



Title	パワーエレクトロニクスと電動機制御入門
Author(s)	辻, 峰男
Citation	パワーエレクトロニクスと電動機制御入門; 2015
Issue Date	2015
URL	http://hdl.handle.net/10069/35225
Right	

This document is downloaded at: 2019-01-19T08:33:07Z

第 8 章 DSP 制御システム

誘導モータのベクトル制御を例にとり，電流制御や速度制御の方法と制御器の設計，それらを実現するための DSP(digital signal processor)によるシステム構成を述べる。

○ 電流制御と速度制御

図 8-1 に誘導モータのベクトル制御を利用した速度制御系のブロック図を示す。

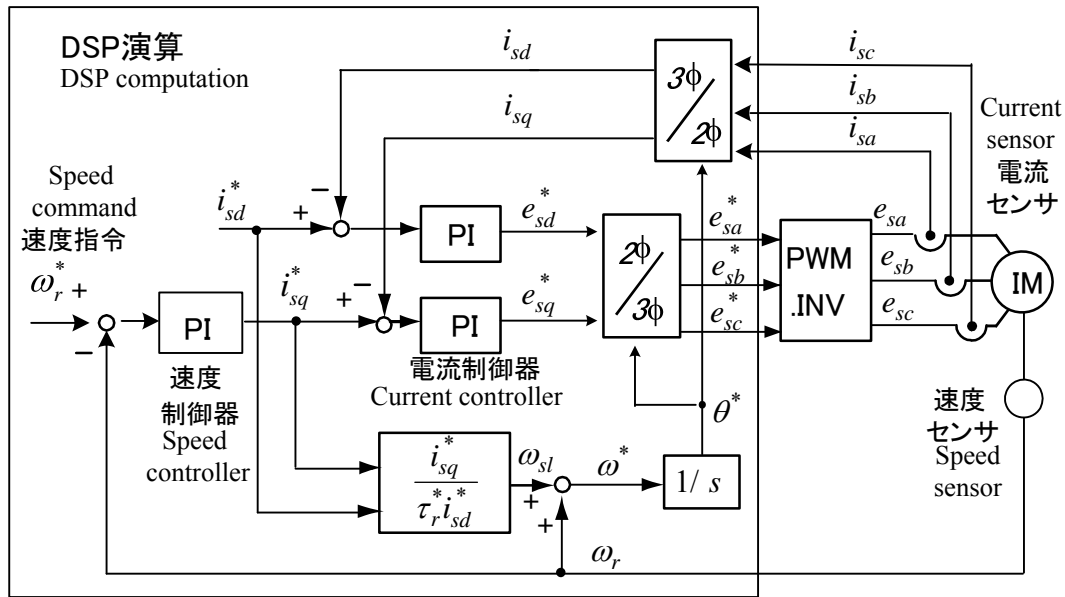


図 8-1 誘導モータのベクトル制御を用いた速度制御システム

電流制御法は種々あるが，交流電流を d - q 軸の直流電流に変換し，**PI(比例+積分)制御器**を用いて制御する方式が一般的である。電流制御の性能をさらに向上するため，非干渉制御が用いられることもある。 d - q 軸の電流制御が瞬時に行われると i_{sq}^* に比例してモータのトルクを瞬時に発生させることができる。これが**ベクトル制御**である。 i_{sq}^* は DC モータの電機子電流に相当し， e_{sq}^* から右側を等価な DC モータと思えばよい。速度制御はベクトル制御とは関係ない。速度を増加させるにはトルクを増加させればよいので，指令速度より実速度が小さいほどトルクを増加させるため，速度 PI 制御の出力を i_{sq}^* とすることで，速度を制御できる。これは次式で表わせる。

$$i_{sq}^* = K_{ps}(\omega_r^* - \omega_r) + K_{is} \int_0^t (\omega_r^* - \omega_r) dt \quad K_{ps} > 0, K_{is} > 0 \quad (8-1)$$

