

電圧形PWMインバータ駆動電力変換器・電動機系の制御に関する研究

泉, 勝弘

<https://doi.org/10.11501/3111008>

出版情報：九州大学, 1995, 博士（工学）, 論文博士
バージョン：
権利関係：

電圧形PWMインバータ駆動電力変換器・
電動機系の制御に関する研究

泉 勝 弘

①

電圧形PWM インバータ駆動電力変換器・
電動機系の制御に関する研究

1996年1月

泉 勝 弘

目次

1	序論	1
1.1	本研究の背景と目的	1
1.2	論文の概要	7
2	ファジィオートチューニング	10
2.1	特徴量を用いたオートチューニング	10
2.1.1	特徴量	12
2.1.2	ファジィ推論	15
2.1.3	制御対象	17
2.2	シミュレーションによる検討	20
2.2.1	参照モデルによる制御ゲイン	20
2.2.2	ルールの構成	22
2.2.3	シミュレーション結果	28
2.3	実機実験	32
2.3.1	制御回路の構成	32
2.3.2	ソフトウェア構成	33
2.3.3	実験結果	36
2.4	結論	38
3	M系列による同定と制御	39
3.1	伝達関数を用いたオートチューニング	39
3.1.1	M系列信号	40
3.1.2	相互相関関数による同定	41
3.1.3	伝達関数変換	44
3.1.4	制御ゲイン決定	46
3.1.5	制御対象	47
3.2	実機実験	49

3.2.1	制御回路の構成	49
3.2.2	ソフトウェア構成	50
3.2.3	実験結果	54
3.3	結論	61
4	デッドビート制御	62
4.1	2自由度デッドビート制御	62
4.1.1	制御対象	62
4.1.2	コントローラ的设计	63
4.1.3	DSP用コントローラ	66
4.2	シミュレーションによる検討	68
4.3	実機実験	74
4.3.1	制御回路の構成	74
4.3.2	ソフトウェア構成	75
4.3.3	実験結果	79
4.4	結論	86
5	MFS制御	87
5.1	MFS制御器の基本設計	87
5.2	動作点における線形モデル	90
5.2.1	誘導電動機ベクトル制御系のモデル	91
5.2.2	MFS制御系の線形モデル	92
5.3	シミュレーションによる検討	93
5.4	実機実験	103
5.4.1	制御回路の構成	103
5.4.2	ソフトウェア構成	104
5.4.3	実験結果	104
5.5	結論	110
6	結論	111

謝辞	113
参考文献	114
A 2自由度デッドビートコントローラ的设计	i

第 1 章 序論

1.1 本研究の背景と目的

直流電動機は制御が比較的簡単で、サイリスタによる静止レオナード方式として、十数年前まで、可変速運転に広く利用されてきた⁽¹⁾。しかし、直流機はブラシと整流子を持っているため、発火性ガス等の特殊雰囲気中では使用できず、また、ブラシ交換等の保守も必要である。一方、誘導電動機や同期電動機の交流機は比較的構造が簡単で保守も容易であるが、そのトルクや速度の制御には可変周波数の交流電源を必要とするため、速度制御に利用することは困難であった。しかし、1961年 W. McMurray 氏らによって考案された帰還ダイオード付並列形インバータ⁽²⁾により出力周波数を可変できる様になり、広範囲な速度制御が可能となった。当時のインバータは、出力電圧波形が180度もしくは120度の方形波であり、その振幅は一定であった。このため、出力電圧を変化させるためには直流リンク電圧を変化する必要があるが、また、低周波時には方形波ゆえに正弦波電圧との差が顕著に現れ、交流電動機のスムーズな制御は困難であった。

最近、パワートランジスタやIGBTなどの大容量で高速スイッチングが可能な自己消弧形電力用半導体素子の開発により、PWM (パルス幅変調) 技術を使用した電圧形インバータは高効率で高速制御が可能となった。さらに、マイクロプロセッサなどのLSI技術の進歩により、任意の波形の電圧・電流を発生することができ、複雑な制御方法も実現可能となってきている。

インバータによる交流電動機制御系では、三相交流電力を直流に変換し、更に、これを任意の周波数と振幅を持つ交流に変換して電動機に供給している。電動機電流をトルク成分電流と励磁成分電流に分離して制御するいわゆるベクトル制御を用いた交流電動機速度制御系⁽³⁾は、回転角検出回路(ロータリエンコーダ等)、電流検出回路(ホールCT等)、プロセッサを用いた制御回路、操作量出力部としてPWM発生回路および電圧形インバータで構成されること

が多い。この制御系では、速度指令と回転角検出回路からの電動機速度とでPI演算等を用いて速度制御が行われている。速度制御演算の出力であるトルク電流指令は電動機二次磁束の座標上(回転座標系)で表された値であり、これと励磁電流指令を静止座標系へ座標変換することで、三相の電動機一次側電流指令が演算され、この電流指令に基づいて、インバータの各スイッチを動作させるPWM信号を発生する。したがって、ベクトル制御により交流機の瞬時トルクを高速に制御するためには、電流の位相まで細かく制御する必要があり、電動機の電流を指令値にできるだけ迅速にかつ精度よく追従させることが必要となってくる^{(4)~(6)}。このような電流制御系は電動機だけではなく、パワーエレクトロニクスが使用されている数多くの装置にもみられ、電力系統の高調波抑制のために使用されているアクティブフィルタ⁽⁷⁾やコンピュータに使用されている無停電電源⁽⁸⁾⁽⁹⁾はその一例である。それゆえ、電流制御系の高精度・高速化は多方面で要求されている。

電動機電流制御で使用されるPWM信号作成法は大別して以下のように分類できる。

(1) 瞬時電流追従方式

電流指令を三相で出力し、直接検出してきた電流とヒステリシスコンパレータで比較してPWM信号を得る方式で、高速な電流制御が可能である。この方式では、スイッチング周波数が動作状態によって変動するので、スイッチングによるリップルを除いた電動機の電流値(平均電流値)を正確に検出することが難しい。また、出力電圧パターンを制御装置は知ることができないので、電圧センサによる出力電圧の計測が必要であり、電流サンプリング同様に正確な値を得るのは容易ではない。

(2) 静止座標系での電流制御と三角波比較PWM制御を組合せた方式

三相の電流指令と直接検出してきた電流でPI制御等を行い、この出力と三角波を比較してPWM信号を作成する方式で、高調波電流成分の抑制が可能であり、純ハードウェアで制御系を構成できる。また、周期的スイッチングのために電流・電圧のサンプリングは比較的容易である。しかし、静止座標系での制御であるために、電動機電流を正弦波指令に追従させる必要があり、PI制御ゲインの選定が重要となる。

(3) 回転座標系での電流制御と三角波比較PWM制御を組合せた方式

検出してきた電流を座標変換して得られるトルク成分，励磁成分電流とこれらの指令値でPI制御等を行う方式である。これは，原理的に回転座標系での直流制御になり，また二軸間の干渉に対する補償が容易になるので，高精度の電流制御が容易になる。しかし，ソフトウェアの演算時間やインバータのスイッチング周期等により，PI制御だけでは過渡的に十分な電流追従特性が得られない。

(4) 瞬時電圧ベクトル方式

高速なプロセッサを用いてトルク演算を行い，最適電圧ベクトルを選択する方式で，高速なトルク制御が可能である。しかし，PWM信号の分解能が制御周期以下にはできないため，狭いパルスが要求される低速と高速域では電流制御の行き過ぎ量が発生し，トルクリップルも多くなる。

この電流制御系の高速化に関する研究がいくつか行われている。文献(4)では，アナログ回路による静止座標系でのPI制御とヒステリシスコンパレータによる瞬時値制御を常時平行して行っている。この方式では両方の特性を兼ね備えた制御ができるが，経年変化等のアナログ制御回路の欠点を有している。また，PI制御に電動機電流がそのまま入力されているため，インバータのスイッチング作用による脈動電流により，制御ゲインの大きさが制限され，PI制御の利点が反映されにくい。文献(5)では2種類の制御法を行っている。まず，指令電流急変時には回転座標系でのハイゲイン制御を使用し，ゆっくりした変化時には回転座標系でのPI制御をプロセッサ上で切り替えて使用している。次に，指令電流急変時にはアナログ回路による瞬時電流追従方式を使用し，ゆっくりした変化時にはプロセッサによる回転座標系でのPI制御を切り替えて使用している。この方式では，PI制御と他の制御を切り替えているため，PI制御のI動作で使用する積分項を正しく設定しないと，PI制御への切り替え時にオーバシュートを生じる可能性がある。文献(6)では，回転座標上でのPI制御と電流指令から計算した理想電圧のフィードフォワード項を用いて電流制御を行っている。この方式では，フィードフォワード項の演算で使用するパラメータが電動機のパラメータと一致している必要があるが，電動機のパラメータは温度等により変化するため，フィードフォワード項だけでは

補償できない。これらの文献では電流制御の基本にPI制御を用いているが、そのゲイン決定は試行錯誤的であり、電動機逆起電力等の外乱や電動機巻線抵抗の温度による変化が、さらにゲインの決定を難しくしている。

本論文では、電流制御系の検討を、最も一般的な回転座標系での電流制御と三角波比較PWM制御を組合せた方式に関して行っている。このとき、電流制御系の設計法としてファジィオートチューニングによる方法、M系列信号による方法および2自由度デッドビート制御による方法の三種類の方法について検討している。以下、これらに関するこれまでの研究の概要について述べる。なお、本論文ではPID制御のかわりにI-PD制御を用いている。これらはいずれも積分、比例、微分動作で構成されているが、PID制御は目標値と制御量の差に対してこれらの演算を行うのに対して、I-PD制御では積分動作のみ目標値と制御量の差に適用し他の動作は制御量にのみ適用する構成である。この構成により、PID制御は除法的補償、I-PD制御は加法的補償であると言われている⁽¹⁰⁾。PID制御では部分的補償はそれなりに適切に設計することができるが、目標値変化に対して即応性を良くしようとする、極・零点相殺を起こして可制御性を損う欠点がある⁽¹⁰⁾。これに対して、I-PD制御は最適サーボ系へ進化的に近づいていけ、極・零点相殺も起こさない⁽¹⁰⁾⁽¹¹⁾利点がある。

一般のPID制御系に対するこれまでの調整法は、制御系の応答波形を観察し、現場での経験則に基づいて調整を繰り返し行う試行錯誤法が主体で、調整に長時間を費やすなど必ずしも簡単でない。この制御応答を改善するために、制御ゲインの調整を自動で行うオートチューニングが用いられている⁽¹²⁾。チューニングには数式モデル法と経験法がある⁽¹³⁾。数式モデル法は時系列データから制御対象の同定を行い、その同定結果を用いて制御系設計法により最適な制御ゲインを決定する方法である。経験法はステップ応答時の立ち上がり時間やオーバシュート量等から直接制御ゲインを修正する方法であり、制御対象を数式モデルで推定する必要がないので、制御対象が既知でなくても適用できる。この経験法に分類されるファジィ推論を用いたオートチューニングは主としてプロセス制御系において行われている^{(13)~(18)}。ファジィ推論の特徴は、一般に幾つかのルールが入力値により0から1までの度合いで常に励起し、これら各ルールによる推論結果の釣合として全体の出力を決定する点に

ある。このため、ファジィ演算を用いると、入出力間の非線形な関係を容易に実現でき、また、修正も簡単になる⁽¹⁹⁾。このファジィ演算のルールを構成するときに熟練者の経験則を用いることができる。ファジィ・オートチューニングにおいては、ステップ応答時の制御量の波形から得られるオーバシュート量、減衰比、制御面積比等の特徴量とゲイン修正値の間の非線形な関係を実現するためにファジィ推論が使用されている^{(13)~(15)}。しかし、従来^{(13)~(15)}の方法では制御量から得られる特徴量のみを使用しているために、オーバシュートがなければ特徴量を求めにくい。このため、目標オーバシュート量が0%の場合、制御応答の改善は難しい。

ファジィ・オートチューニングの様な経験法では制御対象が既知でなくても適用でき、応答も改善されるが、得られた制御ゲインが最適とは限らない。制御対象の動特性を同定し、それに基づき制御定数を決定する方法では、制御対象の数式モデルを必要とするが、最適な制御系を設計することが可能である。離散時間系はパルス伝達関数、または、(単位)インパルス応答で表現される⁽²⁰⁾。これらの中で、パルス伝達関数は少ないパラメータで制御対象を表現できるが、そのパラメータの推定に多くのメモリと複雑な計算を必要とし、応答の速い制御対象の同定には適用しにくい⁽²¹⁾。M系列信号を用いて相互相関関数によりインパルス応答を同定する方法は、高速な演算には適しているが、M系列信号が真の白色信号ではないことによる誤差がインパルス応答の推定値に生じる⁽²²⁾。それゆえ、この誤差をなくすために、試験信号を加工して制御対象に加える幾つかの方法が提案されているが⁽²³⁾⁽²⁴⁾、いずれも、M系列信号の加工に多くの演算を必要とする割には、精度はあまり改善できていないように思われる。一方、連立一次方程式を解くことによりインパルス応答を正確に求める方法があるが、多量の積・和演算を必要とする点に問題がある⁽²²⁾。また、パルス伝達関数からs領域の伝達関数への変換と制御定数決定法にも、幾つかの方法が提案されている^{(25)~(28)}。しかし、これらの方法では式が複雑であるため、そのまま演算したのでは制御用プロセッサの負荷となる。以上述べたように、同定と制御定数決定には相当の量の演算を必要とするので、プロセス制御系に比べ応答の速いパワーエレクトロニクス分野への適用ではシミュレーション解析はなされているものの⁽²⁹⁾、実際のシステムに適用

した例はあまり見当たらない。

I-PID 制御より更に高速な電流制御を実現するために、デッドビート制御の応用が考えられる。デッドビート制御は制御対象が既知の系に使用されているが、制御対象の変動や外乱に対して極度に制御性能が低下する欠点がある。このとき問題となるのは、目標値追従特性と外乱除去特性であるが、通常のPID 制御ではこの二つを同時に満足させることはできないので、その二つの特性を独立して考慮するために、2個の補償器を有する制御系が考えられている⁽³⁰⁾⁽³¹⁾。最近、この2自由度の考え方をデッドビート制御に適用した方法が提案されている^{(32)~(34)}。文献(32)は設計時に使用する評価関数に目標値と制御量の差および操作量変動分の二乗和を用いていることから、適当な整定段数を指定すれば、操作量の飽和は起きない。しかし、操作量の飽和が起らないように整定段数を多めに設定すれば応答は遅くなる。また、リアルタイムな整定段数の変更では、制御系の再設計に時間を要し高速制御には向かない。文献(33)も設計時に使用する評価関数に目標値と制御量の差および操作量変動分の二乗和を用いていることから、過大な操作量は発生しない。しかし、その分応答が犠牲になる可能性がある。文献(34)では前二者と異なる評価関数を使用しているので、操作量の飽和は起こりうるが、制御対象の変動やモデル化誤差を意識して設計する方法である。電流制御系では広範囲な制御を行うため、操作量の限度一杯で制御を行うことが多く、操作量の変動分を考慮して制御系を設計しても飽和が起こる可能性がある。このため、電流制御系では、文献(34)の方式が適していると思われる。しかし、この方式により設計された制御演算をそのまま電流制御系に適用したのでは、2自由度にしたためにコントローラの次数が大きくなり、多くの演算を必要とする問題点がある。

これまで主として電流制御に用いる同定・制御法に関する研究の背景を述べたが、その上位制御であるトルク制御には、ベクトル制御が使用されている。しかし、誘導電動機の二次抵抗が温度によって変化することにより、ベクトル制御によって演算された二次磁束に誤差を生じ、トルクとその指令値が一致しなくなり、制御応答が悪くなる。また、電動機に接続される負荷の変化によっても速度応答が劣化する。このため、さらに上位の速度制御系に、外乱トルクの抑圧を目的とした負荷トルクオブザーバ⁽³⁵⁾や2自由度制御⁽³⁶⁾の適

用が報告されている。これらとは別の制御法であるが、古田ら⁽³⁷⁾⁽³⁸⁾によって提案されたモデル追従サーボ (MFS) 制御は、モデルのステップ応答に最適に追従し、定値外乱やパラメータ変動に対してロバストであるという報告がなされており注目に値するが、評価関数の重み係数に関する指針が示されていない。牟田らは⁽³⁹⁾、MFS 制御のデジタル制御則を導出し、直流電動機の制御に応用しているが、従来の PI 制御との比較については論じていない。一方、隅元らは⁽⁴⁰⁾、Exact Model Matching 手法⁽⁴¹⁾に基づいたモデル規範形適応制御のベクトル制御誘導電動機系への応用を試みているが、MFS 制御と比べ制御系の構成や設計法がやや複雑である。

1.2 論文の概要

本論文は、筆者が行った研究を^{(42)~(47)}、電流制御系の高性能化と電圧形インバータ駆動誘導電動機ベクトル制御系の速度制御についてまとめたものである。

まず第 2 章では、直流電流制御系で、I-PD 制御ゲインのファジィ・オートチューニングについて述べている。本論文で使用している方法は、文献 (15) で述べられている特徴量を使用する直接調整法の一つであり、制御系のステップ応答より得られる一定時間間隔のデータから特徴量を求め、ファジィ推論によって I-PD ゲインの修正量を求め、現在のゲイン値に掛けて修正を行うオートチューニングである。電流制御等の速い応答を得るために行われている制御では、制御量にオーバシュートがなくても、操作量には振動を持つ可能性がある。このため、オーバシュートの無い制御応答でも、操作量から制御性能の評価を行うことが可能と思われる。そこで、オーバシュートの無い制御応答が必要な場合でも応答が改善でき、1 回のステップ応答から得られる波形情報でオートチューニングができるように、新たに操作量の特徴量を用いた I-PD 制御ゲインの修正法を提案している。

このチューニング法を電流制御系に適用した計算機シミュレーションを行い、従来の制御量と目標値だけで特徴量を算出する場合⁽¹³⁾と比較している。ここで演算を高速に行うために、ファジィ変数は高さ 0.5 で交わる三角型、ファジィ

