ディジタルシグナルプロセッサと電動機制御

正 員 松 井 信 行 (名古屋工業大学 工学部 電気情報工学科)

1. はじめに

「ソフトウェア制御」という用語に明確な定義はないが、制御に必要な演算や判断、あるいは比較などの制御処理を従来のアナログ回路やディジタル回路によらずに、制御ループ内のコンピュータのプログラムによって実行する制御形態を指すものと一般には理解されている⁽¹⁾⁽²⁾。ソフトウェア制御によってもたらされる制御系の小形化、標準化、自動検査などの側面はもとより、特性上の観点からフレキシビリティの向上、ドリフトやオフセットからの解放が高く評価され、急速に普及していることは周知のとおりである。

このような発展をみているソフトウェア制御は、アナログ方式で実績のあった制御手法をソフトウェアに置換することで始まった第1世代を終え、ソフトウェアの特徴を生かした多機能性、操作性、インテリジェント性を志向する第2世代を迎えている。そして、これを支えるマイクロエレクトロニクスとして、ディジタルシグナルプロセッサ(以下、DSPと略記)が大きなインパクトを与えている。本稿では主として筆者の研究室の成果をもとに、DSPの電動機制御への応用を概観してみたい。

2. プロセッサ制御の問題点とDSPの適用効果

電動機、なかでも電力変換器のソフトウェア制御に携わる者にとって、CPUの処理時間と分解能が大きな課題であった。従来の制御アルゴリズムをそのままソフトウェア化しても、ソフトウェアの実行時間によるサンプリング周期への制約、あるいはメモリ空間やI/Oの制約などのため、必ずしも良好な特性が得られるとは限らず、かえって特性悪化をもたらすという経

験は誰しもがもっていることだろう。

電動機制御系は通常外側から位置ループ、速度ループ、電流ループを備え、各制御ループは対象となる系の時定数と要求される応答速度によってその細目が決められる。例としてロボットの電動機制御系を考えると、位置および速度ループの応答周波数はそれぞれ10および50 Hz程度、またNC工作機の場合にはそれぞれ10 Hz、100 Hz程度であると言われている。ソフトウェア制御時のサンプリング周期は一般には応答周波数から決まる周期の1/6~1/10程度に選ぶとされているから、位置ループで2 ms程度、速度ループで500 μ s前後になる。これに対して電流ループの応答周波数は500~1,000 Hz、あるいはそれ以上が要求されるので、200~100 μ s以下のサンプリング周期が必要とされよう。現用の16ビットプロセッサによる電動機のソフトウェア制御系の多くが位置、速度ループにとどまっているのは、このような理由によるものと考えられる⁽³⁾。

```

*** 0m PI SUBROUTINE *****
* WPICAL  LT  DELWM
*          MPY  KP
*          PAC
*          SUB  D7FFF
*          BLZ  KPNOF

* KPOF    LAC  D7FFF
*          SACL KPWM
*          B    CALKI

* KPNOF   PAC
*          SUB  D8000
*          BGEZ KPNOUF

* KPUF    LAC  D8000
*          SACL KPWM
*          B    CALKI

* KPNOUF  PAC
*          SACL KPWM

* CALKI   ZALH  SIGHWM
*          ADDS SIGLWM
*          MPY  KI
*          APAC
*          SACL SIGLWM
*          SACH SIGHWM

*          ADDH  KPWM
*          SACH  ISQ

```

図1 速度PI演算プログラム(TMS32010ベース)

Application of Digital Signal Processors to the Motor Control.
By Nobuyuki Matsui, Member (Department of Electrical and
Computer Engineering, Nagoya Institute of Technology).