電流の偏差を PI 制御し，その出力を電圧指令とすることも同じ理屈である(我々は電圧を大きくすれば電流が増えることを知っている)。

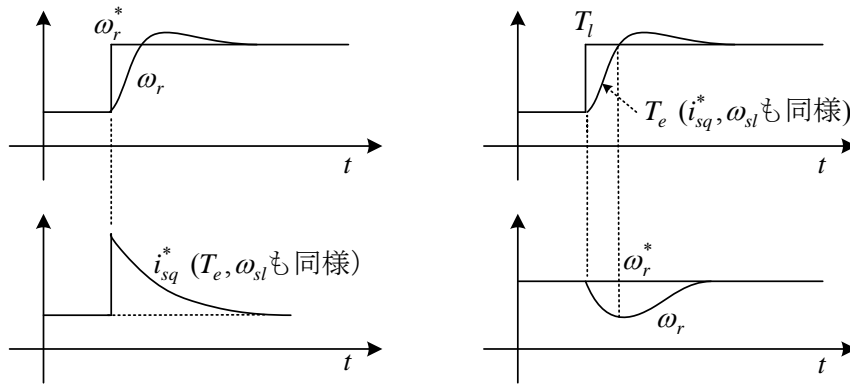


図 8-2 速度指令（左側）と負荷トルク（右側）のステップ変化に対する応答

図 8-2 の速度指令のステップ変化に対する応答では、速度の偏差によりまず P 制御の項が働いて q 軸電流指令が瞬時に立ち上がり、その結果電動機が発生するトルク T_e がベクトル制御により同様に变化する（電流制御が理想的なら）。これにより、電動機が速度が上昇し始める。速度の偏差が徐々に小さくなると P 制御の項は減少するが、I 制御の働きで速度は指令値以上に増加する。なぜなら負荷トルクが一定の場合、 q 軸電流は最後には元の値に戻る。従って、(8-1)の右辺第 2 項について速度指令の変化で新たに生じた偏差の積分値は 0 にならないといけない。これは電流指令がリミッタにかからない場合の話で、リミッタにかかる場合と **ワインドアップ現象** の防止のため通常積分値は増やさないから状況は異なる。次に、負荷トルクのステップ変化では、まず速度が減少し、その結果速度偏差を生じるので PI 速度制御器が働いてトルクが負荷トルクと等しくなるまで増加し、速度は指令値に戻る。速度指令が負の場合（逆回転）にも問題なく動作することを考えよ。

電流制御器の設計

まず、電流 PI 制御器の設計法について述べる。電流制御は高速に行うので、誘導モータを漏れインダクタンスと抵抗（一次+二次）の直列回路として考える。このときのブロック図を示す。付録 1 の T-I 形過渡等価回路で、電流の変化は高速だから周波数が高く励磁回路は無視する。速度に比例する起電力の項は外乱と考える。

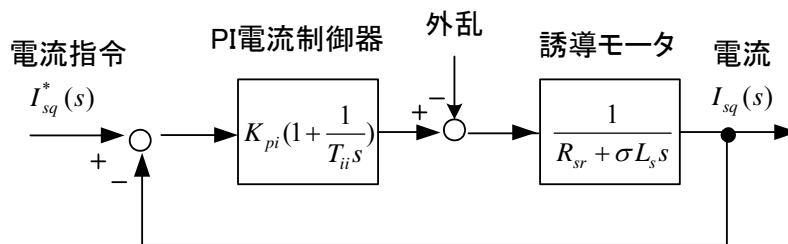


図 8-3 電流制御系のブロック図

電流制御の開ループ伝達関数を求めると次式となる。

$$\frac{I_{sq}}{I_{sq}^*} = \frac{K_{pi}(T_{ii}s + 1)}{(\sigma L_s s + R_{sr})T_{ii}s + K_{pi}(T_{ii}s + 1)} \quad (8-2)$$

$$\text{ここで, } T_{ii} \equiv \sigma L_s / R_{sr} \quad (8-3)$$

とすると、電流の伝達関数は次のように一次遅れ系となる⁽²³⁾。

$$\frac{I_{sq}}{I_{sq}^*} = \frac{K_{pi}}{R_s T_{ii} s + K_{pi}} \equiv \frac{1}{T_{eq} s + 1} \quad (8-4)$$