推論には代数積-加算-重心法，非ファジィ化には高さ法を使用している。さらに，過度のゲイン変更を押さえるために，ゲイン修正量を0.5から2までに制限している。また，DSPを用いた電流制御の実験装置を構成し，本方式の有効性を確認している。

このとき，インバータ(DCチョッパ)に高効率のために導入された電力用半導体素子のスイッチング作用により，負荷の電圧は方形波状になり，電流は脈動している。このため，電圧・電流をそのままサンプリングしたのでは正しい値を得ることは難しい。そこで，V/Fコンバータを電流検出に用いて，これにより電流値をパルスに変換してカウントし，スイッチング周期でサンプルして前回との差により，高速で正確な電流の平均値を求める方式を提案している。

第3章では，システムの同定，伝達関数変換，制御定数決定を行うオートチューニング方式とこれらの直流電流制御系への応用について述べている。システム同定は，最も演算の単純なM系列信号を用いた相互相関関数により重み系列を推定することにより行い，M系列の長さとは異なる個数の重み系列を正確に計算する式を導出している。この重み系列をもとに離散時間から連続時間への伝達関数変換，部分的モデルマッチング法による制御定数決定を行い，文献(25)，(28)から，CPUの演算時間を短縮できる式を誘導している。次に，これらを直流電流制御系へ適用するためのハードウェアとソフトウェアについて述べている。このとき，アセンブリ言語で記述された制御プログラムの高速演算やメモリ消費低減のためにいくつかの工夫を行っている。これにより，低速なプロセッサでも演算が可能なことを実証している。最後に，シミュレーション結果と実験結果の比較検討を行い，本方式がオートチューニング法として有用であることを確認している。

第4章では，直流電流制御系のデッドビート制御について述べている。2自由度補償法による有限整定なロバスト・トラッキング制御を状態量によりパラメータの変化する制御対象に適用している。これは，温度変化により抵抗が変動するため，特に制御対象の変動やモデル化誤差に強いと考えられる設計法を使用する⁽³⁴⁾。文献(34)で述べられている設計法を用いて2自由度デッドビートコントローラを設計するが，これをそのまま使用したのでは高速演算に向かない。そこで，実係数有理関数形の2個の補償要素を有限なインパ

ルス応答形の3個の補償要素に変換し、デッドビート制御演算をすべて積和形式で表現して、DSPでの高速演算を可能にしている。また、この制御系の根軌跡およびシミュレーションにより、コントローラ設計時に使用したパラメータが制御対象と異なるときの挙動を明らかにしている。さらに、流れる電流によって抵抗値が変化する非線形な特性がある電球負荷の電流制御を行う実験システムを構成し、ゲインと状態変数のメモリ上での配置を工夫することにより、積和による高速演算を実現し、PID制御との切り替えも可能にしている。これは、文献(34)の設計法があくまでデッドビート制御であるため、極端な制御対象の変動が発生した場合に制御量の変動が大きくなりうるので、このような場合にPID制御に切り替えることを想定しているからである。この実験結果により、操作量が制限されないときには、応答がデッドビート特性であることを示している。また、操作量が飽和した場合のシミュレーションと実験による応答を示し、たとえ操作量が飽和しても、前章のI-PD制御と同程度の応答を得ることが可能であることを示している。

第5章では、MFS制御の誘導電動機ベクトル制御系への応用について、理論および実験によりその有効性を述べている。まず、ベクトル制御が理想的であると仮定することにより、MFS制御器の基本設計を行っている。次に、MFS制御器のゲイン選定の指針を得るために、二次抵抗変化の影響を考慮した線形モデルを提案し、極および零点の軌跡と過渡応答により動特性を考察している。また、従来のPI制御とMFS制御の比較についても論じている。このとき、負荷変動に対する特性を同じとしたとき、MFS制御はPI制御に対し零点を自由に追加できるので、目標値応答の改善がたとえ二次抵抗変化が生じても維持されるか確認している。更に、電圧形インバータ駆動誘導電動機ベクトル制御系にMFS制御を実際に適用し、理論を検証している。

第6章の結論では、本論文により得られた成果を総括している。

第2章 ファジィオートチューニング

本章では、直流電流制御系で、I-PD 制御ゲインのファジィ・オートチューニングについて述べる。本論文で使用している方法は、文献(15)で述べられている特徴量を使用する直接調整法の一つであり、制御系のステップ応答より得られる一定時間間隔のデータから特徴量を求め、ファジィ推論によってI-PDゲインのオートチューニングを行うものである。このとき、オーバシュートのない制御応答が必要な場合でも応答が改善でき、1回のステップ応答から得られる波形情報でオートチューニングができるように、新たに操作量の特徴量を用いたI-PD制御ゲインの修正法を提案する。このチューニング法を電流制御系に適用した計算機シミュレーションを行い、従来の制御量と目標値だけで特徴量を算出する場合⁽¹³⁾と比較する。また、DSPを用いた電流制御の実験装置を構成し、本方式の有効性を確認する。さらに、インバータ(DCチョッパ)に高効率のために導入された電力用半導体素子のスイッチング作用により、負荷の電圧は方形波状になり、電流は脈動しているため、V/Fコンバータを用いた高速で正確な電流の平均値を求める方式を提案する。

2.1 特徴量を用いたオートチューニング

オートチューニングの構成を図2.1に示す。このシステムでは、ステップ応答時の制御対象や制御ループの入出力信号をもとに、オートチューニング部においてコントローラのゲインを自動的に修正している。ファジィ推論は、この入出力信号から得られる特徴量とコントローラのゲイン修正に用いる修正係数との間の非線形な関係を実現するために使用している。コントローラには、最適サーボ系へ進化的に近づいていけ、極・零点相殺も起こさない⁽¹⁰⁾⁽¹¹⁾I-PD制御を用いている。

図2.2に示すI-PD制御系は次式の連続時間表現で示される。

$$u(t) = \int_0^t \frac{K_i}{T_c} \{r(t) - y(t)\} dt - K_p y(t) - T_c K_d \frac{dy(t)}{dt} \quad (2.1)$$

ここで、 $u(t)$, $r(t)$, $y(t)$:それぞれ操作量, 目標値, 制御量, K_i , K_p , K_d , T_c :それぞれ積分ゲイン, 比例ゲイン, 微分ゲイン, サンプルング周期

上式をサンプルング周期で離散化するとサンプルング点 k では

$$u(k) = K_i \sum_{i=0}^k \{r(i) - y(i)\} - K_p y(k) - K_d \{y(k) - y(k-1)\} \quad (2.2)$$

となり, 速度型⁽⁴⁸⁾では次式となる。

$$u(k) = u(k-1) + K_i \{r(k) - y(k)\} - K_p \{y(k) - y(k-1)\} - K_d \{y(k) - 2y(k-1) + y(k-2)\} \quad (2.3)$$

実機実験および数値シミュレーションでは上式を用いてI-PD制御演算を行うので, この K_i , K_p , K_d をチューニングにより求める。

オートチューニングでは, 目標値のステップ変化より制御量, 操作量をサンプルング周期ごとに取り込み, オーバシュート量, 減衰比, 制御面積比等の特

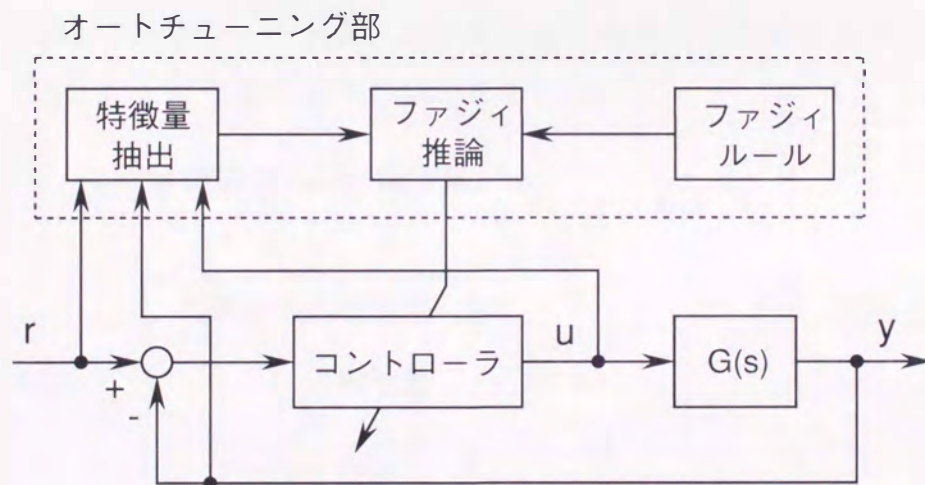


図 2.1 オートチューニングシステム

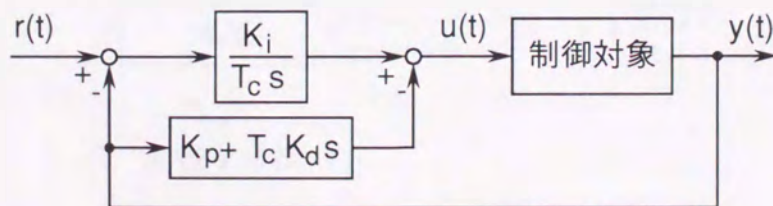


図 2.2 I-PD 制御系

微量を計算する。次に、求めた特徴量から調整ルールを用いてファジィ推論を行う。ファジィ推論を特徴量と K_i , K_p , K_d それぞれの修正係数の非線形な関係を実現するために使用していて、調整ルールは「IF ~ THEN ~」(もし ~ ならば ~ せよ) の形式で記述している。この調整ルールの構成は試行錯誤により行うため、構成によっては、システムが不安定に成る方向に制御ゲインが変化する可能性がある。このため、急激な制御ゲインの変動が起らないように、修正係数は1を基準として下限を0.5, 上限を2.0としている。この制限を行っても初期のゲイン修正が数回多くなる程度で、後半の修正には影響を与えないと、図2.12の根軌跡から推測される。最後に、ファジィ推論により求めた修正係数と前回のゲイン値との乗算により、新しいゲイン値を設定する。この様子を図2.3に示す。

2.1.1 特徴量

特徴量は目標値のステップ変化より制御量, 操作量をサンプリング周期ごとに取り込み, オーバシュート量, 減衰比, 制御面積比等を計算した値であり, ステップ応答の制御性能を表す指標である。

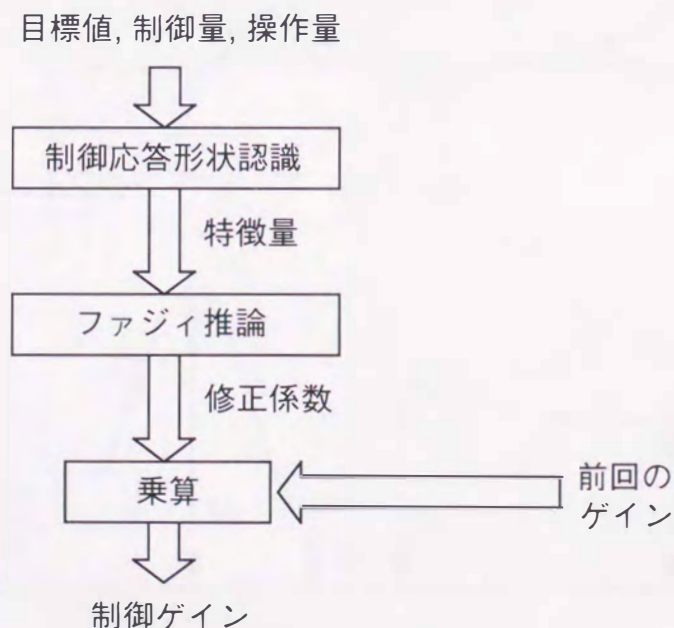


図 2.3 オートチューニングの概要

特徴量として従来使用されてきた方式と本研究で新しく提案する方式を以下に示す。

2.1.1.1 従来方式

従来、特徴量を求めるのに目標値と制御量だけが用いられてきた(13)~(15)。この特徴量には応答波形の極値から求める方法と、半サイクルごとの制御面積値から求める方法がある。雑音に対する強さを考慮して、図2.4に示すように半サイクルごとの制御面積値より、次式で定義される三つの特徴量が用いられている(13)。

$$\text{オーバーシュート量: } OV = \frac{A_2}{A_1} \quad (2.4)$$

$$\text{減衰比: } DP = \frac{A_3 + A_4}{A_2 + A_3} \quad (2.5)$$

$$\text{整定時間比: } R = \frac{\Sigma A_i(\text{前回値})}{\Sigma A_i(\text{今回値})} \quad (2.6)$$

この方式の欠点は、整定時間比が前回値を使用するので、1回のステップ応

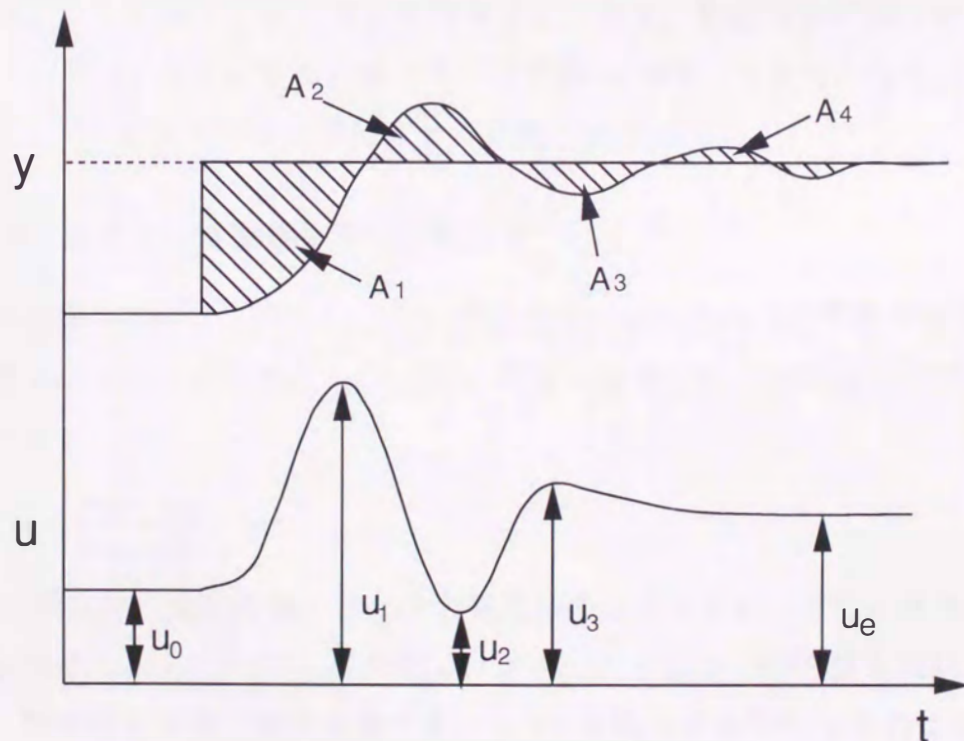


図 2.4 制御応答波形の特徴量

答では制御ゲインの修正ができない点にある。また、ステップ応答が振動しないとこれらの特徴量を計算できず、ゲインの修正を行うことができない。このため、オーバシュートのない制御応答が必要な場合には適用しにくい。

2.1.1.2 操作量の特徴量を用いる場合 I

操作量においては、例えば電球負荷のようにステップ応答の後パラメータが時間と共に変化するような制御対象であれば、半サイクルごとの面積値が誤差を持つことになる。このため、特徴量のオーバシュート量と減衰比には2.1.1.1の場合と同じく面積値を用いるが、整定時間比に変えて操作量のステップ応答前の値 u_0 、極大値 u_1 および極小値 u_2 により、次式で定義する操作量の特徴量 (U_a) を用いる。

$$U_a = \frac{|u_1 - u_2|}{|u_1 - u_0|} \quad (2.7)$$

U_a が大きいと制御量の立上りが急峻になるため、制御系の応答速度を示すと考えられる。また、この特徴量に前回値を用いていないので、1回のステップ応答により制御ゲインの修正ができる。しかし、制御対象が積分型のときには、 u_0 とステップ応答後の操作量の定常値 u_e は共に0となるため、この場合のファジィ変数の位置は移動させる必要がある。

2.1.1.3 操作量の特徴量を用いる場合 II

この場合も2.1.1.2と同様に、2.1.1.1の整定時間比に変えて操作量の極大値 u_1 、極小値 u_2 、2回目の極大値 u_3 により、次式で定義される操作量の特徴量 (U_b) を用いる。

$$U_b = \frac{|u_3 - u_2|}{|u_2 - u_1|} \quad (2.8)$$

U_b は操作量の最初の変化量と次の変化量の比であるから、この値は操作量の減衰特性を示している。この値ではステップ応答前の操作量を使用しないので、制御対象が積分型のときでもファジィ変数の移動を行う必要はない。

2.1.2 ファジィ推論

ファジィ変数の集合では、変数の幅を狭くすると適用される規則がほとんどないため推論できない。一方、変数の幅を広くすると適用される規則は多くなるが、お互いに影響し合って平均的な結果しか得られない。また、ファジィ変数には種々の形があるが、もっとも演算が簡単なのは直線で表される三角型である。このため、高さ0.5で交わる三角型ファジィ変数の集合を用いることにより、高速に演算でき、良い推論を得ることができると言われている⁽¹⁹⁾。そこで、メンバーシップ関数は図2.5に示すように、前件部、後件部とも高さ0.5で交わる三角型を使用している。ここで、ZOは適正、PBは大きい、NBは小さいを表し、CI、CP、CDはそれぞれゲイン K_i 、 K_p 、 K_d の修正係数を表している。

ファジィ推論には min-max-重心法と代数積-加算-重心法がある⁽¹⁹⁾。min-max-重心法はファジィ変数の適合度より下部を集めた全体の面積を求める必要があり高速演算には向かない。代数積-加算-重心法では、演算が簡単な代数積と加算を用いているため高速演算が可能である。そこで、ファジィ推論は、ルールごとに各特徴量に対する適合度を求め、これらの適合度の代数積と後件部のメンバーシップ関数との代数積により行う。