図 2 DSP の応用領域

制御アルゴリズムとして、最も一般的な PI 制御を DSP (TMS 32010) のソフトウェアで構成する例を図 1 に示す。この例ではステップ数で最大 22 ステップ、実行時間で最大 4.4 μ s で処理が完了し、上記の問題点解消の有力なツールであることがわかる。

3. DSP の概要

3.1 DSP の特徴

DSP は独特のハードウェア、バス構造、命令形態を有し、汎用マイクロプロセッサに比して 10 倍以上の処理速度の向上が実現されている。その主な特徴は下記のとおりである。

(1) 内部バスはプログラム用とデータ用に完全に分離し、プログラムメモリとデータメモリが独立したいわゆるハーバード形アーキテクチャを採用し、パイプライン処理が行える。

(2) ハードウェア乗算器を内蔵し、加算器と連動して高速積和演算処理を実現している。

(3) 1 命令が 50~200 ns のサイクルタイムで実行でき、かつ、ほとんどの命令が 1 サイクルで実行できる。

(4) マイクロプログラム方式を採用し、1 命令で複数の機能が実行できる。

表 1 DSP の比較一覧

項目	TI TMS32010	TI TMS320C25	日電 μ PD 7720	日電 μ PD 77230	モトローラ DSP 56000	AD ADSP-2100	沖 MSM 6992	富士通 MB 8764	松下 MN 1901	日立 HD 61810
マシンサイクル(ns)	200	100	250	150	97.5	125	100	100	250	250
データメモリ空間 (word)	144	64 k	128	8 k	128 k	32 k	64 k	1 k	4 k	328
プログラムメモリ空間 (word)	4 k	64 k	512	4 k	64 k	16 k	64 k	4 k	2 k	512
オンチップデータRAM (word)	144	544	128	1 k	512	—	265	256	66	200
オンチップデータROM (word)	—	—	512	1 k	512	—	—	—	—	128
オンチッププログラム RAM (word)	—	256	—	—	—	16	—	—	—	—
オンチッププログラム ROM (word)	1.5 k	4 k	512	2 k	2 k	—	1 k	1 k	2 k	512
データ長(浮動小数点) (bit)	16	16	16	(32)	24	16	(22)	16	16.24	(16)
乗算器	16×16=32	16×16=32	16×16=31	32×32=55	24×24=48	16×16=32	22×22=22	16×16=26	20×20=32	16×16=20
ALU/ACC	32/32	32/32	16/16	55/55	56/56	16/40	22/22	26/26	24/24	20/20
パラレルI/O	8×16	16×16	1×8	1×16	1×8	1×16	1×22	1×16	1×16	1×16
シリアルI/O	—	1	1	1	2	—	—	—	2	1
DMAインタフェース	NO	YES	YES	NO	YES	YES	YES	YES	YES	YES
外部割込み	1	3	1	2	2	4	3	1	1	—
WAIT機能	NO	YES	NO	NO	YES	YES	NO	NO	NO	NO
製造プロセス	2.4 μ NMOS	1.8 μ CMOS	3.0 μ NMOS	1.5 μ CMOS	1.25 μ CMOS	1.5 μ CMOS	2.0 μ CMOS	2.3 μ CMOS	2.0 μ CMOS	3.0 μ CMOS
パッケージ	40 DIP	68 PGA/ PLCC	28 DIP	68/208PGA	88 PGA/ PLOC	100 PGA	132 PGA	88 PGA	84 FLAT	40 DIP
消費電力(W)	0.9	0.5	0.95	0.8	1.0	0.5	0.4	0.3	0.25	0.25

表1に現在の代表的なDSPの比較一覧を示す。第1世代のDSPは16ビット固定小数点演算方式のものがほとんどであったが、第2世代では16ビット、22ビットの浮動小数点を採用するものが出現した。また、第3世代では32ビット浮動小数点方式のものが各社から発売され、今後はこの方式が主流となってくるだろう。

図2はDSPの主たる応用領域と具体的な手法を示したものである。

3.2 DSPのソフトウェア開発環境

図3は一般的なDSPの開発環境の構成である。ホストコンピュータとしてVAX-11 (VMS, UNIX), IBMP (MS-DOS), PC9801 (MS-DOS)などが用いられることが多い。ホストコンピュータ上でエディタを用いて高級言語(特にC言語)、あるいはそれぞれのDSPに用意されているアセンブリ言語によってソースプログラムを作成する。このソースプログラムをコンパイラ、アセンブラにかけてオブジェクトプログラムに変換し、更にリンカーを通して実行可能なオブジェクトモジュール(ロードモジュールともいう)を作成する。

実行可能なオブジェクトモジュールは、一般的に以下のような方法で評価できる。

ソフトウェアシミュレータは、実際にハードウェアで実行させる前にホストコンピュータ上でソフトウェアの機能チェックを行うものである。この場合、リアルタイムでソフトウェアを実行することはできず、そ