この伝達関数の**遮断周波数**を ω_c とすると

$$\omega_c = \frac{K_{pi}}{R_{sr} T_{ii}} = \frac{1}{T_{eq}} \quad (8-5)$$

したがって、設計法としては、遮断周波数 ω_c を決めて、(8-3)式より T_{ii} 、(8-5)式より K_{pi} を

求めればよい。

このとき、一巡伝達関数（開ループ伝達関数） G_o は、次式で与えられる。

$$G_o = \frac{K_{pi}}{R_{sr} T_{ii} s} \quad (8-6)$$

$$R_{sr} = R_s + \left(\frac{M}{L_r}\right)^2 R_r = 1.6 + \left(\frac{0.112}{0.1179}\right)^2 \times 0.85 = 2.367\Omega$$

$$\sigma L_s = \left(1 - \frac{M^2}{L_s L_r}\right) L_s = \left(1 - \frac{0.112^2}{0.1176 \times 0.1179}\right) \times 0.1176 = 0.0112\text{H}$$

$$\text{積分時間 } T_{ii} = \frac{\sigma L_s}{R_{sr}} = \frac{0.0112}{2.367} = 0.00473$$

$$\text{比例ゲイン } K_{pi} = \sigma L_s \omega_c = 0.0112 \times \omega_c$$

$$\text{積分ゲイン } K_{ii} = \frac{K_{pi}}{T_{ii}}$$

$$\omega_c = 1500 \text{ で設計すると, } K_{pi} = \sigma L_s \omega_c = 0.0112 \times 1500 = 16.8$$

$$K_{ii} = \frac{K_{pi}}{T_{ii}} = \frac{16.8}{0.00473} = 3552$$

となる。

速度制御器の設計

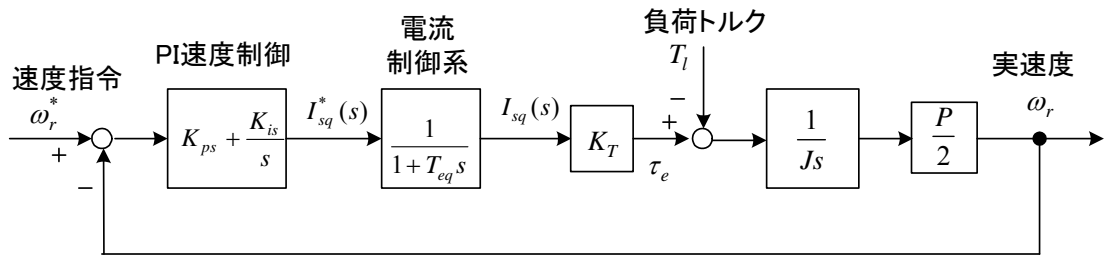


図 8-4 速度制御器設計のためのシステムのブロック図

図 8-4 ではベクトル制御が理想的に行われて、トルクは次式で制御できるものとしている。

$$\tau_e = \frac{PM^2}{2L_r} i_{sd}^* i_{sq}^* \equiv K_T i_{sq}^* \quad (8-7)$$

このブロック図より速度制御系の開ループ伝達関数は次式で表せる。

$$\frac{\omega_r}{\omega_r^*} = \left(K_{ps} + \frac{K_{is}}{s} \right) \frac{1}{1 + sT_{eq}} \frac{PK_T}{2Js} \quad (8-8)$$

これを基に、PI 速度制御系を設計するためのボード線図を図 8-5 に示す。

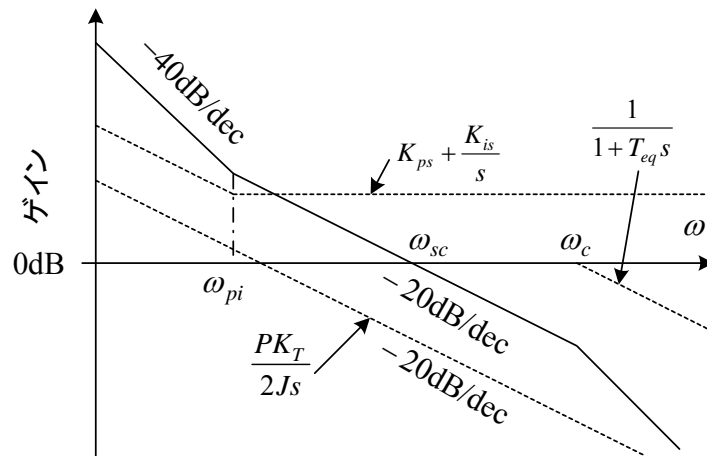


図 8-5 PI 速度制御系の閉ループ伝達関数⁽²³⁾

設計の基本的考え方として、速度制御系の交差角周波数 ω_{sc} 付近では、 -20dB/dec の特性を持つようにして安定性を確保する。これは -20dB/dec の特性が長く続けば位相遅れが 90 度近くになり、 180 度に達しないので不安定とはならないからである。速度制御系の交差角周波数 ω_{sc} が電流制御系の交差角周波数 ω_c に近いとオーバーシュートを生じやすくなるため、