非ファジィ化には重心法、面積法、高さ法等の連続的に出力が変化する方式と、最大高さ法、最大平均法、最大面積法等のある種の最大値を用いて出力を求める方法がある。この最大値を用いて代表点を決める方法では、特定の値を出力する傾向があり⁽¹⁹⁾、きめ細かい推論が行えない。また、重心法等の変

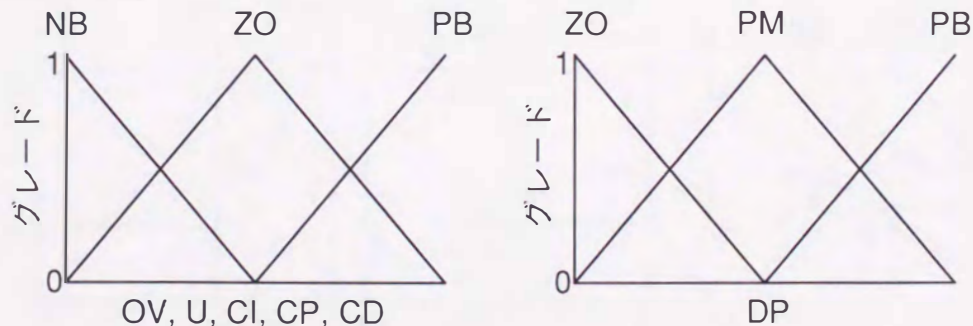


図 2.5 メンバーシップ関数

化する面積を使用する方式では、面積の計算に時間を要し高速演算には向かない。面積法では、ファジィ変数の情報として、変数の面積も必要とする。そこで、きめ細かい修正が行え高速演算が可能な高さ法⁽¹⁹⁾を非ファジィ化に用いる。

例えば、図2.6に示すように適用されるルールが

(1) IF OV=PB and DP=ZO and U=PB

then CI=ZO

(2) IF OV=PB and DP=ZO and U=ZO

then CI=NB

の二つのとき、(1)の条件のときの適合度 ω_{ep} , ω_{dz} , ω_{up} と(2)の条件のときの適合度 ω_{ep} , ω_{dz} , ω_{uz} を求める。次に、それぞれの適合度の積 h_1 , h_2 を

$$h_1 = \omega_{ep} \times \omega_{dz} \times \omega_{up} \quad (2.9)$$

$$h_2 = \omega_{ep} \times \omega_{dz} \times \omega_{uz} \quad (2.10)$$

により求めて、後件部のメンバーシップ関数との積をとり推論結果とする。最後に非ファジィ化法として

$$CI_0 = \frac{z_1 \times h_1 + z_2 \times h_2}{h_1 + h_2} \quad (2.11)$$

により修正係数が得られる。ここで、 z_1 , z_2 はNB, ZOのグレードが1になるときの台の値である。

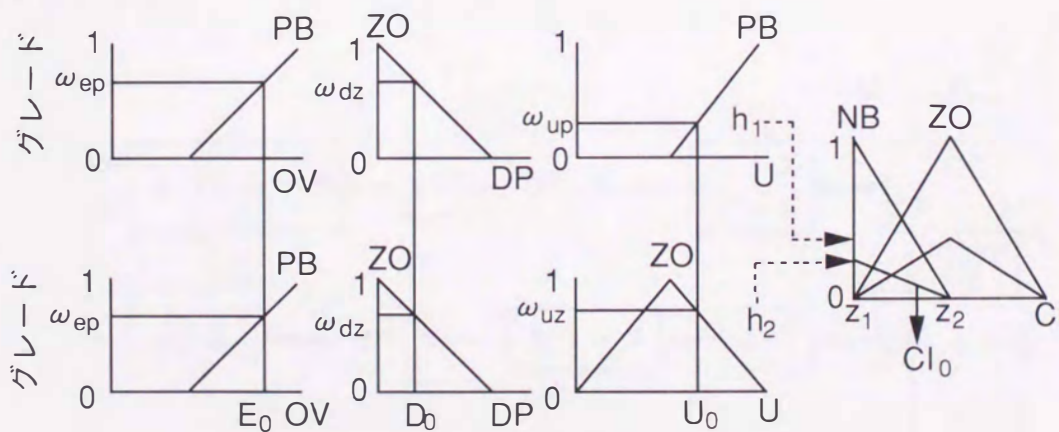


図 2.6 ファジィ推論の方法

2.1.3 制御対象

電流制御系の構成を図2.7に示す。制御対象はDCチョップパ(インバータ)に負荷抵抗 R_d とインダクタンス L_d を接続したもので、入力は負荷の端子電圧 v_d , 出力は負荷電流 i_d である。負荷抵抗として220V, 100Wの電球を15個使用している。制御対象の伝達関数を次式に示す。

$$\frac{I_d(s)}{V_d(s)} = \frac{1}{R_d + L_d s} \quad (2.12)$$

この電流制御系では、DCチョップパの作用により電流は脈動しているため、瞬時値を使用して制御したのでは希望する電流を得るのはむずかしい。そのため、1スイッチング周期中の平均値を検出することが望まれる。しかし、電流値をA/Dコンバータにより入力したのでは値の平均を求める必要があり、複数回のA/Dコンバータ変換時間と平均値を計算する時間が必要である。そこで、このシステムでは電流をV/Fコンバータによりパルスに変換し、カウンタにより電流の積分値を求め、積分値の差により電流値を得ている。

零次ホールド回路、V/Fコンバータによる積分、差による微分を考慮したパルス伝達関数は次式により得られる。

$$G(z) = \frac{1 - z^{-1}}{T_c} \mathcal{Z} \left\{ \frac{1 - \exp(-T_c s)}{s} \frac{1}{R_d + L_d s} \frac{1}{s} \right\} z^{-1} \quad (2.13)$$

ここで、 \mathcal{Z} は z 変換を表す。

上式より、図2.7の離散時間系における検出演算も含めた制御対象の伝達関数は次式となる。

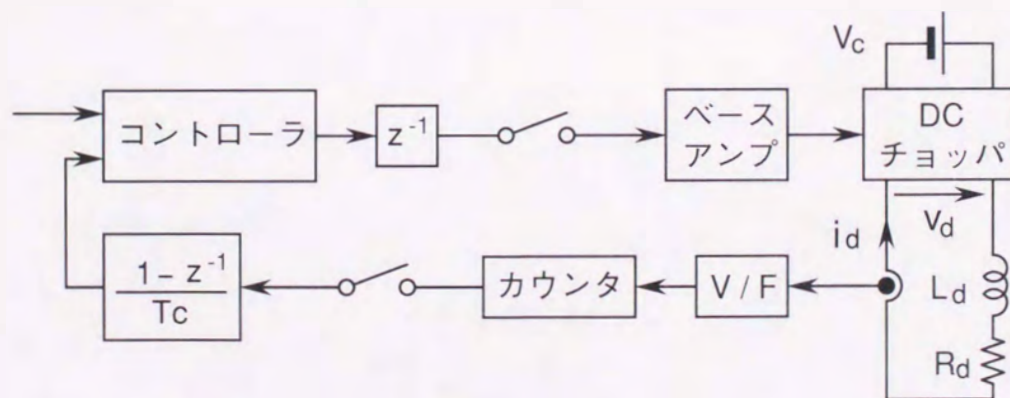


図 2.7 電流制御系

$$G(z) = z^{-2} \frac{b_0 + b_1 z^{-1}}{1 + a_1 z^{-1}} \quad (2.14)$$

ここで,

$$a_1 = -\exp\left(-\frac{R_d T_c}{L_d}\right)$$

$$b_0 = \frac{1}{R_d} \left\{ 1 - \frac{L_d}{R_d T_c} (1 + a_1) \right\}$$

$$b_1 = \frac{1}{R_d} \left\{ a_1 + \frac{L_d}{R_d T_c} (1 + a_1) \right\}$$

このシステムの定数を表2.1に示す。また、負荷電流2A時の定数を用いて計算した a_1 , b_0 , b_1 を図2.8, 2.9に示す。

表 2.1 定数

コンデンサ電圧	V_c	100 V
負荷抵抗 (2A 時)	R_d	8.8 Ω
負荷抵抗 (3A 時)	R_d	17.6 Ω
負荷インダクタンス	L_d	0.075 H
平滑用コンデンサ	C	5600 μF
制御周期	T_c	1.024 ms

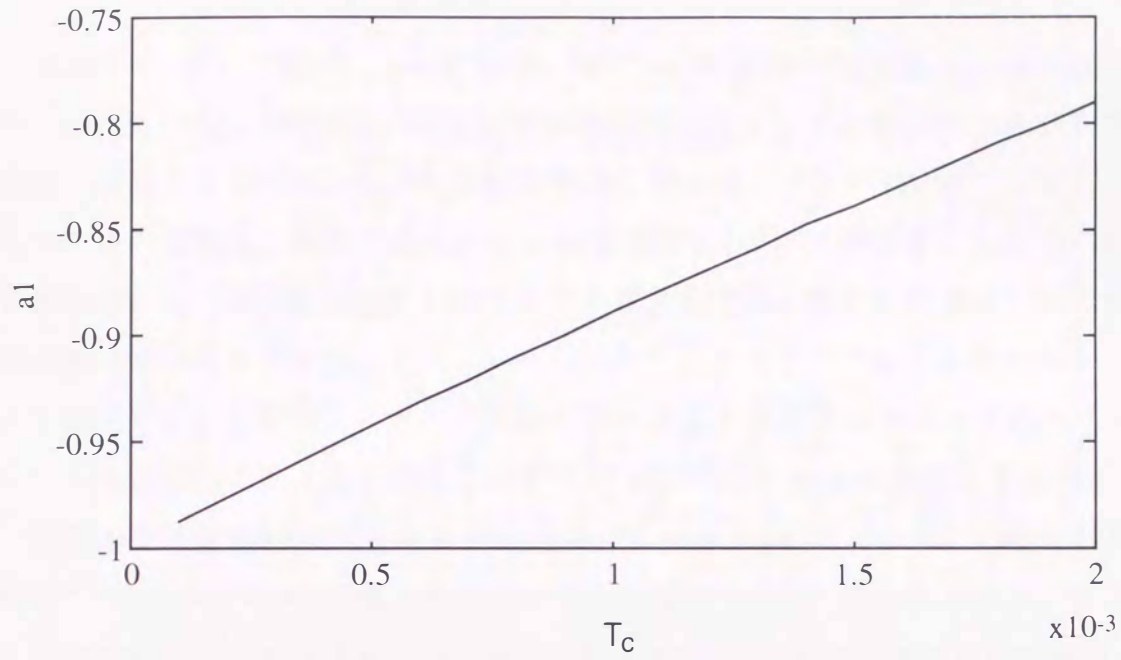


図 2.8 a_1 の計算値

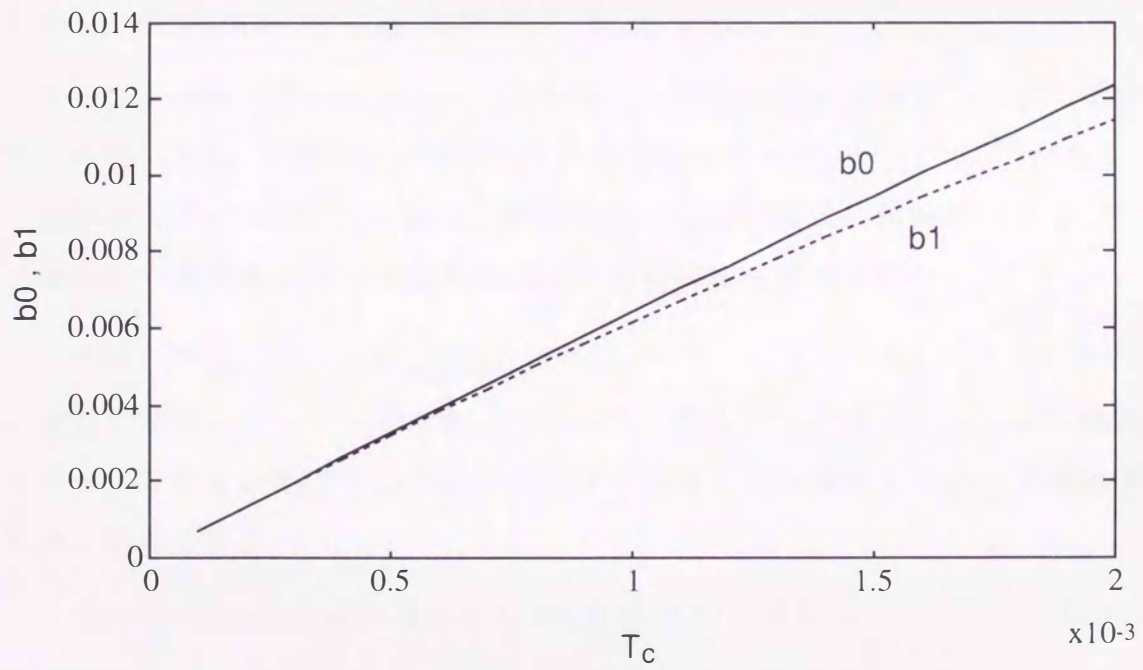


図 2.9 b_0, b_1 の計算値

2.2 シミュレーションによる検討

ファジィ・オートチューニングでは、ステップ応答時の制御量から得られるオーバシュート量、減衰比、制御面積比等の特徴量とゲイン修正値の非線形な関係を実現するためにファジィ推論を使用している。ファジィ推論で必要とするルールの原案は、制御対象のパルス伝達関数とI-PD制御演算により得られる根軌跡から、制御量の振動を抑えるように決定する。整定時間比については前回の値が関係するため、シミュレーションによるステップ応答波形からルールを決定する。その後、ステップ応答の繰り返しによるチューニングのシミュレーションを用いて、より適切な応答を得るためにルールの手直しを行う。

根軌跡は特定のI-PDゲインを中心に描いている。このゲインは、北森氏の系数列⁽²⁷⁾を持つモデル(以後、北森モデルと記述する)に一致するように、サンプル値制御系に対する部分的モデルマッチング法⁽²⁸⁾を用いて求めた値である。

得られた推論ルールを用いて、整定時間比、 U_a および U_b を用いた場合のシミュレーションを行い、これらの方式の比較検討を行う。

2.2.1 参照モデルによる制御ゲイン

サンプル値制御系の制御ゲイン決定法に、制御対象の動特性に関する部分的な知識により、制御ゲインを設定する部分的モデルマッチング法⁽²⁸⁾がある。

部分的モデルマッチング法は、制御対象の伝達関数に制御装置を付加した目標値から制御量までの伝達関数の分母系列表現を参照モデル

$$G_m(s) = \frac{1}{\alpha_0 + \alpha_1 \sigma s + \alpha_2 \sigma^2 s^2 + \dots} \quad (2.15)$$

に調整可能なパラメータの数が許すかぎり、分母の s の低次のほうから係数を合せ、しかも σ ができるだけ正で小さくなるように選定することで構成される。北森モデル

$$\{\alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \dots\} = 1, 1, 0.5, 0.15, 0.03, 0.003, \dots \quad (2.16)$$

を使用すると、行き過ぎ量約10%のステップ応答の系が得られる。図2.10に時間軸を規格化した北森モデルの応答波形を示す。

制御対象の伝達関数 $G(s)$ を

$$G(s) = \frac{1}{g_0 + g_1 s + g_2 s^2 + g_3 s^3 + \dots} \quad (2.17)$$

の分母系列表現とすることにより，図2.11のサンプル値制御系のI-PDゲインは以下の手順で決定できる(28)。

サンプリング周期を T_c とする変数 σ の方程式

$$\begin{aligned} & (g_2 + T_c g_1 + T_c^2 g_0) \alpha_4 \sigma^3 \\ & + \left(-g_3 + \frac{7}{12} T_c^2 g_1 + \frac{1}{4} T_c^3 g_0\right) \alpha_3 \sigma^2 \\ & + \left(-g_3 - \frac{7}{12} T_c g_2 + \frac{1}{18} T_c^3 g_0\right) T_c \alpha_2 \sigma \\ & + \left(-\frac{1}{3} g_3 - \frac{1}{4} T_c g_2 - \frac{1}{18} T_c^2 g_1\right) T_c^2 \alpha_1 = 0 \end{aligned} \quad (2.18)$$