の実行速度はホストコンピュータのCPUの速度で決まる。

価格的にやや高価であるが、最も強力なツールはICEと呼ばれるハードウェアのエミュレータである。これを用いればソフトウェアの評価はもとより、ハードウェアの評価(暴走の検出、信号のトレースなど)がリアルタイムで実行できるので、ターゲットシステムのデバッグを行ううえで極めて有用である。

また、ハードウェアの評価はできないがソフトウェアの評価をリアルタイムで行うためのツールとして、一般に評価用ボードと呼ばれるものが各社から提供されている。評価用ボードはスタンドアロンで、RS232C経由でホストコンピュータからコマンドを与えるタイプのものと、パソコンなどのアドオンボードになっているタイプがある。

4. ブラシレスモータのDSP制御例

4.1 DSPによる電流制御の考え方

一般に交流電動機の制御系では、電流制御精度が運転特性を直接左右する。電流制御法としては瞬時値比較方式や三角波比較方式が広く用いられているが、これらの制御方式をそのままソフトウェア化しても必ずしも良い結果が得られない⁽⁴⁾。前述のようにDSPは高速積和演算機能を有するので、筆者らはこれを積極的に用いた演算方式の電流制御を提案している。

図3 DSPの開発環境

図4 ブラシレスモータの電流制御系

```

*
* *** CURRENT 3/2 TRANSFORM *****
* C32ID  LT  IU
*        MPY SIN120
*        PAC
*        LT  IV
*        MPY SIN000
*        APAC
*        SACH ID,4
* C32IQ  MPY COS000
*        PAC
*        LT  IU
*        MPY COS120
*        APAC
*        SACH IQ,4
*

```

図5 座標変換プログラム(TMS32010ベース)

いま、ブラシレスモータの電流制御系を図4のように考えよう。すなわち、電流、電圧をよく知られた三相／二相変換によって直流量に変換し、直流モデルを用いて電流制御則を実行した後、逆変換によって三相インバータの指令電圧 v_3^* を出力する。ここで、直流モデルに基づくブラシレスモータの電圧方程式は次式で与えられる。

$$v = (R + LP)i + e \dots\dots\dots (1)$$

ただし、 v, i, e : 印加電圧、電流および逆起電力ベクトル

ここで用いる座標変換を電子回路的に対処しようとする一般にはかなり複雑となるが、変換そのものが積和演算による行列計算であるから、DSPのソフトウェア処理では図5に示すようにわずか13ステップ、処理時間にして2.6 μ s程度で実行できる。この場合、三相／二相変換は三相の対称性から事実上二相／二相変換になっていること、また $\sin/\cos$ 関数値を得る時間が考慮外になっていることに留意したい。メモリに制限がない場合は所望の精度で電気角360°に対して関数を用意できるが、実際にはメモリ容量の制約から180°、あるいは90°分のテーブルを用意して補間法により所望の関数値を得なければならず、この補間に要する時間が図5に加味されることになる。

さて、ソフトウェア制御にあたっては(1)式を適当な近似のもとで離散値系に書き改め⁽⁵⁾

$$\begin{aligned} \bar{v}^*(n) = & 2Ri(n-1) - \bar{v}(n-1) + 2e(n) \\ & + \frac{L}{T}\{i^*(n+1) - i(n-1)\} \dots\dots (2) \end{aligned}$$

$$\begin{aligned} e(n) = & \theta \left[Li_q(n-1) + T\{\bar{v}_q(n-1) - K_e\theta - Ri_q(n-1)\} \right] \\ & K_e \end{aligned} \dots\dots\dots (3)$$

ただし、 K_e : 起動係数が得られる。ここで、*印を付した諸量は指令値を示し、バーを冠した諸量はサンプル点間の平均値である。上式によれば、サンプル点 $n \sim n+1$ で印加すべ

図6 PWM アルゴリズム

き指令電圧 $\bar{v}^*(n)$ をサンプル点 n 以前の情報で予測演算できるので、事実上のむだ時間の解消が可能となる。

こうして得られた $\bar{v}^*(n)$ を図4に示すように再び座標変換して三相上の電圧ベクトル v_3^* に変換し、これをもとにPWM制御を実行する。このとき、 v_3^* をそのまま三角波比較の信号とする方法もあるが⁽⁶⁾、ここでは以下の二方法⁽⁷⁾を説明しよう。