ω_c は ω_{sc} より数倍以上高く設計する。このため、 ω が ω_{sc} 付近では、電流制御の伝達関数は1と考えてよい。また、PI制御器の折れ点角周波数 ω_{pi} は、

$$\omega_{pi} = K_{is} / K_{ps} \quad (8-9)$$

であるが、 ω_{sc} において-20dB/decの傾きを確保するためには、 ω_{pi} は ω_{sc} の1/5以下にする。この結果、 ω_{sc} 付近では K_{is} の項は無視してよい。従って、交差周波数 ω_{sc} は以下のよう求められる。

$$\left| \frac{PK_T K_{ps}}{j 2J \omega_{sc}} \right| = 1 \quad \therefore \omega_{sc} = \frac{PK_T K_{ps}}{2J} \quad (8-10)$$

従って、PI速度制御器のゲインは、 ω_{sc} を与えて

$$K_{ps} = 2J\omega_{sc} / (PK_T) \quad (8-11)$$

とする。積分ゲインは、

$$\omega_{pi} \leq \omega_{sc} / 5 \quad (8-12)$$

のように ω_{pi} を選んで、

$$K_{is} = \omega_{pi} K_{ps} \quad (8-13)$$

で設計する。

ω_{sc} の目安としては、サイリスタレオナード速度制御系で30rad/sが限界、誘導モータの可変速ドライブで50rad/s以上（速度範囲1:100以上）、誘導モータのサーボシステムで200rad/s以上と言われている。また、800WのPM同期モータのサーボシステムを500rad/sで設計した例もある。また、電流制御については、PWM制御のキャリア周波数が10kHz（IGBT使用）の場合に、 $\omega_c = 2000\text{rad/s}$ とした例がある。

以下は設計例である。

$$K_T = \frac{PM^2}{2L_r} i_{sd}^* = \frac{4 \times 0.112^2}{2 \times 0.1179} \times 4.2 = 0.894$$

$$\text{比例ゲイン} \quad K_{ps} = \frac{J\omega_{sc}}{2K_T} = \frac{0.014}{2 \times 0.894} \omega_{sc} = 0.00783\omega_{sc}$$

$$\text{積分ゲイン} \quad K_{is} = \frac{\omega_{sc} K_{ps}}{5}$$

$$\text{積分時間} \quad T_{is} = \frac{K_{ps}}{K_{is}}$$

$\omega_{sc} = 30$ で設計すると、

$$K_{ps} = 0.00783 \times 30 = 0.235 \quad K_{is} = \frac{30 \times 0.235}{5} = 1.41$$

$$T_{is} = \frac{K_{ps}}{K_{is}} = \frac{0.235}{1.41} = 0.167 \quad (\omega_{pis} = \frac{K_{is}}{K_{ps}} = \frac{1.41}{0.235} = 6 \text{ rad/s})$$