の根の中で，正で最小の根を

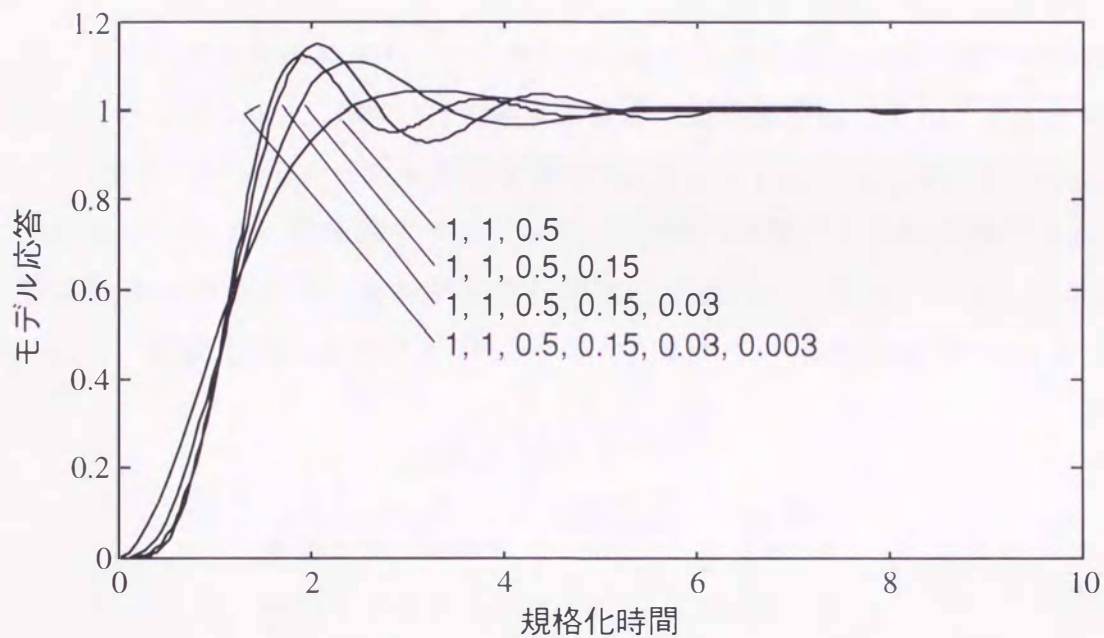


図 2.10 参照モデルの応答

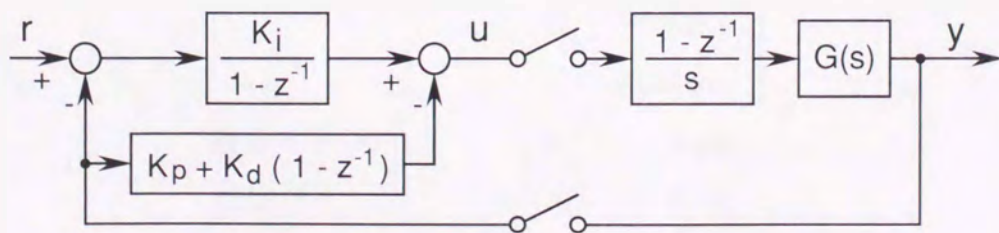


図 2.11 サンプル値系のI-PD制御

$$K_i = \frac{3g_2 + 3T_c g_1 + T_c^2 g_0}{3\alpha_3 \sigma^3 + 3T_c \alpha_2 \sigma^2 + T_c^2 \alpha_1 \sigma} \quad (2.19)$$

$$K_p = K_i \alpha_1 \sigma - g_0 \quad (2.20)$$

$$K_d = K_i \alpha_2 \sigma^2 - g_1 + \frac{1}{2} T_c K_p \quad (2.21)$$

に代入することにより，積分ゲイン K_i ，比例ゲイン K_p ，微分ゲイン K_d を求めることができる。

ファジィ推論のルール原案を決めるときに使用する根軌跡の中心ゲインは，サンプル値制御系に対する部分的モデルマッチング法から求め，この値を表 2.2 に示す。

2.2.2 ルールの構成

図 2.11 に示す制御系に関し， K_i ， K_p ， K_d をパラメータとしたときの根軌跡を図 2.12 (a)，(b)，(c) に示す。この根軌跡から，制御量の振動を抑えるように，ファジィ推論に用いるルールの原案を決めた。ここで， K'_i ， K'_p ， K'_d は，サンプル値制御系に対する部分的モデルマッチング法⁽²⁸⁾を用いて求めた値である。

離散時間制御系では，実軸から離れ単位円に近づくとオーバシュートが大きくなり，減衰比は小さくなる⁽⁴⁸⁾。そこで，すべての根が単位円の中心に近

表 2.2 I-PD ゲイン

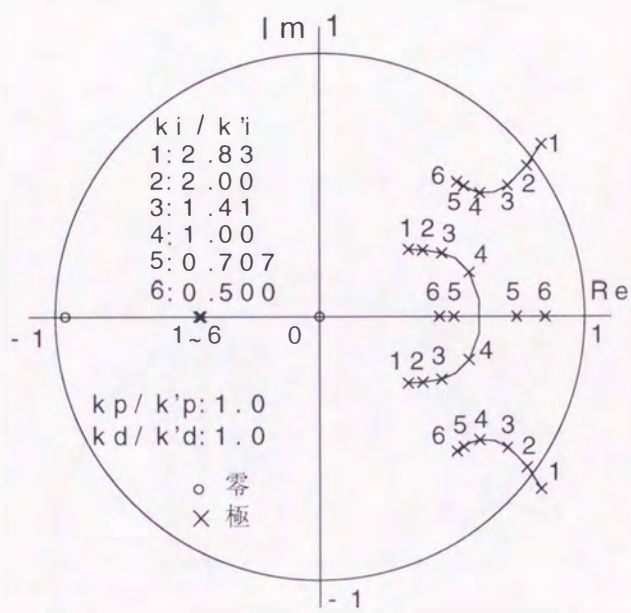
負荷抵抗 (2A 時)	R_d	8.8	Ω
負荷インダクタンス	L_d	0.075	H
制御周期	T_c	1.024	ms
	a_1	-0.8868	
	b_0	0.006561	
	b_1	0.006304	
積分ゲイン	K'_i	10.08	
比例ゲイン	K'_p	33.02	
微分ゲイン	K'_d	16.85	

づくようにルールを決定した。例えば，図 2.12 (a) の点 2 の場合 OV ， DP ， U_b はそれぞれ PB，PB，PB であり，ゲイン K_i を小さくすればよいので， U_b のルール RI では CI を NB とした。ただし，整定時間比については前回の値が関係するため，シミュレーションによるステップ応答波形からルールを決定した。その後，ステップ応答の繰り返しによるチューニングのシミュレーションを用いて，より適切な応答を得るためにルールの手直しを行った。その他の特徴量を用いたときも同様にしてルールを構成した。ここで使用したファジイ変数のグレード 1 における台の値を表 2.3 に示す。

以上により，ファジイオートチューニングのルールを得ることができる。表 2.4 には整定時間比を用いた場合のルールを示し，表 2.5 には提案方式のルールを示す。表 2.5 には，チューニング目標としてオーバシュートが 4% になる場合と 0% になる場合を示している。

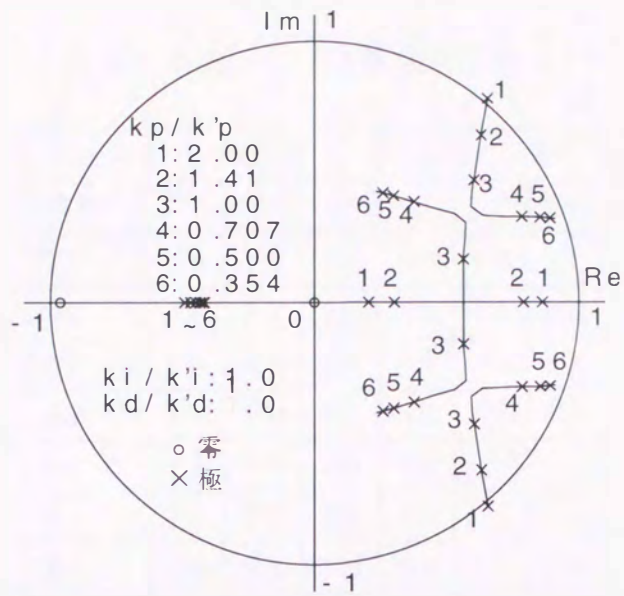
表 2.3 ファジイ変数の台

	NB	ZO	PM	PB
CI, CP, CD	0.5	1	—	2
OV	0.0	0.04	—	0.08
DP	—	0.0	0.3	0.6
R	0.0	1.0	—	2.0
U_a , U_b	0.0	0.4	—	0.8

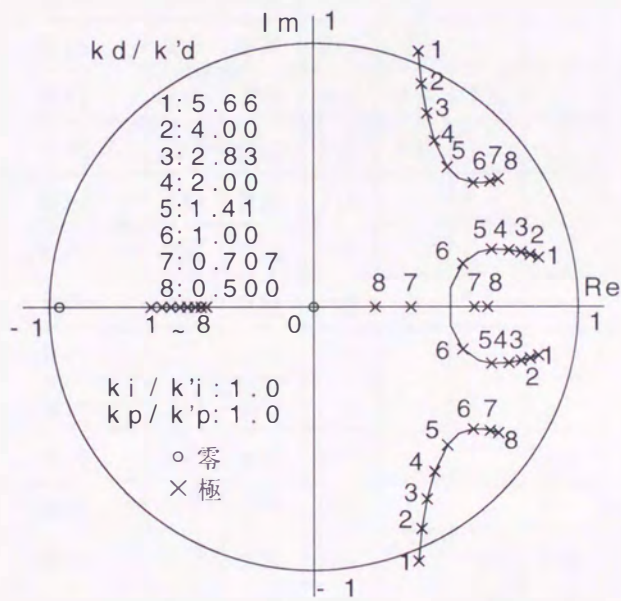


(a) パラメータ k_i

図 2.12 根軌跡



(b) パラメータ K_p



(c) パラメータ K_d

図 2.12 根軌跡

表 2.4 調整ルール (整定時間比 R)

ルール 番号	特徴量			目標 OV 4%		
	OV	DP	R	CI	CP	CD
R1	PB	PB	PB	NB	PB	NB
R2	PB	PB	ZO	NB	PB	NB
R3	PB	PB	NB	NB	PB	NB
R4	PB	PM	PB	NB	PB	NB
R5	PB	PM	ZO	NB		
R6	PB	PM	NB	NB		
R7	PB	ZO	PB	NB		NB
R8	PB	ZO	ZO	NB		
R9	PB	ZO	NB			
R10	ZO	PB	PB	NB		
R11	ZO	PB	ZO	NB		
R12	ZO	PB	NB	NB	PB	
R13	ZO	PM	PB	NB		
R14	ZO	PM	ZO			
R15	ZO	PM	NB			
R16	ZO	ZO	PB			
R17	ZO	ZO	ZO			
R18	ZO	ZO	NB			
R19	NB	PB	PB			NB
R20	NB	PB	ZO			
R21	NB	PB	NB		NB	PB
R22	NB	PM	PB		NB	NB
R23	NB	PM	ZO			
R24	NB	PM	NB		PB	
R25	NB	ZO	PB			NB
R26	NB	ZO	ZO	PB		
R27	NB	ZO	NB			

注) 空欄は ZO を表す

表 2.5 調整ルール (U_a, U_b)

ルール 番号	特徴量			目標 OV 4%			目標 OV 0%		
	OV	DP	U_a, U_b	CI	CP	CD	CI	CP	CD
R1	PB	PB	PB	NB			NB		
R2	PB	PB	ZO	NB	PB		NB	PB	
R3	PB	PB	NB	NB			NB		
R4	PB	PM	PB	NB					
R5	PB	PM	ZO	NB		NB			NB
R6	PB	PM	NB				NB		
R7	PB	ZO	PB	NB					
R8	PB	ZO	ZO						
R9	PB	ZO	NB				NB		
R10	ZO	PB	PB						
R11	ZO	PB	ZO						
R12	ZO	PB	NB						
R13	ZO	PM	PB						
R14	ZO	PM	ZO						
R15	ZO	PM	NB						
R16	ZO	ZO	PB						
R17	ZO	ZO	ZO						
R18	ZO	ZO	NB				PB		
R19	NB	PB	PB						
R20	NB	PB	ZO		NB			NB	
R21	NB	PB	NB	PB		PB	PB		PB
R22	NB	PM	PB						
R23	NB	PM	ZO						
R24	NB	PM	NB				PB		
R25	NB	ZO	PB	PB					
R26	NB	ZO	ZO	PB			PB		
R27	NB	ZO	NB	PB			PB		

注) 空欄は ZO を表す

2.2.3 シミュレーション結果

3組の特徴量を用いたファジィ・オートチューニングのシミュレーションを行い、比較検討する。このとき、表2.5に示すルールで、目標オーバershoot 4%のルールを使用した。また、表2.6にオートチューニングによる制御ゲインの最終値を示す。

図2.13は整定時間比 R を用いたときのシミュレーション結果であり、制御量は数回で望ましい応答になっているが、操作量の波形が同じ波形の繰り返しにならず、制御ゲインの収束に時間を要している。

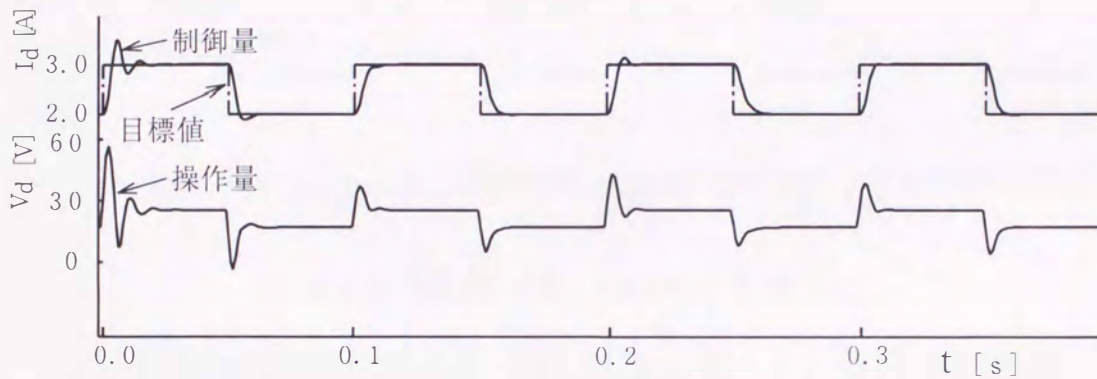
図2.14は操作量の特徴量が U_a のときのシミュレーション結果で、操作量は数回で収束し、特徴量に整定時間比を用いた場合と同様の制御応答が得られている。

図2.15は操作量の特徴量が U_b のときのシミュレーション結果で、制御量は数回で望ましい応答になっている。そして、各ステップ応答の操作量の最大値も同じになり、制御ゲインの収束がすみやかに完了していることがわかる。ただし、初期応答にオーバershoot有と無で制御ゲインの収束値が異なるため(表2.6参照)、(a)、(b)図の後半で応答波形が異なる。これはファジィ推論の非線形性によると思われる。

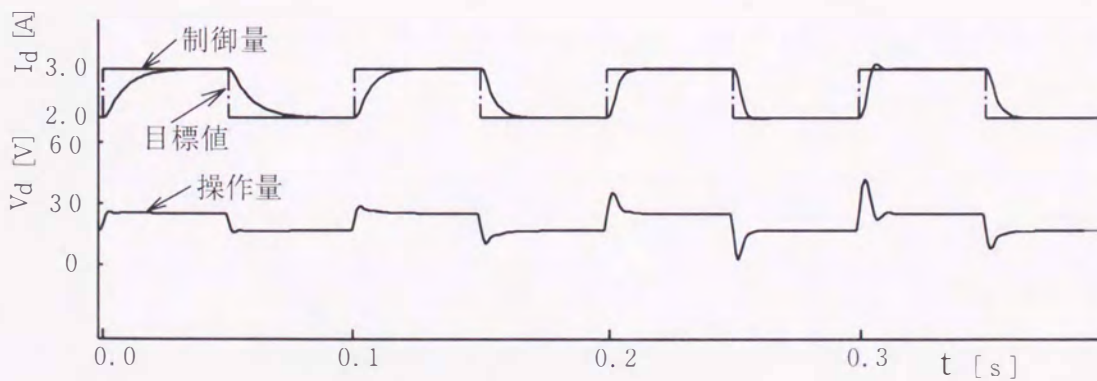
これらの応答の後半は同様であるが、整定時間比ではゲインの収束に時間を要し、 U_a では負荷抵抗の変化が大きい場合(表2.1参照)ファジィ変数の移動を行う必要がある。そこで、この欠点がない U_b を操作量の特徴量として用いたファジィ推論を実機システムに適用することにした。

表 2.6 制御ゲインの最終値

特徴量	初期 OV	K_i	K_p	K_d
R	有	10.3	33.6	9.4
R	無	11.0	31.8	10.7
U_a	有	8.6	31.9	15.6
U_a	無	8.9	31.9	15.6
U_b	有	11.3	32.1	13.2
U_b	無	10.2	31.9	15.6

