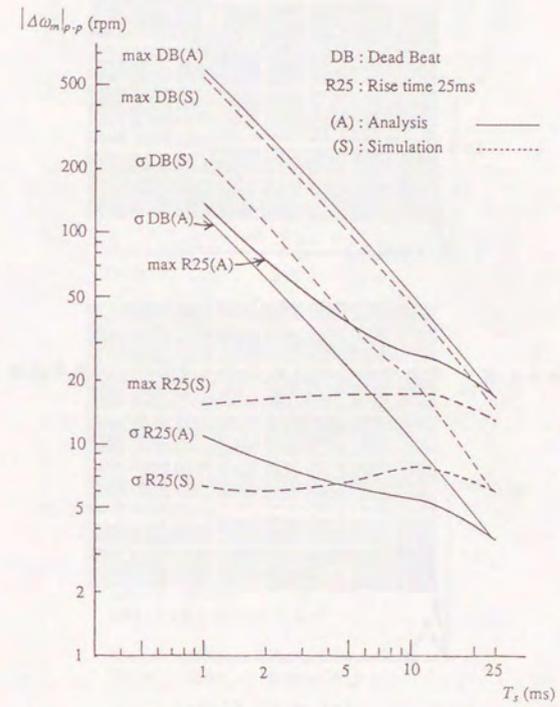


図4.6-8 応答時間25msの系とデッドビート制御系の速度誤差の比較

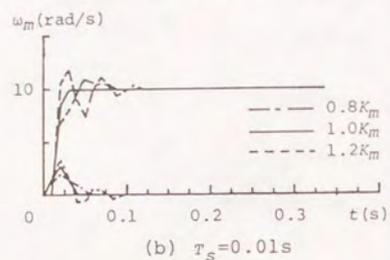
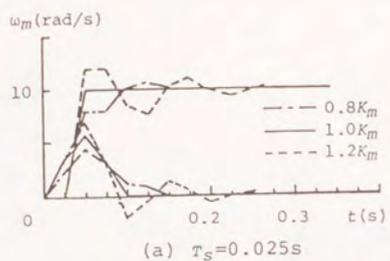


図4.6-9  $K_m$  変動時の予測型オブザーバを用いた系の過渡応答

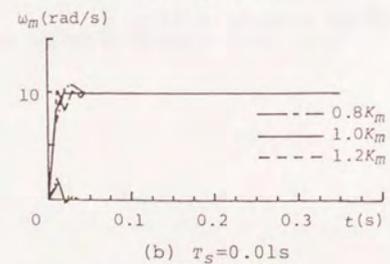
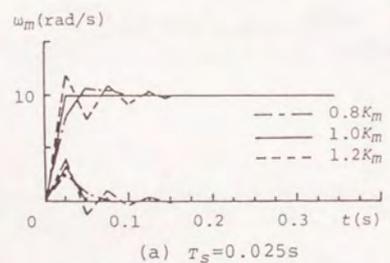
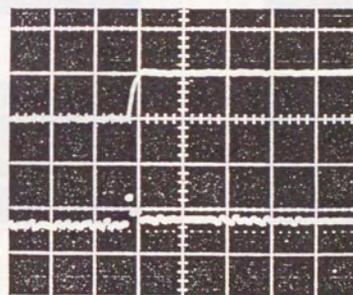
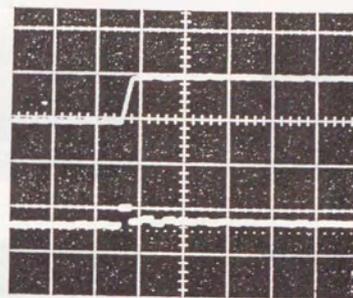


図4.6-10  $K_m$  変動時の理想的I-P制御系の過渡応答



hor: 100ms/div  
var: 100rpm/div

図4.6-11 設計状態での制御応答