図 8-6 に電流制御とベクトル制御が理想的とした場合のシステムのブロック図を示す。負荷トルクを 0 と考えた場合の閉ループ伝達関数は次式で与えられる。

$$\frac{\omega_r(s)}{\omega_r^*(s)} = \frac{as+b}{s^2+as+b}$$

ただし、 $a = \frac{K_{ps}PK_T}{2J}$ 、 $b = \frac{K_{is}PK_T}{2J}$

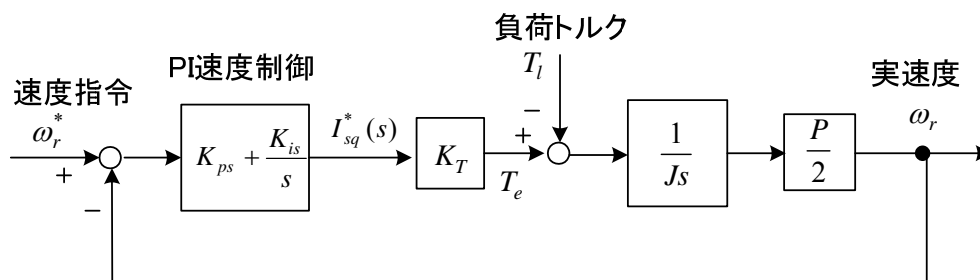


図 8-6 速度制御系のブロック図（電流制御とベクトル制御が理想的）

○ DSP 制御システム

図 8-7 に**デジタルシグナルプロセッサ** (Digital Signal Processor :DSP)によるインバータ-交流モータ制御システムを示す。

整流器、平滑コンデンサ、IGBT インバータ、交流電動機の主回路についてはこれまで述べた。交流機には**トルクセンサ**を介して負荷用の直流機をつないでいる。トルクセンサは軸のねじれを検出するもので、100rpm 以下では測定が困難である。制御回路は DSP を用いて構成しており、直流電圧と電動機の電流（2 相分）を A/D 変換器を通して DSP に取り込む。PWM ゲート信号発生器は変調率（電圧指令）が入力されてゲートパルス PWM インバータに送り IGBT をオン、オフさせる。また、PWM ゲート信号発生器は電力回生時のエネルギーを処理するためブレーキ回路の IGBT もオン、オフする。これらの信号はノイズの影響を受けない光ファイバーで信号を伝えている。ブレーキ回路には抵抗が接続されており、**回生エネルギー**（誘導電動機が発電機として動作）は熱として消費される。この回路がないとダイオードの整流回路は電源にエネルギーを戻せないから平滑コンデンサの電圧が上昇し危険である。PWM ゲート信号発生器から DSP に送られる INT1 信号は、PWM の周期に合わせて DSP に割り込みをかけ、電流検出等に都合の良いタイミングを知らせ、**割り込み処理**を INT1 信号（例えばスイッチング周波数 5kHz なら 200μs）ごとに行う。**ホストコンピュータ**は DSP と接続して、**制御プログラム**を転送したり、DSP の情報を画面に表示したりする役目をもつ。DSP 制御回路や PWM インバータは Myway プラス（株）の製品を用いている。

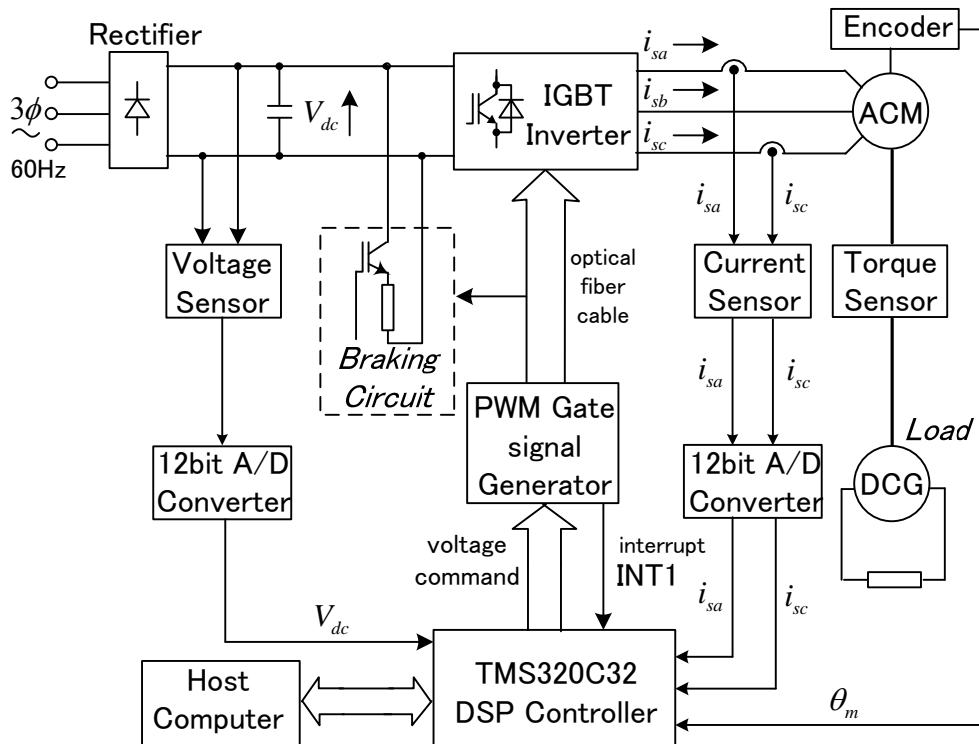


図 8-7 DSP によるインバータ-交流モータ制御システム

図 8-8 に TMS320C32 の機能ブロックを示す。TMS320C32 は、外部クロック 50MHz で命令サイクルタイムは 40ns で動作する。TMS320C33 は、外部クロック 150MHz である。