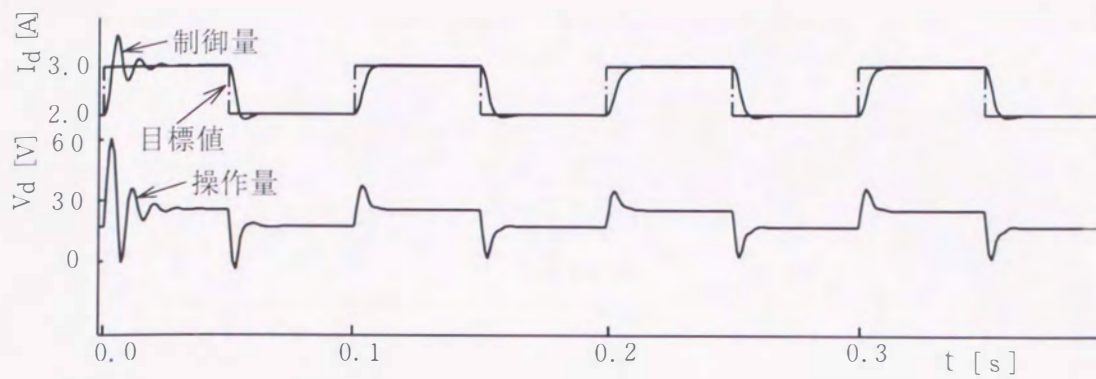


(a) 初期応答にオーバシュート有

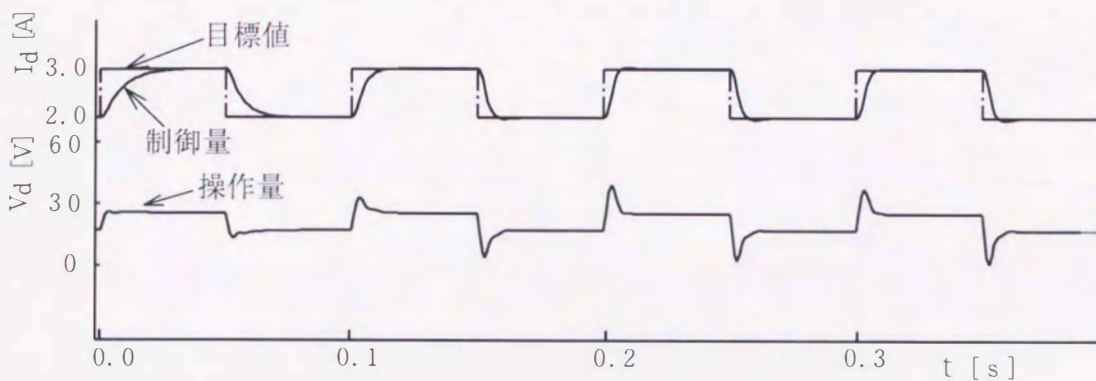


(b) 初期応答にオーバシュート無

図 2.13 特徴量 OV , DP , R を用いた場合のシミュレーション結果

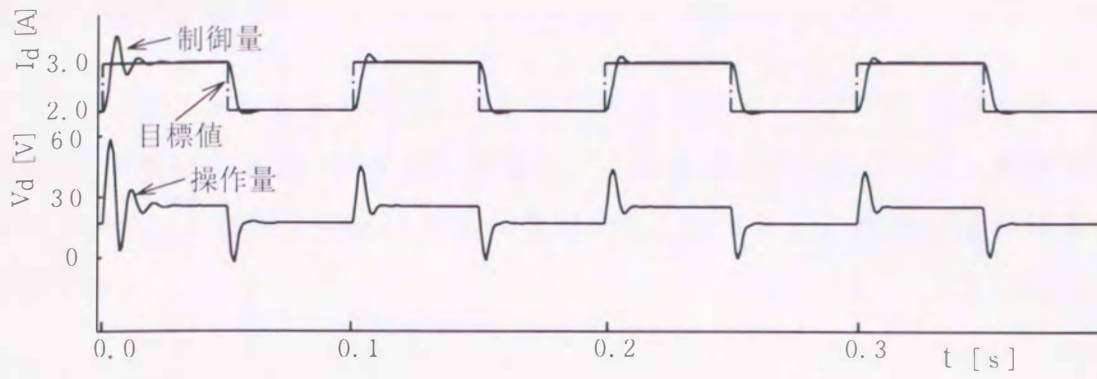


(a) 初期応答にオーバシュート有

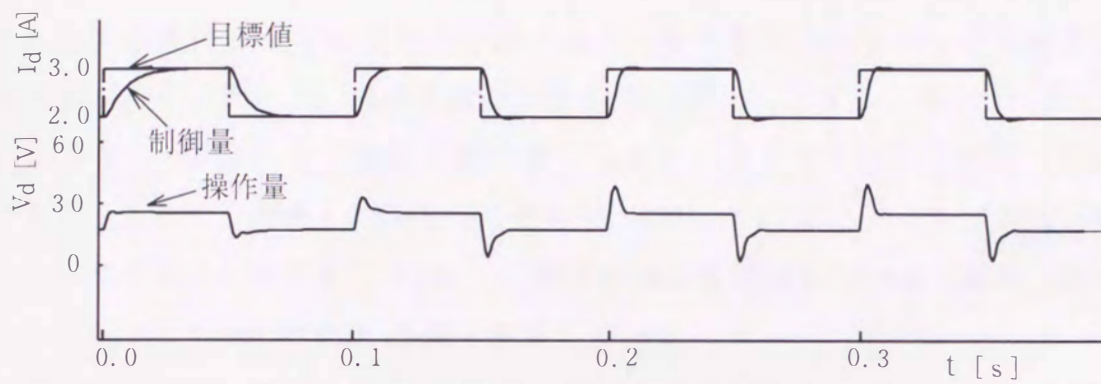


(b) 初期応答にオーバシュート無

図 2.14 特徴量 OV , DP , U_a を用いた場合のシミュレーション結果



(a) 初期応答にオーバシュート有



(b) 初期応答にオーバシュート無

図 2.15 特徴量 OV , DP , U_b を用いた場合のシミュレーション結果

2.3 実機実験

シミュレーションとその検討により良い応答を得ることができた操作量の特微量 U_b を用いて実機システムを構成し、その実験を行う。さらに、制御量にオーバershootを起こさないルールを用いて、制御応答の改善が行われるか確認する。

2.3.1 制御回路の構成

DCチョッパによる電流制御実験システムの構成を図2.16に示す。このシステムは検出部分として電流検出回路があり、操作量出力部分としてPWM発生回路がある。そして、内部変数などを出力してモニタするための12ビットD/Aコンバータと、一定時間間隔の割り込みを発生させる同期信号発生回路がある。さらに、ホストコンピュータのPC-9801のCPUをメインプロセッサとし、サブプロセッサとしてDSP (TMS320C30) を使用している。また、負荷として220V、100Wの電球15個を使用している。

本制御システムでは、1.024msごとに発生される割り込み信号に従ってすべ

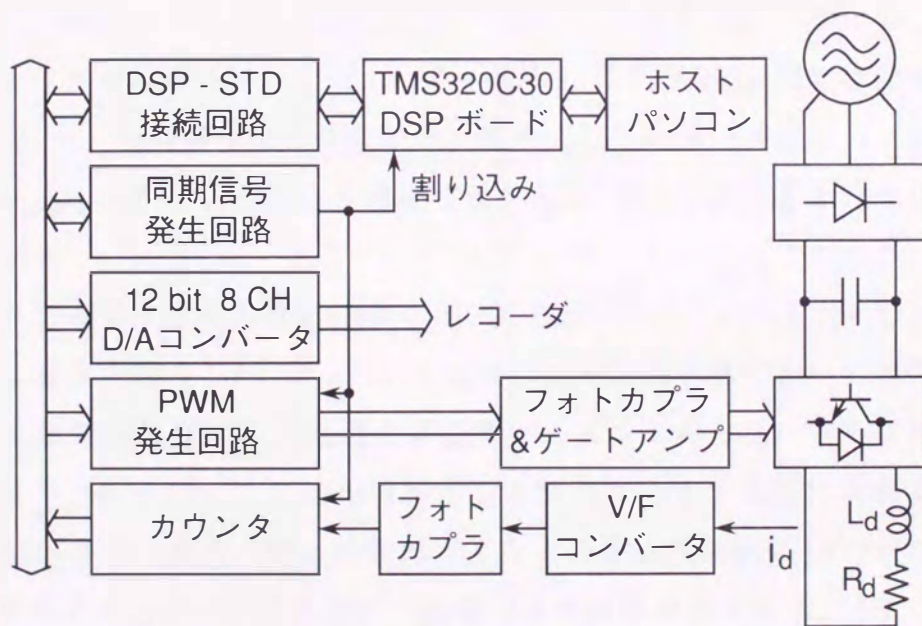


図 2.16 実験システム

での回路が動作する。まず、この信号に同期して、V/Fコンバータパルスのカウント値がラッチされる。この値がDSPに読み込まれて、前回との差により電流 I_d を演算し、指令値 I_d^* との制御演算を行うことによりデューティ比を求め、これをPWM発生回路へ送る。デューティ比は次回の割り込み信号により、PWMパターンとしてチョッパへ送られ、直流電流 I_d を制御している。

2.3.2 ソフトウェア構成

本システムのソフトウェアは、ホストコンピュータ(PC9801)側のプログラムとDSP側のプログラムに分けられる。

ホストコンピュータのプログラムは、C言語で書かれていて、DSPとの通信やIEEE形式の浮動小数点数とDSP形式との変換ルーチンなどから成る。このプログラムは、電源投入後のDSPボードの初期化やDSPのオブジェクトファイルをボード上の高速SRAMへロードし、オペレータからの指示や制御ゲインをDSPに伝送したりDSP上のデータを画面に表示する。

DSP側のプログラムは、アセンブラで書かれていて、初期化ルーチン、コマンド処理ルーチン、割り込みルーチンに分けられる。初期化ルーチンは、定数、変数、周辺デバイスの初期設定を行う。コマンド処理ルーチンは、ホストコンピュータからボードコンピュータへ送られてきた命令(例えば制御開始や終了など)や、定数変更の処理を行う。割り込み処理ルーチンは、一定時間間隔(1.024 ms)の割り込みにより直流電流の検出、I-PD制御等を行うルーチンである。

本システムでは浮動小数形DSPを用いて制御を行っているが、A/Dコンバータ等による入力部やD/AコンバータおよびPWM出力部においてはすべての状態量を整数形に変換して処理する必要がある。このとき、値を規格化することにより、オーバーフローや情報落ちを防止できる。実際に物理量を内部変数に対応させる場合、例えば電流10Aをそのまま内部値の10に対応させると、プロセッサのワード長をフルに活用できず精度が悪くなる。そこで、プロセッサのワード長で最上位ビットの下位に小数点を持つ固定小数点とすると、1ワードで表現できる値は-1.0~+1.0となる。これにより、物理量の最大値

を 1.0 に対応させて，ワード長を有効に利用し精度を保てる。

2.3.2.1 割り込み処理の概要

割り込み処理ルーチンはコントロールプログラムの要の部分であり，割り

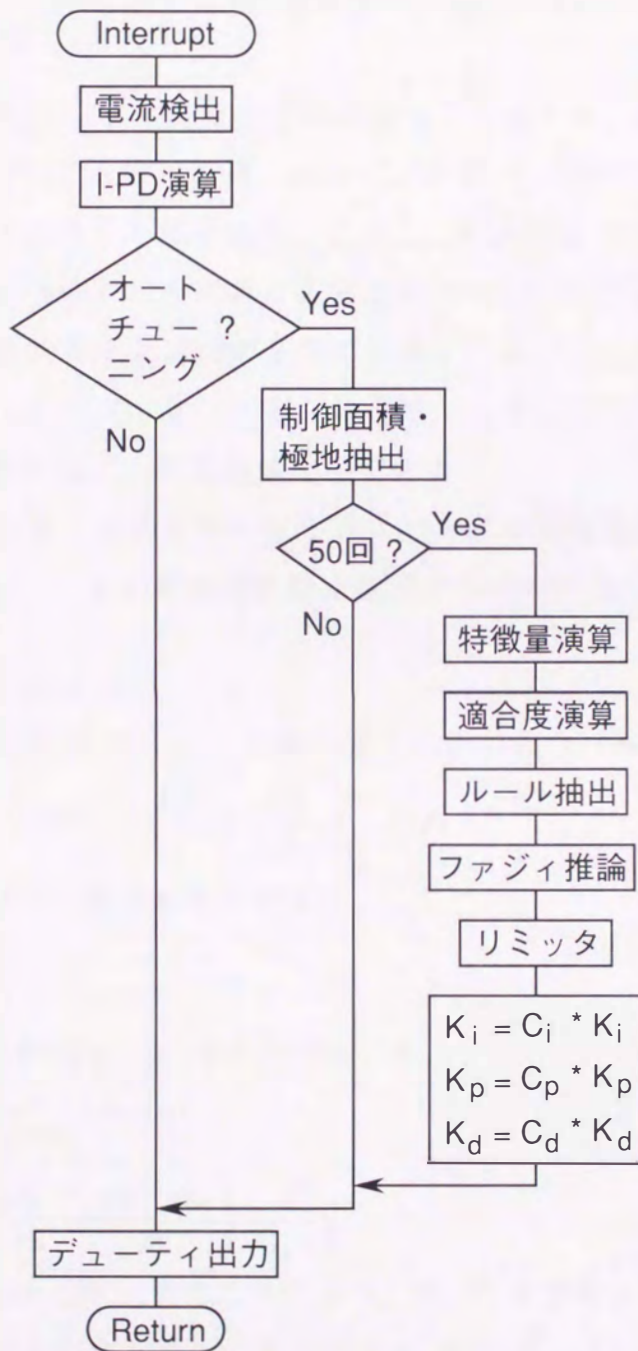


図 2.17 割り込み処理ルーチン

込み信号に同期して、電流検出を行い、オートチューニング演算や電流のI-PD制御を行う。このとき、I-PD制御の出力からデューティ比を求め、この値によりインバータ制御部トランジスタの点弧パターンおよび点弧時間が決まり、直流電流を制御することになる。このルーチンの流れ図を図2.17に示す。

2.3.2.2 電流検出

DCチョップパの作用により直流電流は脈動しているため、瞬時値を使用して制御したのでは希望する電流を得るのはむずかしい。そのため、チョップパ周期の平均値を使用することが望まれる。しかし、電流値をA/Dコンバータにより入力したのでは値の平均を求める必要があり、複数回のA/Dコンバータ変換時間と平均値を計算する時間が必要である。そこで、このシステムでは直流電流をV/Fコンバータによりパルスに変換し、カウンタにより電流の積分値を求め、積分値の差により電流値を得ている。

電流の最大値を I_m 、そのときのV/Fコンバータの発振周波数を F_m (1MHz) とすれば、電流 i のときの発振周波数は次式で与えられる。

$$f = \frac{F_m}{2} \left(1 + \frac{i}{I_m} \right) \quad (2.22)$$

サンプリング周期を T_c とすると k 回目までに出力したパルス数は、

$$p(k) = \int_0^{kT_c} f dt = \frac{F_m k T_c}{2} + \frac{F_m}{2 I_m} \int_0^{kT_c} i dt \quad (2.23)$$

となる。 k 回目までの電流の積分値を

$$I(k) = \int_0^{kT_c} i dt \quad (2.24)$$

とおくと、 k 回目の電流 $i(k)$ は次式で求まる。

$$\begin{aligned} \frac{i(k)}{I_m} &\simeq \frac{I(k) - I(k-n)}{n T_c I_m} \\ &= \left\{ \frac{p(k)}{2^{31}} - \frac{p(k-n)}{2^{31}} \right\} \frac{2^{32}}{F_m n T_c} - 1 \end{aligned} \quad (2.25)$$

ここで、変数を $p(k)/2^{31}$ とすることにより、32ビットで最上位ビットの下位に小数点を持つ固定小数点と考えることができ、 $2^{32}/(F_m n T_c)$ が整数になるように F_m , n , T_c を決定すると、演算誤差のない検出周期中の平均値を得ることができる。

2.3.2.3 ファジィオートチューニング

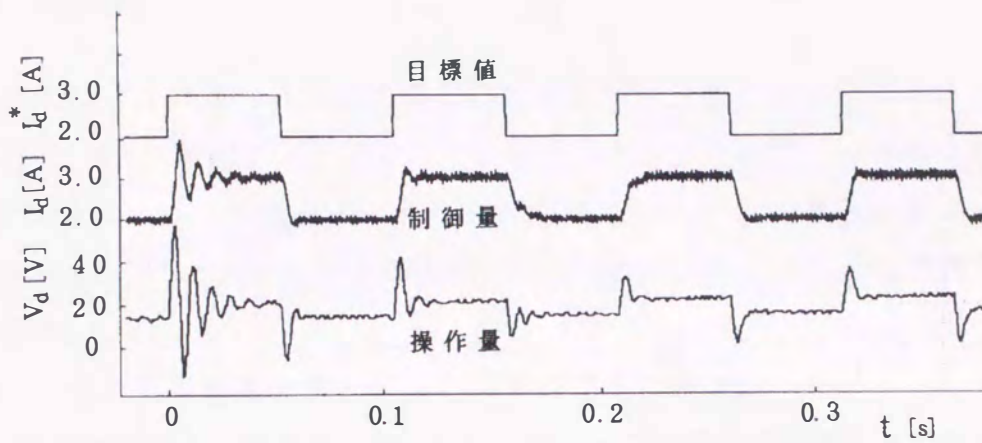
オートチューニングは、以下の手順で実行される。

- (1) 割り込み周期ごとに目標値と制御量の差の絶対値と操作量をメモリに格納
- (2) ステップ応答の指令が入ってから、決まった個数のデータを集めてオートチューニングルーチンを実行(実験では50回の割り込み)
- (3) それぞれの特徴量について適合度を演算
- (4) 当てはまるルールの抽出
- (5) ファジィ推論を行い、修正係数(C_i , C_p , C_d)を演算
- (6) その修正係数を0.5から2.0に制限
- (7) 前回の制御ゲインと修正係数を乗算することにより、新しいゲインを決定
この動作を指定した回数繰り返すことにより、適切な制御ゲインを決定する。

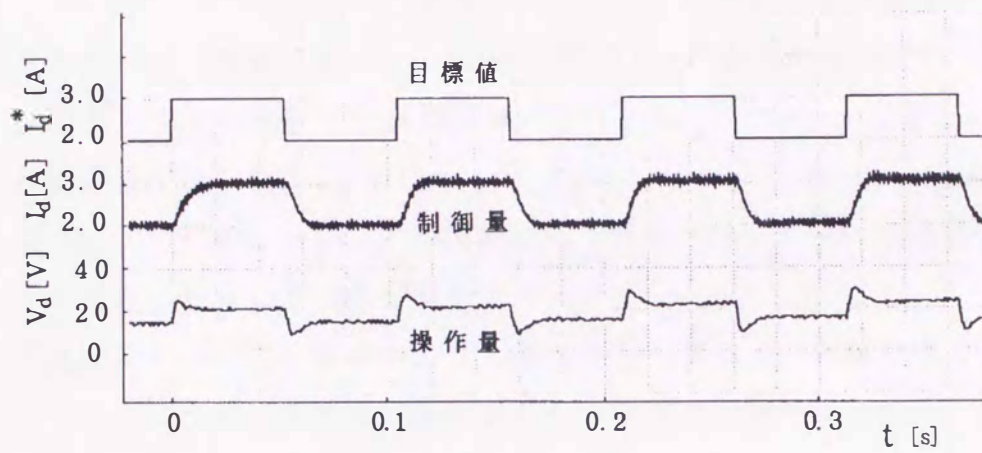
2.3.3 実験結果

オートチューニングの実験は電流指令2.0Aから3.0Aへのステップ応答を指定した回数繰り返して、それぞれのステップ応答波形の特徴量を求め表2.5のルールに従って行った。チューニング目標としてはオーバシュートが4%となるように設定した。