(1) ベクトル選択法 図6(a)において、 v_3^* の存在領域により、例えば①なら $v(100)$ 、②なら零ベクトルというように最も近い電圧ベクトルを選択する。

(2) 平均電圧法 図6(b)に示すように v_3^* に隣り合うベクトルと零ベクトルをそれぞれ対応する時間だけ出力して、等価的な v_3^* を実現させる。

4.2 制御特性⁽⁷⁾

TMS32010を用いた制御システムはレゾルバからの位置情報の処理、位置情報の差分による速度演算と速度ループ(PI演算)の演算および前述の電流制御を実行する。前述の平均電圧法によるPWMの実行には時間管理が必要となるので、内部カウンタをもた

表2 PWM制御法の比較

	ベクトル選択法		平均電圧法	
	メモリ容量 (ワード)	演算時間 (μ s)	メモリ容量 (ワード)	演算時間 (μ s)
位置・速度演算	180	25	180	25
速度PI演算	60	12	60	12
電流制御	430	37	520	50
定数・sinデータ	320	—	1160	—
合計	990	74	1920	85

図7 電圧・電流波形(定格時)

図 8 トルク電流のステップ応答

ないTMS32010では8ビットカウンタを外付けしている。表2は各制御処理に要するメモリ容量と処理時間を比較したものであるが、いずれのPWMアルゴリズムを用いても演算ピッチで100μs以下の制御プログラムが構築でき、表示、通信その他のサービスルーチンを含めてもサンプリング周期100μsが実現できる。

図7に定格運転時の電圧、電流波形の一例を電流の波形分析結果と共に示した。ともに電流ループのサンプル周期は100μsであるが、平均電圧法によるPWMは v_3^* に対するベクトルの忠実度が高いため、図からも明らかなように5kHz以下の高調波が減少して波形の改善効果は大きい。しかし聴感での騒音レベルという点では、ベクトル選択法に捨て難い魅力があることを付記しておこう。

図8はトルク電流指令を2Aから10Aにステップ状に変化させた場合の応答で、PWM法の差異に無関係に約3ステップ(300μs)で整定し、十分な高速応答を示していることが理解できよう。

4.3 他の制御システムへの展開

上記の制御法は二軸上の電圧方程式を出発点にしているので、そのまま同期機形DDモータやハイブリッドステッピングモータにも適用できる。また、近年高調波障害や力率の点で注目させている力率1の正弦波入力のコンプータ制御にも簡単なアナロジーによって応用できる⁽⁹⁾。更に、PWMコンバータ+電圧形PWMインバータを1DSPで一括制御することも可能である⁽⁹⁾。

また、二軸の電圧方程式に若干の差はあるものの、電流ループを含む誘導電動機のベクトル制御の全ソフトウェア制御も実現している⁽¹⁰⁾。

5. DSPによるインテリジェントコントローラの実現

前章からも明らかなように、DSPを用いると微分方程式レベルの制御則をそのままリアルタイムで実行することができるので、単なる制御にとどまらず上位系との通信機能、状態表示、故障検出と自動復帰、適

図 9 位置角補正効果

図 10 ソフトウェア上の印加電圧
 v_{dc} : コントローラ内 d 軸印加電圧指令値
 v_{id} : デッドタイム補償後のコントローラ内 d 軸印加電圧指令値

図 10 ソフトウェア上の印加電圧

応能力などをもつインテリジェントコントローラ実現へ一歩近付くことができる。ここでは、制御機能に直接関係するものに限定して説明しよう。

5.1 位置角補正⁽⁵⁾ 位置検出器にレゾルバを用い、位置角情報の差分から速度を得る方法では、角度および速度の分解能 $\theta_{\min}$ 、 $\omega_{\min}$ は励磁周波数を f 、高周波クロック周波数を f_c 、極対数を p とすると

$$\left. \begin{aligned} \theta_{\min} &= f / (p \cdot f_c) \\ \omega_{\min} &= f^2 / (p \cdot f_c) \end{aligned} \right\} \dots \dots \dots (4)$$