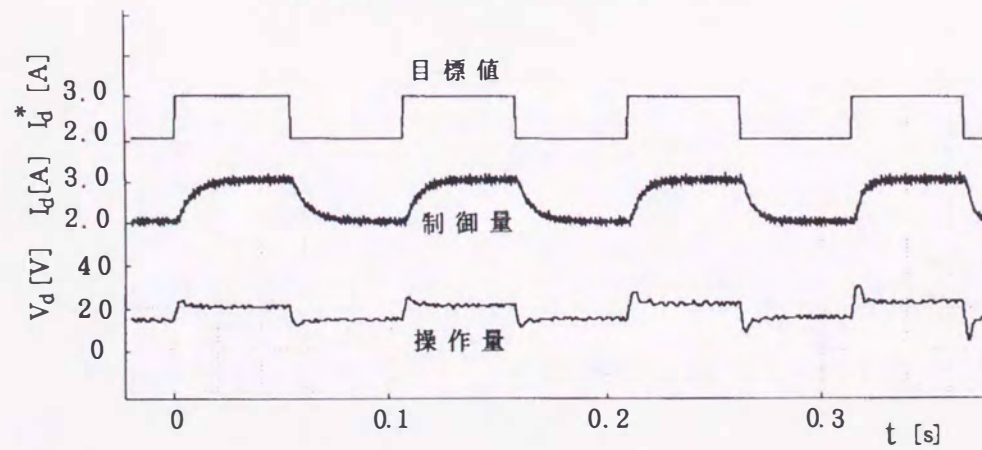
図2.18(a)~(c)は実験結果で、目標値 I_d^* 、制御量 I_d 、操作量 V_d を示している。このとき、 I_d は負荷に流れている電流波形であるが、 I_d^* と V_d は制御装置のD/Aコンバータより出力した波形である。(a)図は初期応答のオーバシュートが大きい場合、(b)図は初期応答にオーバシュートがない場合、(c)図はチューニング目標としてオーバシュートが0%となるようにした場合の結果である。いずれの場合にも数回の繰り返しで望ましい応答が得られている。ステップ応答波形でシミュレーションと実験結果が異なるのは、前者では瞬時に負荷抵抗が変化しているが、後者では徐々に変化するためである。この様子は、図中のステップ応答後における操作量の変化に現れている。チューニング目標としてのオーバシュートを0%にした場合、従来の方法ではチューニングは難しいが、操作量の特徴量を使用することにより改善が見られる。



(a) 初期応答にオーバシュート有



(b) 初期応答にオーバシュート無



(c) 最適値がオーバシュート無

図 2.18 実験結果

2.4 結論

本章では、目標値のステップ変化に対する操作量波形から得られる特徴量を用いた、I-PD 制御ゲインのファジイオートチューニングの一方法を提案した。このチューニング法をインバータ (DC チョッパ) による電流制御系に適用し、シミュレーションおよび実機実験によりその性能を検証し、期待通りの成績を得た。

本章で得られた結果を要約すると以下のようなになる。

- (1) 従来方式では困難であった、1 回のステップ応答での制御ゲインの修正が可能となった。
- (2) シミュレーション結果より、提案方式は従来の制御量と目標値だけで特徴量を算出する場合に比べて、I-PD 制御ゲインの収束が速やかになった。
- (3) DSP を用いた電流制御の実機実験により、数回のチューニングで適切な制御応答が得られた。
- (4) 従来方式ではチューニングが困難であったオーバシュート 0% のチューニング目標の場合にも改善が可能となった。
- (5) 電流検出において、電流のリプル分の影響を受けず、係数を適当な値にすることにより、高速で演算誤差のない検出が可能となった。

第 3 章 M 系列による同定と制御

本章では、M 系列を試験信号として用いたときの、M 系列の長さとは異なる個数のインパルス応答を正確に計算し、リアルタイム処理のための同定および制御定数演算式の単純化を提案し、これらの直流電流制御系への応用について述べる。さらに、オートチューニングを直流電流制御系へ適用するためのハードウェアとソフトウェアについて述べ、低速なプロセッサでも演算が可能なことを実証する。最後に、シミュレーション結果と実験結果の比較検討を行い、本方式がオートチューニング法として有用であることを確認する。

3.1 伝達関数を用いたオートチューニング

図 3.1 に示す I-PD 制御則を用いたオートチューニングは、I-PD 制御の比例、積分、微分の 3 個のゲインを自動的に決定する。このシステムでは、M 系列を用いた相互相関関数により制御対象の動特性を同定し、 z 領域から s 領域への変換を行い、その結果を用いて、制御対象に合った I-PD パラメータを決定する。

制御対象の動的モデルが必要なとき、そのシステムが線形システムで近似できるならば、数学モデルの表現法の 1 つとしてインパルス応答モデルがあ

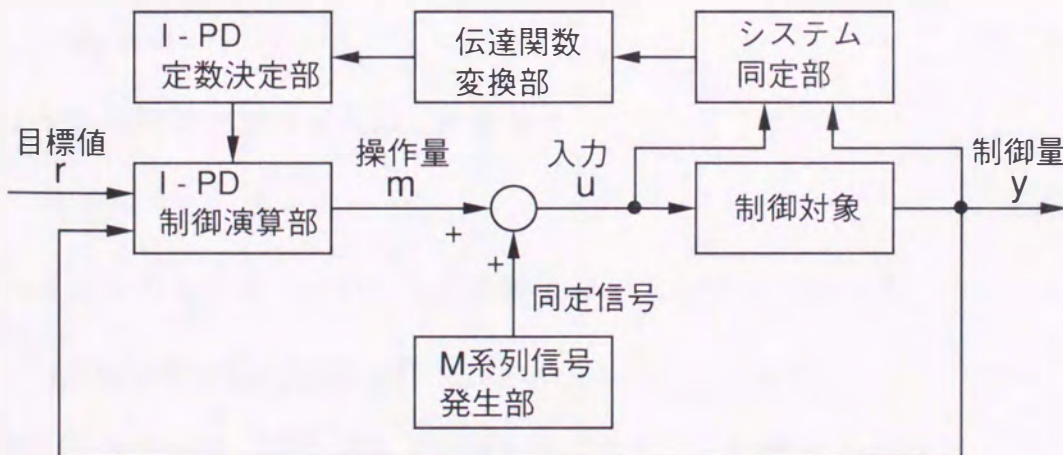


図 3.1 オートチューニングシステム

る。インパルス応答とは、単位インパルスに対する応答であるから、制御対象にインパルスを印加して、その応答を測定すればよいが、インパルスの振幅はシステムに許容される制限があり、あまり大きくできないので、出力測定値に雑音が入るといふ問題がある。この欠点を除くため一般に用いられているのが、特定の試験信号を加え相関関数を計算することにより、間接的にインパルス応答を求める方法である。この試験信号として、M系列信号を用いる。なお、本章ではM系列信号により同定を行っている間I-PD制御は行なわず、制御演算出力を一定にしている。

3.1.1 M系列信号

一般に、 n 個の遅れ素子をもち次式を満たすもののなかで、周期が最長のものがM系列(maximum period sequence)と呼ばれている⁽⁵⁰⁾。

$$(z^{-n} \oplus \cdots \oplus 1)x(k) = 0 \quad (3.1)$$

ここで、 z^{-1} 、 \oplus はそれぞれ遅れ素子、排他的論理和を意味し、 $x(k)$ は0または1の値をもち、 k は整数である。本章では、 $n=7$ の場合のM系列を用いる。この場合には、

$$(z^{-7} \oplus z^{-6} \oplus 1)x(k) = 0 \quad (3.2)$$

すなわち、

$$x(k) = x(k-6) \oplus x(k-7) \quad (3.3)$$

によって $x(k)$ が求まる。また、初期値は

$$x(k) = 1 \quad (k = 0, \dots, 6) \quad (3.4)$$

で与えるものとする。このとき、周期 L は $127(=2^7-1)$ となり、

11111110000001000001100001010 ...

の数列を出力する。 $x(k)$ は0または1であるから、振幅 K_m の

$$u(k) = K_m \{1 - 2x(k)\} \quad (3.5)$$

を M 系列信号として制御対象に加える。

M 系列信号の自己相関関数は次式により与えられる。

$$\psi_{uu}(i) = \frac{1}{M} \sum_{k=0}^{M-1} u(k) u(k+i) \quad (3.6)$$

ここで、観測する個数 M は L の整数倍

振幅 K_m が 1 のとき、(3.6) 式において i が 0 または L の整数倍であれば、

$$\psi_{uu}(i) = 1 \quad (3.7)$$

となる。上記以外の i については、次式となる。

$$\psi_{uu}(i) = -1/L \quad (3.8)$$

ここで、 i が 0 または L の整数倍以外するとき自己相関関数が負になるのは、(3) 式の系列 $\{x(k)\}$ において、1 がひとつ多く、 $u(k)$ で考えれば負の値がひとつ多いからである。振幅が 1 の場合の自己相関関数を図 3.2 に示す。

3.1.2 相互相関関数による同定

いま、M 系列信号 $\{u(k)\}$ を入力信号として、重み系列 $h(k)$ の線形システムに印加したときの出力系列 $\{y(k)\}$ は、

$$y(k) = \sum_{i=0}^k h(k-i) u(i) \quad (3.9)$$

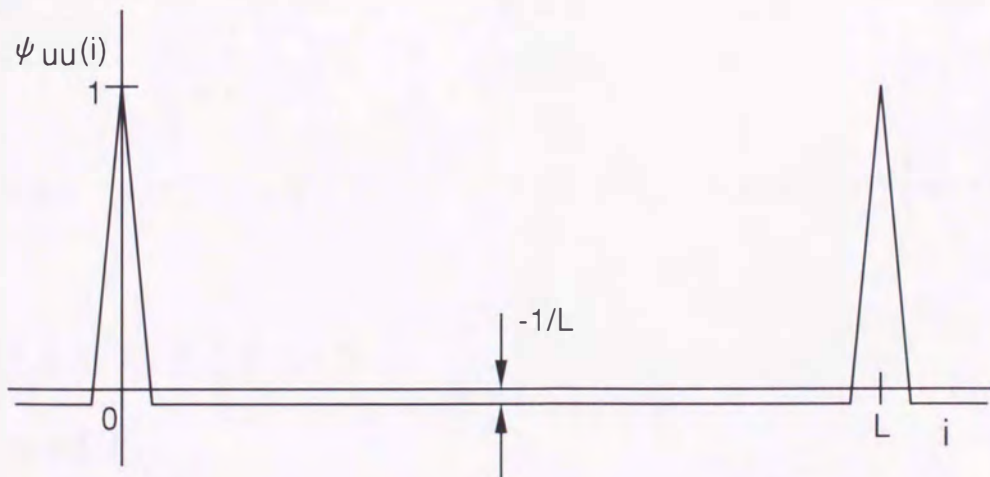


図 3.2 M 系列信号の自己相関関数

により与えられ、入力系列と出力系列の相互相関関数は、

$$\psi_{uy}(i) = \frac{1}{L} \sum_{k=0}^{L-1} u(k) y(k+i) \quad (3.10)$$

で与えられる。従来、重み系列は(3.10)式により得られていたが⁽⁴⁸⁾、M系列信号の自己相関関数がインパルス応答と異なることによる誤差がある。また、制御系で必要とするインパルス応答の重み系列の個数はM系列の長さとは異なる。

振幅が K_m のときの、M系列の長さとは異なる個数の重み系列を求めるために、 $L+N-1$ 個のM系列信号を制御対象に入力し、後半の L 個の出力データをサンプルすれば、

$$\begin{aligned} y(N-1) &= u(N-1)h(0) + u(N-2)h(1) + \cdots + u(0)h(N-1) \\ y(N) &= u(N)h(0) + u(N-1)h(1) + \cdots + u(1)h(N-1) \\ &\vdots \\ y(N+L-2) &= u(N+L-2)h(0) + u(N+L-3)h(1) + \cdots + u(L-1)h(N-1) \end{aligned} \quad (3.11)$$

となる。ここで、

$$h(k) \simeq 0 \quad (k \geq N) \quad (3.12)$$

と近似している。行列では次式で表現できる。

$$\begin{bmatrix} y(N-1) \\ y(N) \\ \vdots \\ y(N+L-2) \end{bmatrix} = \begin{bmatrix} u(N-1) & u(N-2) & \cdots & u(0) \\ u(N) & u(N-1) & \cdots & u(1) \\ \vdots & \vdots & \ddots & \vdots \\ u(N+L-2) & u(N+L-3) & \cdots & u(L-1) \end{bmatrix} \begin{bmatrix} h(0) \\ h(1) \\ \vdots \\ h(N-1) \end{bmatrix} \quad (3.13)$$

上式を次式の形式で表現する。

$$\mathbf{y} = \mathbf{U} \mathbf{h} \quad (3.14)$$

この式より、重み系列は次式により得ることができる⁽²²⁾。

$$\mathbf{h} = (\mathbf{U}^T \mathbf{U})^{-1} \mathbf{U}^T \mathbf{y} \quad (3.15)$$

$\mathbf{U}^T \mathbf{y}$ の i 行の要素は

$$[\mathbf{U}^T \mathbf{y}]_i = \sum_{k=0}^{L-1} u(N-i+k) y(N-1+k) \quad (3.16)$$

であるから、M 系列信号の相互相関関数の式より、

$$[\mathbf{U}^T \mathbf{y}]_i = L \psi_{uy}(i-1) \quad (3.17)$$

$$\mathbf{U}^T \mathbf{y} = L \begin{bmatrix} \psi_{uy}(0) \\ \psi_{uy}(1) \\ \vdots \\ \psi_{uy}(N-1) \end{bmatrix} \quad (3.18)$$

を得る。また、(3.15) 式の行列 $\mathbf{U}^T \mathbf{U}$ の i 行 j 列の要素は

$$[\mathbf{U}^T \mathbf{U}]_{ij} = \sum_{k=0}^{L-1} u(N-i+k) u(N-j+k) \quad (3.19)$$

であるから、M 系列信号の自己相関関数の性質より

$$[\mathbf{U}^T \mathbf{U}]_{ij} = \begin{cases} L K_m^2 & (i=j) \\ -K_m^2 & (i \neq j) \end{cases} \quad (3.20)$$

$$\mathbf{U}^T \mathbf{U} = K_m^2 \begin{bmatrix} L & -1 & \cdots & -1 \\ -1 & L & \cdots & -1 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & \cdots & L \end{bmatrix} \quad (3.21)$$

を得る。この逆行列は

$$(\mathbf{U}^T \mathbf{U})^{-1} = \frac{1}{(L+1)K_m^2} \begin{bmatrix} 1 + \frac{1}{L+1-N} & \frac{1}{L+1-N} & \cdots & \frac{1}{L+1-N} \\ \frac{1}{L+1-N} & 1 + \frac{1}{L+1-N} & \cdots & \frac{1}{L+1-N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{L+1-N} & \frac{1}{L+1-N} & \cdots & 1 + \frac{1}{L+1-N} \end{bmatrix} \quad (3.22)$$

であるから、

$$h = \frac{L}{(L+1)K_m^2} \begin{bmatrix} 1 + \frac{1}{L+1-N} & \frac{1}{L+1-N} & \cdots & \frac{1}{L+1-N} \\ \frac{1}{L+1-N} & 1 + \frac{1}{L+1-N} & \cdots & \frac{1}{L+1-N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{L+1-N} & \frac{1}{L+1-N} & \cdots & 1 + \frac{1}{L+1-N} \end{bmatrix} \begin{bmatrix} \psi_{uy}(0) \\ \psi_{uy}(1) \\ \vdots \\ \psi_{uy}(N-1) \end{bmatrix} \quad (3.23)$$

$$h(k) = \frac{L}{(L+1)K_m^2} \left\{ \psi_{uy}(k) + \frac{1}{L+1-N} \sum_{i=0}^{N-1} \psi_{uy}(i) \right\} \quad (3.24)$$

を得る。

3.1.3 伝達関数変換

文献(25)の方法を用いて、パルス伝達関数から s 領域の伝達関数への変換を行う。この方法は、制御対象のパルス伝達関数から求めたステップ応答が制御対象のステップ応答にサンプル時刻で一致することから、パルス伝達関数のステップ応答を時間 t の多項式でフィッティングし、その多項式関数をラプラス変換することにより伝達関数を導出する方法である。

文献(25)では、制御対象の伝達関数 $G(s)$ を

$$G(s) = \frac{1}{g_0 + g_1 s + g_2 s^2 + \cdots} \quad (3.25)$$

の分母系列表現とし、パルス伝達関数が

$$G(z) = \frac{z^{-k} \sum_{i=0}^n b_i z^{-i}}{1 + \sum_{i=1}^m a_i z^{-i}} \quad (3.26)$$

ここで、 k :無駄時間、 m :分母の次数、 n :分子の次数
の場合の s 領域伝達関数への変換を次式により行っている。

$$A_0 = 1 + \sum_{j=1}^m a_j \quad (3.27)$$

$$A_i = \frac{(-T_c)^i}{i!} \sum_{j=1}^m j^i a_j \quad (3.28)$$

$$B_0 = \sum_{j=0}^n b_j \quad (3.29)$$

$$B_i = \frac{(-T_c)^i}{i!} \sum_{j=0}^n (k+j)^i b_j \quad (3.30)$$

$$H_0 = \frac{A_0}{B_0} \quad (3.31)$$

$$H_i = \frac{A_i - \sum_{j=0}^{i-1} H_j B_{i-j}}{B_0} \quad (3.32)$$

$$g_0 = H_0 \quad (3.33)$$

$$g_i = H_i - \sum_{j=0}^{i-1} \xi_{i-j} g_j T_c^{i-j} \quad (3.34)$$

ただし、 T_c はサンプリング周期であり、 ξ_i は三次多項式を用いて変換精度を向上させた補正項で、次式の値を持つ⁽²⁵⁾。

$$\{\xi_0, \xi_1, \xi_2, \xi_3, \xi_4, \dots\} = 1, \frac{1}{2}, \frac{1}{12}, 0, \frac{1}{40}, \dots \quad (3.35)$$