と記せる。分解能をあげるため励磁周波数を下げると検出周期が長くなるので、図4で電流制御周期に同期して位置情報を得ることは一般には困難である。また、厳密には図4の二相/三相変換と三相/二相変換で用いる位置角も違ってくる。このような場合には位置情報の周期内で

$$\theta(t) = \theta(k) + \dot{\theta} \cdot t \dots \dots \dots (5)$$

と補正演算を行うことが有効である(k :位置情報のサンプル数)。図9は補正効果の一例である。

5.2 電圧情報

制御情報処理には正確な印加電圧情報がしばしば必要になるが、インバータの飽和やデッドタイムのために指令電圧と実際の印加電圧には差がある。一方、PWM電圧の正確な検出は経験的にも難しい面があり、もし可能でもコストアップ要因となる。コントローラにはPWM情報と直流電圧情報があるから、素子のターンオン、オフ時間とデッドタイムを知れば、

ソフトウェア上で指令電圧に対する実際の印加電圧を求めることができる。

図 10 はこの結果の一例である⁽¹⁰⁾。ソフトウェア上でこの電圧情報をフィードバックすれば、何らの外付け部品を用いることなく V/f 制御誘導電動機の無負荷時の振動抑制が可能である⁽¹¹⁾。

検出器レスでソフトウェア上で得られる印加電圧情報は、ベクトル制御精度向上のための磁束シミュレー

タや誘導機の低速駆動特性の改善などに大きく貢献するものと考えている。

5.3 電動機パラメータの同定⁽¹²⁾⁽¹³⁾

ベクトル制御や現代制御の適用を考える際には、正確な電動機定数の把握（静的同定と呼ぼう）はもとより、運転状況により時々刻々変化する定数の把握（動的同定）が必須である。適応制御理論はこれに対する有力なツールではあるが、同定すべきパラメータの数が増えると現状の DSP では処理時間的に問題も残る。

図 11 はブラシレスモータ制御系に現代制御でいう STR（セルフチューニングレギュレータ）⁽¹²⁾と、MRACS（モデル規範適応制御）⁽¹³⁾の考え方を導入した例である。この例では、第 4 章の電流制御系を対象とし、同定アルゴリズムは処理時間の点から、(1)式

図 11 ブラシレスモータのパラメータ同定

表 3 同定法の比較

	STR	MRACS
使用 CPU	TMS 320 C 25	TMS 320 C 25
制御周期	100 μ s	100 μ s
演算時間	約 97 μ s	約 96 μ s
プログラム容量	約 2.5 kワード	約 2.5 kワード
同定方法	電圧、電流より直接演算	規範モデルとの比較による PI 演算
同定誤差に対する補正法	<ul style="list-style-type: none"> 同定値に対する条件づけ 同定値を 256 回平均 	特になし
デッドタイムに対する補正	補正あり	補正あり
同定値の精度	定常状態	L の同定値に誤差大 低速時に MI_r の同定値に誤差大
	過渡状態	<ul style="list-style-type: none"> 同定開始直後にオーバーシュート 収束性は PI ゲインに依存 200 μs ごとに同定値を更新

(注) 制御周期 100 μ s を超えるため L および MI_r の同定を交互に行っている。

図 12 トルク制御特性

図 13 二次抵抗補償の有無によるトルク

をもとに展開されている。同図(b)は同定結果の一例で、両方式の比較を表3に示す。

5.4 トルクセンサレストルク制御⁽¹⁴⁾

パラメータ同定の結果、トルクセンサを用いずに電流制御のみによるトルク制御が可能となる。図12(a)はフェライト永久磁石モータによる定電流制御例であるが、時間と共に温度上昇によってトルク定数 $K_T (=MI_f)$ が減少する結果、トルクそのものが減少する。同図(b)はトルク定数の同定を付加して定トルク指令で運転した例で、温度上昇と共に電流指令も変更されて定トルク制御が実現している。トルク定数の同定精度は、現状では1~2%程度なので希土類磁石モータに応用するには精度的に今一步の向上が望まれる。

5.5 ベクトル制御の R_2 補償⁽¹⁰⁾

筆者らがさきに発表したDSPによる非干渉ベクトル制御系では、電流ループ処理を含むベクトル制御処理を85 μ s、 R_2 補償を60 μ s (TMS32010 ベース)で実行する。図13はその一例である。