文献(25)では、(3.28)、(3.30)、(3.34)式にサンプリング周期 T_c を含んでいる。そこで、制御対象の伝達関数 $G(s)$ を

$$G(s) = \frac{1}{g_0 + g_1 T_c s + g_2 T_c^2 s^2 + \dots} \quad (3.36)$$

の分母系列表現とすることにより、 s 領域伝達関数への変換式は次式に書き直せる。

$$A_0 = 1 + \sum_{j=1}^m a_j \quad (3.37)$$

$$A_i = \frac{(-1)^i}{i!} \sum_{j=1}^m j^i a_j \quad (3.38)$$

$$B_0 = \sum_{j=0}^n b_j \quad (3.39)$$

$$B_i = \frac{(-1)^i}{i!} \sum_{j=0}^n (k+j)^i b_j \quad (3.40)$$

$$H_0 = \frac{A_0}{B_0} \quad (3.41)$$

$$H_i = \frac{A_i - \sum_{j=0}^{i-1} H_j B_{i-j}}{B_0} \quad (3.42)$$

$$g_0 = H_0 \quad (3.43)$$

$$g_i = H_i - \sum_{j=0}^{i-1} \xi_{i-j} g_j \quad (3.44)$$

これにより、制御対象の伝達関数は複雑になるが、パルス伝達関数から s 領域の伝達関数への変換式は簡単になる。同定を行うときに演算しなければならないのは変換式であり、制御対象の伝達関数は演算する必要はない。このため、文献(25)の変換式に比べて、本方式では高速演算が可能となる。

3.1.4 制御ゲイン決定

制御対象の動特性に関する部分的な知識により、サンプル値制御系の制御ゲインを設定する方法に、2.2.1節で述べた部分的モデルマッチング法⁽²⁸⁾がある。この方法は、制御対象の伝達関数に制御装置を付加した目標値から制御量までの伝達関数を求め、これがあらかじめ定めておいた参照モデルの伝達関数に低次から一致するように、I-PD 制御ゲインを調整して決定するものである。この参照モデルとしては、北森モデル⁽²⁷⁾を使用する。

2.2.1節で述べた部分的モデルマッチング法では、I-PD 制御ゲインを計算する(2.18), (2.19), (2.20), (2.21)式にサンプリング周期 T_c を含んでいる。

そこで、制御対象の伝達関数の分母系列表現 $G(s)$ を(3.36)式、参照モデルを

$$G_m(s) = \frac{1}{\alpha_0 + \alpha_1 \sigma T_c s + \alpha_2 \sigma^2 T_c^2 s^2 + \dots} \quad (3.45)$$

の分母系列表現とすることにより、2.2.1節のゲイン演算式は次式に書き直せる。変数 σ の方程式

$$\begin{aligned} & (g_2 + g_1 + g_0)\alpha_4 \sigma^3 \\ & + (-g_3 + \frac{7}{12}g_1 + \frac{1}{4}g_0)\alpha_3 \sigma^2 \\ & + (-g_3 - \frac{7}{12}g_2 + \frac{1}{18}g_0)\alpha_2 \sigma \\ & + (-\frac{1}{3}g_3 - \frac{1}{4}g_2 - \frac{1}{18}g_1)\alpha_1 = 0 \end{aligned} \quad (3.46)$$

の根の中で、正で最小の根を

$$K_i = \frac{3g_2 + 3g_1 + g_0}{3\alpha_3\sigma^3 + 3\alpha_2\sigma^2 + \alpha_1\sigma} \quad (3.47)$$

$$K_p = K_i\alpha_1\sigma - g_0 \quad (3.48)$$

$$K_d = K_i\alpha_2\sigma^2 - g_1 + K_p/2 \quad (3.49)$$

に代入することにより、積分ゲイン K_i 、比例ゲイン K_p 、微分ゲイン K_d を求めることができる。この時間に関する正規化により、 σ も制御周期 T_c に対する比率となる。

3.1.5 制御対象

電球とリアクトルを負荷とする電流制御系の s 領域での伝達関数は次式で与えられる。

$$I_d(s) = \frac{1}{R_d + L_d s} V_d(s) \quad (3.50)$$

デジタル制御においては、操作量の出力に 1 制御周期 (T_c) の遅れが存在する。従って、上式にむだ時間を考慮した伝達関数は

$$\begin{aligned} \frac{I_d(s)}{V_d(s)} &= \frac{\exp(-T_c s)}{R_d + L_d s} \\ &= \frac{1}{R_d + (R_d T_c + L_d)s + (R_d T_c^2/2 + L_d T_c)s^2 + \dots} \end{aligned} \quad (3.51)$$

で与えられる。ここで、(3.51) 式の導出には $\exp(T_c s)$ のマクローリン展開を使用した。

本章の実験システムでは整数形 CPU を使用しているため、状態量の値そのものを使用することは難しい。そこで、電流、電圧の最大値 (I_m, V_c) と制御周期で正規化を行うことにより、次式が得られる。

$$\frac{I_d(s)}{I_m} = \frac{1}{\frac{I_m R_d}{V_c} + \frac{I_m L_d}{V_c T_c} s} \frac{V_d(s)}{V_c} \quad (3.52)$$

$$G(s) = \frac{1}{\frac{I_m R_d}{V_c} + (\frac{I_m R_d}{V_c} + \frac{I_m L_d}{V_c T_c})T_c s + (\frac{I_m R_d}{2V_c} + \frac{I_m L_d}{V_c T_c})T_c^2 s^2 + \dots} \quad (3.53)$$

(3.36) 式と (3.53) 式を比較することにより次式を得る。

$$g_0 = \frac{I_m R_d}{V_c} \quad (3.54)$$

$$g_1 = \frac{I_m R_d}{V_c} + \frac{I_m L_d}{V_c T_c} \quad (3.55)$$

$$g_2 = \frac{I_m R_d}{2V_c} + \frac{I_m L_d}{V_c T_c} \quad (3.56)$$

$$g_3 = \frac{I_m R_d}{6V_c} + \frac{I_m L_d}{2V_c T_c} \quad (3.57)$$

電流の検出に V/F コンバータとカウンタを使用すると、2.1.3節のように制御対象のパルス伝達関数の分子は z^{-1} の三次式となる。しかし、PWM 周期と電流検出周期を制御周期の 1/4 にしているため、積分と差による平均処理に伴う遅れを無視することが可能となり、 T_c を制御周期とした RL 負荷のパルス伝達関数は次式に近似できる。

$$G(z) = z^{-2} \frac{b_0}{1 + a_1 z^{-1}} \quad (3.58)$$

ここで、

$$a_1 = -\exp\left(-\frac{R_d T_c}{L_d}\right)$$

$$b_0 = \frac{V_c}{R_d I_m} (1 + a_1)$$

上式は s 領域の伝達関数と同様に、電流、電圧の最大値 (I_m , V_c) で正規化している。

重み系列を用いてパルス伝達関数を表現すれば a_1 , b_0 は次式となる。

$$\left. \begin{aligned} a_1 &= -\frac{h(3)}{h(2)} \\ b_0 &= h(2) \end{aligned} \right\} \quad (3.59)$$

ただし、 $h(2)$, $h(3)$ は重み系列の 2 個目と 3 個目の値であり、(3.24) 式より得られる。

これにより、M 系列を用いた相互相関関数によりパルス伝達関数を求めることができる。

3.2 実機実験

オートチューニングを直流電流制御系で実現するための制御プログラムをアセンブリ言語で記述し、高速演算やメモリ消費低減のためにいくつかの工夫を行い、低速なプロセッサでも演算が可能なことを実証している。この実験装置による結果とシミュレーション結果の比較検討を行い、本方式がオートチューニング法として有用であることを確認している。

3.2.1 制御回路の構成

DCチョッパによる直流電流制御システムの構成を図3.3に示す。このシステムでは検出部分として電流検出回路、操作量出力部分としてPWMパターン発生回路および制御演算を行うマイコン制御回路から構成している。

CPUには16ビットのV20を用いており、RS-232Cによりパソコンに接続されていて、パソコンから制御側CPUへ指令を出したり、逆にシステムの状態をパソコンへ表示することも可能である。このためのソフトウェアは独自に開発した⁽⁵¹⁾。直流電流をV/Fコンバータによりパルスに変換し、カウンタにより電流の積分値を得る。CPUは1.024msごとにカウンタの値を取り込んで電

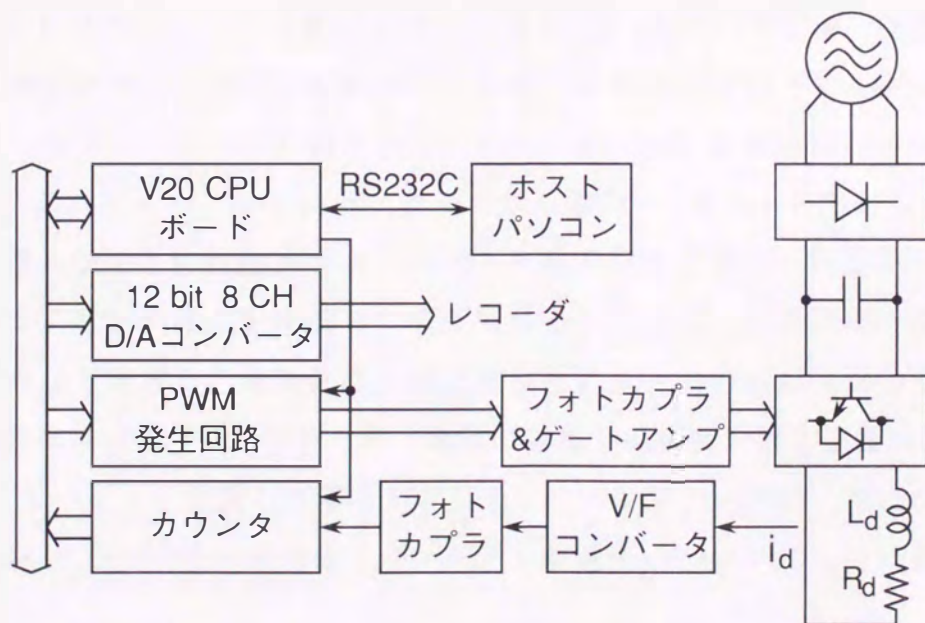


図 3.3 システム構成

流を求め、パソコンから送られてきた電流目標値との間で I-PD 演算を行う。I-PD 演算により求まる端子電圧をデューティに変換して出力し、トランジスタの点弧パターンを決定する。PWM パターン発生回路は 1 周期を $256 \mu\text{s}$ とし、デューティに対応したオンパルスを出し、トランジスタに出力する。これにより、1 制御周期につき 4 回スイッチングしている。

3.2.2 ソフトウェア構成

制御プログラムはすべて制御用の CPU 上で動作し、ホストパソコンはプログラム開発と端末動作のみを行う。制御プログラムのメイン・ルーチンは周辺回路の初期化、オペレータからの目標値変更等の指令処理や状態表示を行い、割り込みルーチンは電流検出、I-PD 制御、M 系列信号による重み系列の演算および制御ゲインの演算を行う。このとき、整数形 16 ビット CPU を用いて制御を行っているため、システムの状態変数は 16 ビット固定小数点で表現している。I-PD 演算式も正規化した変数を使用しているため、制御ゲインも正規化した値を使用している。

3.2.2.1 内部変数

浮動小数点を用いれば実際の数値を CPU 内部で処理できるが、浮動小数点演算は数値演算プロセッサを使用してもかなりの時間を要する(表 3.2 参照)。そこで、16 ビットデータを最上位ビットのつぎに小数点を持つ固定小数点と考える。これにより、CPU 内部で表現できる値は -1 から 1 までとなり、状態変数の最大値を 1 に対応させて、16 ビット長を有効に利用し精度を保つことができる。固定小数点を使用して演算処理を行う場合、加減算は小数点の位置を合わせて演算し、乗除算は小数点の位置に気をつけなければならない。なお、入力や演算結果は 2 バイトであるが、演算の過程では精度を落とさないように 4 バイト、6 バイトと拡張して演算している。このシステムでは、電流の最大値 I_m を 10A とし、電圧の最大値 V_c (コンデンサ電圧) を 100V にしている。

3.2.2.2 電流検出

電流検出は2.3.2.2節と基本的に同じであるが、本章で使用したCPUの1ワードは16ビットであるため、次式により電流値を計算する。

$$\frac{i(k)}{I_m} = \left\{ \frac{p(k)}{2^{15}} - \frac{p(k-n)}{2^{15}} \right\} \frac{2^{16}}{F_m n T_c} - 1 \quad (3.60)$$

ここで、 $2^{16}/(F_m n T_c)$ が整数になるように各値を決定する。また、 $p(k)/2^{15}$ としているのは、16ビットで最上位ビットの次に小数点を持つ固定小数点と考えているからである。これにより、 $n=1$ とした場合、 $i(k)/I_m$ は上位ビットから10ビットの精度で値を得ることができる。また、係数が整数であるため演算誤差も生じない。

3.2.2.3 I-PDゲインの決定

I-PD制御は(2.3)式

$$\begin{aligned} m(k) = & m(k-1) + K_i \{r(k) - y(k)\} \\ & - K_p \{y(k) - y(k-1)\} - K_d \{y(k) - 2y(k-1) + y(k-2)\} \end{aligned} \quad (3.61)$$

ここで、 r :目標値、 y :制御量、 m :制御演算出力で演算するが、本章のシステムでは整数形16ビットCPUを用いて制御を行っているため、状態変数は

$$r(k) = \frac{i_d^*(k)}{I_m},$$

$$y(k) = \frac{i_d(k)}{I_m},$$

$$m(k) = \frac{v_d(k)}{V_c},$$

ここで、 i_d^* :指令電流により正規化している。また、ゲインも I_m/V_c を乗ずることにより内部値とし、以下の手順で求める。

重み系列をCPUで演算するさいに、3.1.2節の(3.10)式をそのまま演算したのでは(M系列の長さ L)+(重み系列の個数 N)-1個の入力と後半の L 個の出

力データおよび(3.16)式より $L * N$ 回のかけ算が必要であるため、多くのメモリと演算時間が必要である。実際の入力 u は、M 系列による同定に入る前の操作量に、操作量最大値の 12.5% の振幅 ($K_m = 12.5V$) を重畳した信号を用いている。同定開始前の操作量、制御量をそれぞれ m_0, c_0 としたときの同定の流れ図を図 3.4 に示す。ここで、32 個の重み系列を求めるには、M 系列はビット単位であるから 4 バイト (32 ビット) 分の過去のデータを保存しておけばよい。同定時の割り込み演算は 158 回行なわれ、最初の 31 回は M 系列信号のみ発生して相互相関関数の演算は行なわず、後半の 127 回は M 系列信号の発生と相互相関関数の演算を行なう。この後、 L で割ることにより相互相関関数を求めている。更に、(3.24) 式の補正を行い重み系列を演算し、(3.58) 式によりパルス伝達関数を得る。

(3.37)~(3.44) 式を使用して、同定したパルス伝達関数を s 領域の伝達関数に変換する。最後に、(3.46)~(3.49) 式を使用して、I-PD ゲインを決定する。このとき、(3.46) 式の σ は 1 より大きい (制御周期より大きい) 最小の根であるから、根を求めるために、 σ を 1 より徐々に大きくして式の左辺の符号が変化したところからニュートン法を用いている。

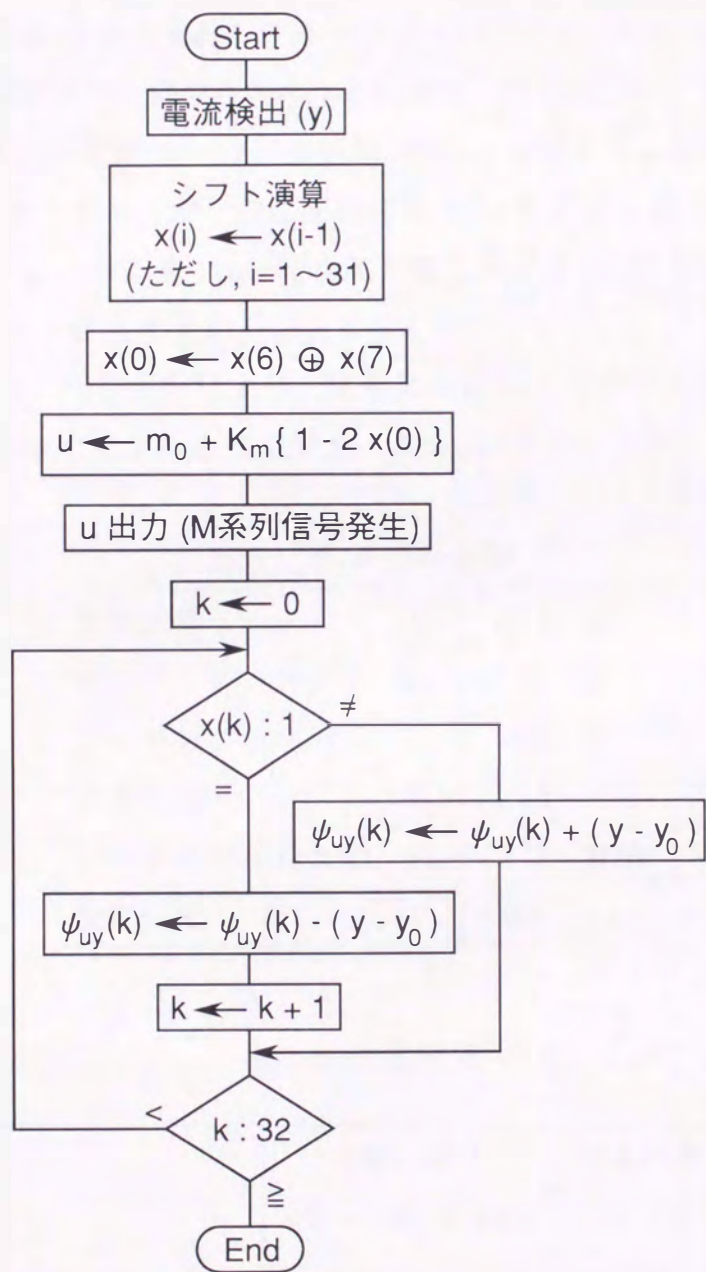


図 3.4 同定の流れ図

3.2.3 実験結果

表3.1に本システムに用いた負荷等の諸元を示す。本章のシステムではV20CPUのアセンブラプログラムにより、M系列信号の発生、相互相関関数による同定、伝達関数変換および北森モデルによるI-PDゲインの決定を行っている。このとき、同定演算1回につき約 $500\mu\text{s}$ を要し、同定のために158(M系列の長さ127+重み系列の個数32-1)制御周期(162ms)を要する。各部の演算時間は表3.2に示すとおりであるが、伝達関数変換とI-PDゲイン決定の間もI-PD制御を行なっているために約15msの時間を要した。(3.24)式も含めた、同定開始からI-PDゲイン決定まで約183msを要した。