5.6 電流センサのオフセット補償

図4の制御系で電圧情報を5.2節の方法によっても、完全な意味での全ディジタル制御系は実現できない。すなわち、現状では電流検出はアナログによっており、ここにオフセットのドリフトが潜んでいるからである。しかしながら、(1)式で記述されるタイプの電動機ではこのオフセットの補償をソフトウェアで実行できることが示された⁽¹⁵⁾。今後、誘導機への適用に期待がもたれる。

6. あとがき

DSPの高速演算能力を前提にすれば、電動機制御上のハードウェア的制約は次第に弱くなっていくように感じられる。それだけに、制御アルゴリズムそのものが問われる時機が到来してこよう。

一方、DSPそのもののハードウェアについては、制御用I/Oを内蔵した専用DSP、一層の演算能力の強化と高速化がASICとのからみで期待される。次世代DSPによるチューニングレスコントローラの実現

は既に掌中にあると言えるだろう。

なお、本稿は協同研究者の岐阜工業高等専門学校 竹下隆晴氏を始め、筆者の研究室にDSPを導入した昭和58年秋以後の歴代の院生である亀田晃史(松下電産)、木下久史(日本電装)、水谷清信(住友重機)、山井広之(ダイキン工業)、大橋敬典(日立製作所)、山田英治(トヨタ自工)、赤尾憲彦(トヨタ自工)、神谷新吾(豊田工機)君ら、および岩崎 誠(現在、名古屋工業大学博士課程)、執行正謙(同修士課程)君らの努力に負うところが大きい。また、日本テキサスインスツルメンツ(株)名古屋営業所 神谷 稔氏には日ごろ有益な御助言をいただいている。末尾ながら謝意を表す。
(平成元年3月2日受付)

文 献

- (1) 「メカトロニクスのための新しいサーボ技術」, 昭58 精機学会講習会
- (2) 「ソフトウェアサーボ」, 日経メカニカル (昭63-4)
- (3) N. Matsui & T. Takeshita: "Digital Signal Processor-Based Controllers for Motors", SICE '86 Proc. ESS. 1-8 (1986)
- (4) 山井・神谷・松井: 「DDモータのソフトウェア化制御」, 昭62 電気学会全大, No. 609
- (5) 竹下・亀田・大橋・松井: 「デジタルシグナルプロセッサによるブラシレスモータの高速電流制御法」, 電学論 B, 106, 753 (昭61-9)
- (6) 松井・亀田・竹下: 「DSPによるブラシレスモータのソフトウェア化非干渉電流制御」, 同上 D, 107, 215 (昭62-2)
- (7) 大橋・松井: 「ブラシレスモータの高速電流制御法とそのパラメータの依存性」, 昭62 電気学会産業応用部門全大, No. 99
- (8) 竹下・岩崎・松井: 「三相PWMコンバータのパラメータ変動を考慮した電流制御法」, 電学論 D, 107, 1339 (昭62-11)
- (9) 神谷・松井: 「インバータ・コンバータ同時制御による電圧形ACドライバのコンデンサ容量低減」, 昭62 電気学会東海支大, No. 144
- (10) 岩崎・松井: 「DSPによる誘導電動機二次時定数同定機能を有するベクトル制御システム」, 電学論 D, 108, 845 (昭63-9)
- (11) 岩崎・松井: 「ソフトウェアデッドタイム補償機能を持つ汎用インバータ」, 昭63 電気学会東海支大, No. 141
- (12) 大橋・執行・松井: 「パラメータ同定機能をもつブラシレスDCモータの適応電流制御」, 電学論 D, 108, 1091 (昭63-12)
- (13) N. Matsui & H. Ohashi: "DSP-Based Adaptive Control of a Brushless Motor" Proc. of IAS Ann. Meeting (1988)
- (14) 執行・遠藤・松井: 「パラメータ同定によるブラシレスモータの高精度トルク制御」, 平成元年電気学会全大, No. 636
- (15) 神谷・牧野・松井: 「DDモータの脈動トルクに及ぼす電流センサドリフトの影響とその補償法」, 同上, No. 628