図3.5はシミュレーションにより、M系列を使用して求めた相互相関関数とその補正により求めた重み系列を示す。このシミュレーションは、振幅1の理

表 3.1 システム定数

負荷抵抗	R_d	12.8 Ω
負荷インダクタンス	L_d	0.06 H
コンデンサ電圧	V_c	100 V
最大電流	I_m	10 A
V/F 最大発振周波数	F_m	1 MHz
制御周期	T_c	1.024 ms

表 3.2 演算時間

	浮動小数点	固定小数点
	V20+8087 (8MHz)	V20 (8MHz)
	(ms)	(ms)
重み系列計算	6.1	3.1
伝達関数係数計算	4.8	2.4
制御ゲイン計算	11.8	5.3

想的な M 系列信号が負荷に加えられているとして、C 言語により行った。実線が解析的に求めたインパルス応答の理論値で、○印が (3.10) 式の相互相関関数から求めたものである。また、△印は (3.24) 式により補正を行った値である。図より M 系列の自己相関関数がインパルスと異なることによるずれがあるが、補正を行うことにより解析的に求めた重み系列とよく一致しているのがわかる。この補正を使用して実際のシステムにより求めた重み系列を図 3.6 に示す。ここで、 k が 0 と 1 において重み系列が 0 であるのは、CPU の演算時間のために操作量の出力が 1 制御周期だけ遅れるからである。図 3.7 にシステム同定の際に使用した M 系列信号とそのときの電流波形を示す。この図における M 系列信号は、マイコン内部の制御対象入力を D/A コンバータにより出力したものである。

表 3.3 に制御対象の s 領域での伝達関数の係数を示す。理論値は (3.54) 式から (3.57) 式を用いて R_d , L_d の測定値から求め、同定法と I-PD 制御時のステップ応答のシミュレーションは、チョッパのスイッチングを考慮して数値計算により行った。実験値は V20CPU のシステムにより RL 負荷に M 系列信号を加えてパルス伝達関数を求め、(3.37) 式から (3.44) 式までを用いて求めた伝達関数である。シミュレーション値と理論値の間のわずかな誤差は、重み系列を求めるとき相互相関関数の個数を 32 で打ち切ったために生じたものである。表 3.4 は実験により得られた I-PD ゲインであり、比較のために σ が 4 と 5 の場合も示す。

図 3.8 に実験により得られた I-PD ゲインを用いて、電流目標値を 3A から 5A に変化したときのステップ応答を示す。図 3.8 (b) のモデルの応答は、(3.45) 式の参照モデルにおいて σ が 3.18 であり α_0 から α_4 までを使用して求めたものである。ここでも、実際の応答とシミュレーション結果がよく一致している。同様に、図 3.9, 3.10 に σ がそれぞれ 4, 5 の場合のステップ応答を示す。この図から σ に対応して応答が遅くなるのがわかる。

表 3.3 伝達関数の同定値

	理論値	シミュレーション値	実験値
g_0	1.28	1.28	1.28
g_1	7.14	7.16	7.11
g_2	6.50	6.74	6.69
g_3	3.14	3.86	3.83

表 3.4 I-PD 制御ゲイン

σ	3.18	4.00	5.00
K_i	1.30	0.75	0.43
K_p	2.85	1.73	0.88
K_d	0.89	-0.24	-1.27

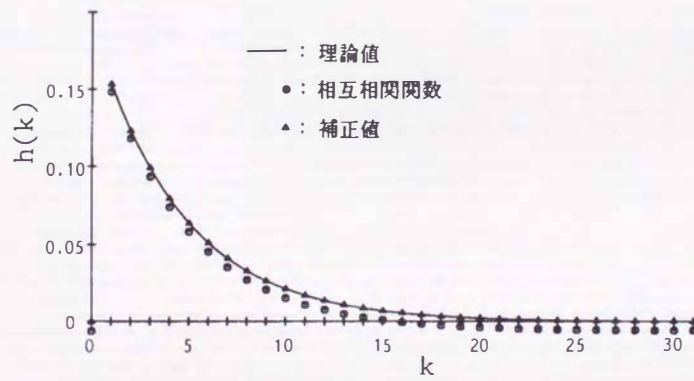


図 3.5 シミュレーションによる相互相関関数

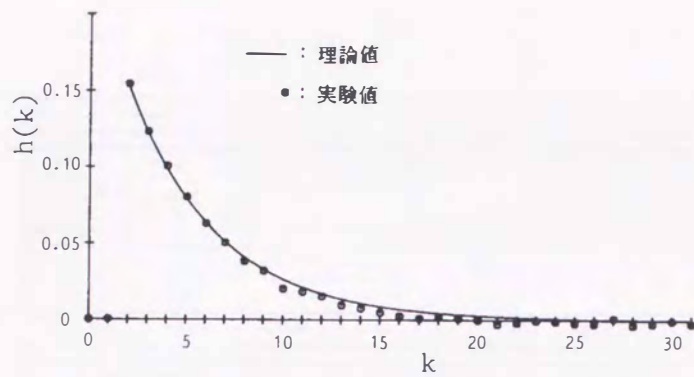


図 3.6 重み系列の実験値

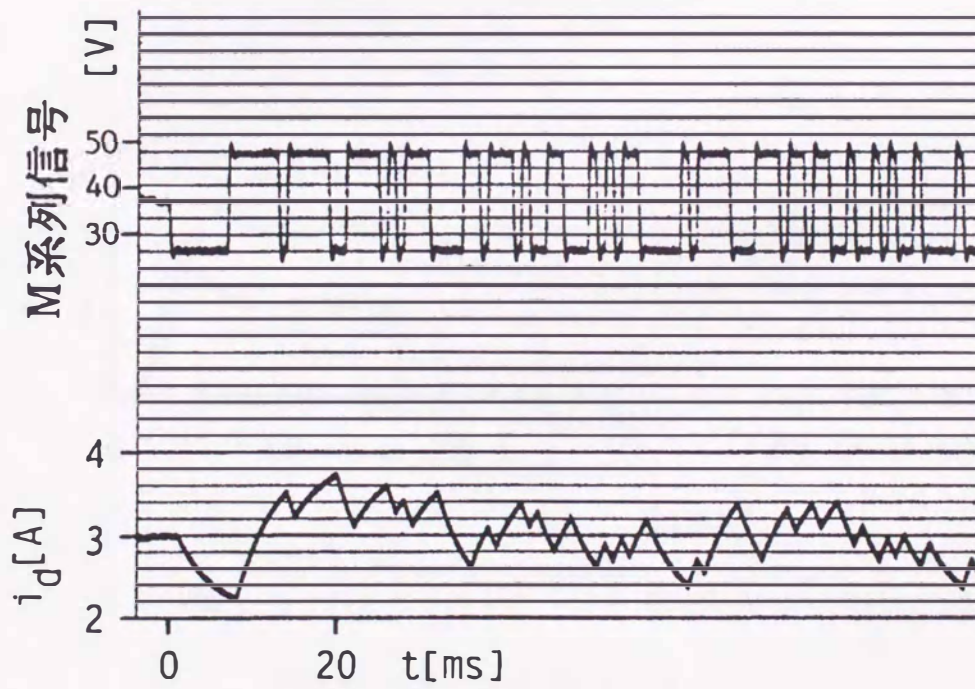
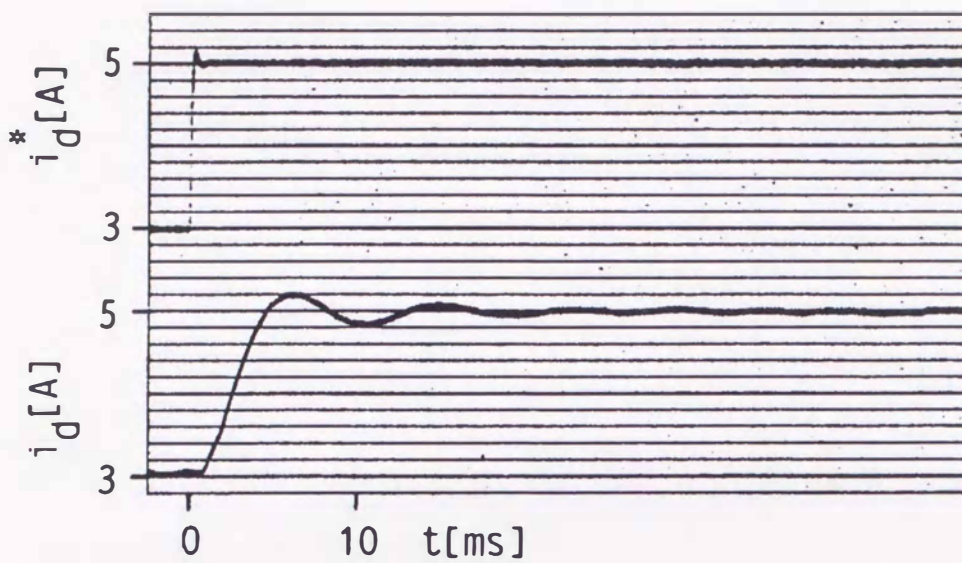
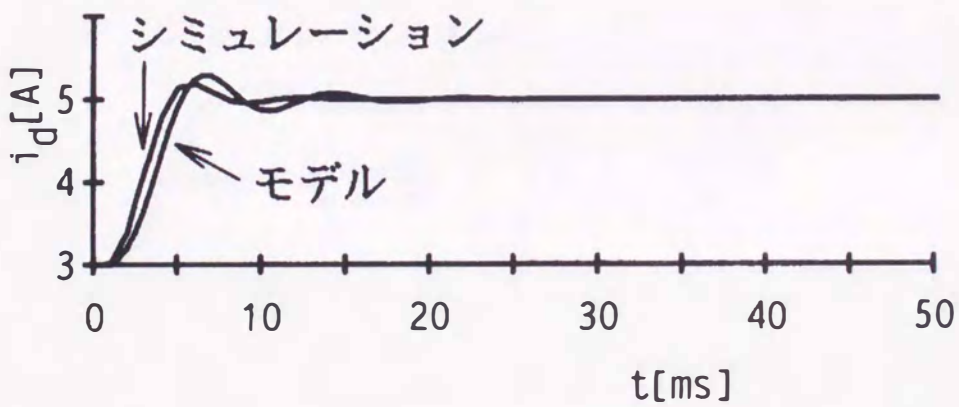


図 3.7 M 系列信号



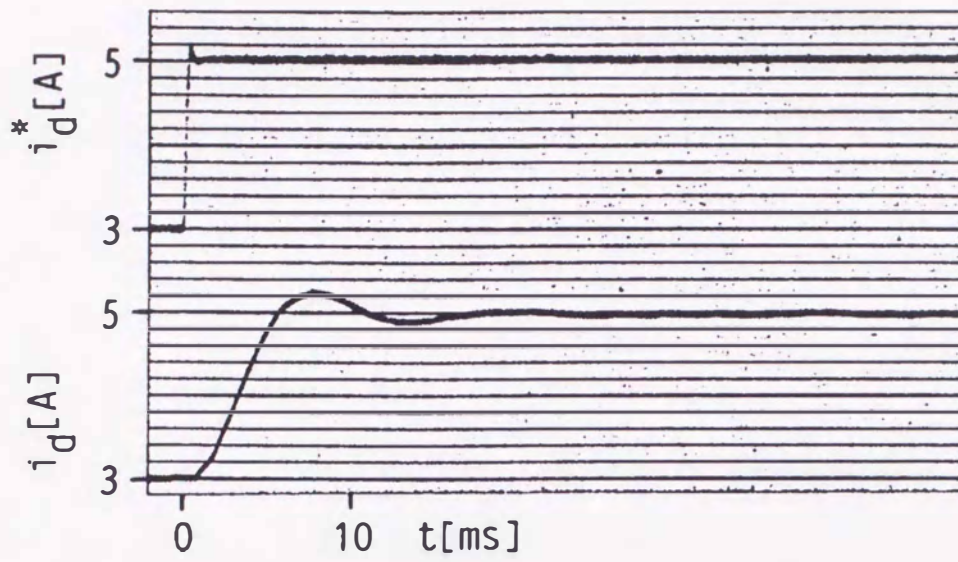
$K_i=1.30, K_p=2.85, K_d=0.89, \sigma=3.18$

(a) 実験値



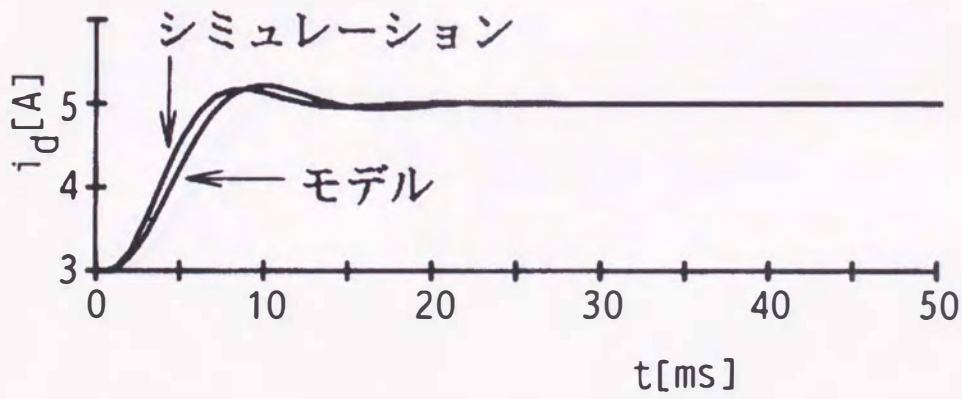
(b) シミュレーション値

図 3.8 ステップ応答 ($\sigma = 3.18$)



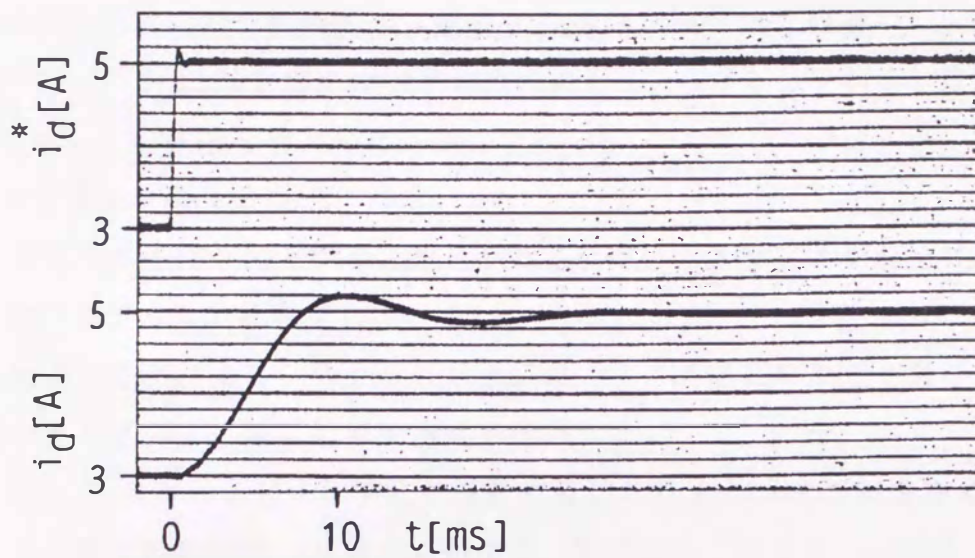
$$K_i=0.75, K_p=1.73, K_d=-0.24, \sigma=4.0$$

(a) 実験値



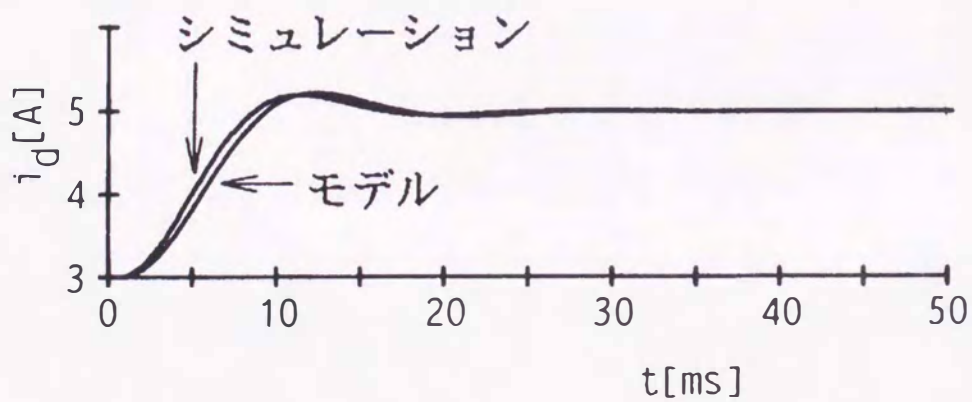
(b) シミュレーション値

図 3.9 ステップ応答 ($\sigma = 4.0$)



$K_i=0.43, K_p=0.88, K_d=-1.27, \sigma=5.0$

(a) 実験値



(b) シミュレーション値

図 3.10 ステップ応答 ($\sigma = 5.0$)

3.3 結論

本章では、M系列信号を用いた相互相関関数による制御対象の同定と、I-PD制御ゲイン決定法を高速演算が要求される電流制御系に適用した。このとき、従来の相互相関関数や制御ゲイン演算より高速な演算法を提案して、汎用の整数形16ビットCPUを用いた実験装置に適用し、シミュレーション結果と比較しながらその性能を検討した。

主な改善点と特長を以下に示す。

- (1) 演算の高速化を図るため、固定小数点を用い、アセンブリ言語ですべてのソフトウェアを開発した。
- (2) M系列信号によるシステムの同定において、M系列の長さとは異なる個数の重み系列を正確に計算する式を導出した。
- (3) 過去のM系列をビット単位で保存することによりメモリ消費を低減し、また相互相関関数の演算においては、積ではなく和または差を行うことにより演算の高速化を達成した。
- (4) 伝達関数変換やI-PDゲインの決定において演算式の時間の項を分離することにより、演算の高速化を図った。