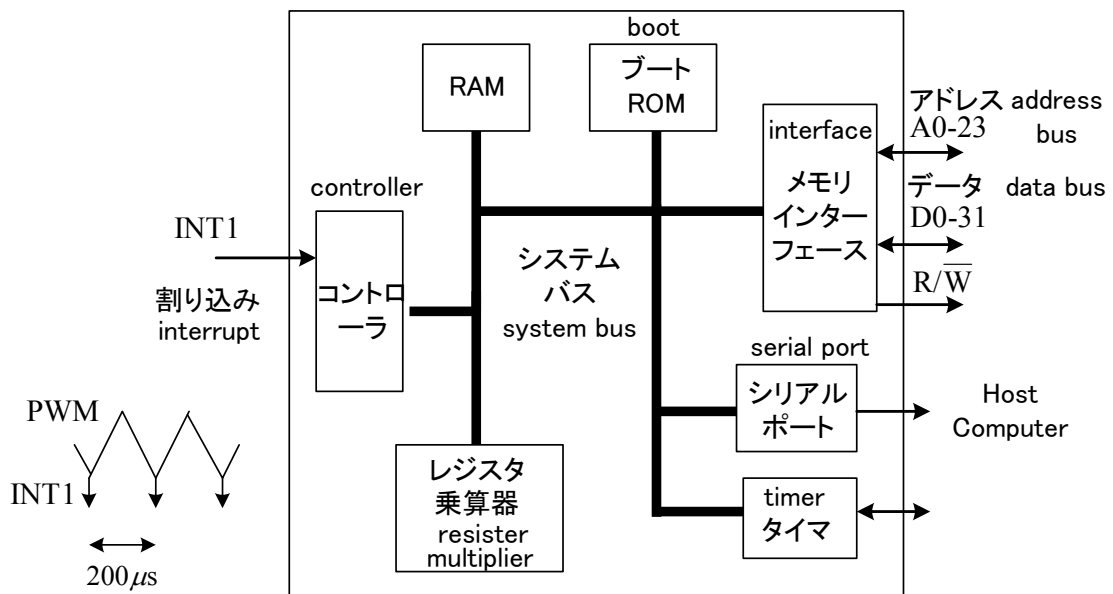


図 8-8 TMS320C32 の機能ブロック (Texas Instruments)

○ ソフトウェア構成

DSP によるデジタル制御を行う場合、微分や積分を伴う制御演算は差分方程式の形に離散化して実現する。PI 制御の離散化についてのべる。

PI 制御は、その入力 $e(t)$ と出力 $u(t)$ に対し、時間領域では次式で表せる。

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt \quad (8-14)$$

これをサンプリング周期 T について離散化すると、積分の近似の違いで次式が考えられる。

$$\textcircled{1} \quad u(k) = K_p e(k) + K_i \sum_{m=0}^k e(m)T \quad (\text{進み矩形近似}) \quad (8-15)$$

$$\textcircled{2} \quad u(k) = K_p e(k) + \frac{K_i}{2} \sum_{m=0}^k (e(m) + e(m-1))T \quad (\text{台形近似}) \quad (8-16)$$

(8-15)と(8-16)を書き換えて、差分方程式の形にすると、次式が得られる。

$$\textcircled{1} \quad u(k) = u(k-1) + K_p(e(k) - e(k-1)) + K_i T e(k) \quad (8-17)$$

$$\textcircled{2} \quad u(k) = u(k-1) + K_p(e(k) - e(k-1)) + \frac{K_i T}{2}(e(k) + e(k-1)) \quad (8-18)$$

デジタル PI 制御演算（電流及び速度）には、(8-17)か(8-18)を用いるが、実際に使う場合には出力の上限と下限いわゆるリミッタをかけておくことが必要である。(8-17)、(8-18)式を用い、 $u(k)$ のみにリミッタをかけると、リミッタにかかったときでも積分器に値が蓄積し、リミッタからぬけたときに大きなオーバーシュートが生じる**ワインドアップ現象**が起きる。しかし、(8-17)や(8-18)を使い $u(k)$ にリミッタをかけるとワインドアップ現象は生じない。

制御プログラムのフローチャートを図 8-9 に示す。制御プログラムは図 8-9 に示すように 3つのプログラムから成る。DSP はこの 3つのプログラムを同時に実行することはできないが、優先順位を決めて順番に実行する。人間も時間を決めていろいろの仕事をしているのと同じである。DSP は、まず最初に**メインプログラム**を上から順番に実行していく。DSP とつながっているハードウェアの初期化や設定が主で、大部分は Myway プラス（株）が作ったソフトを利用している。AD 変換器や PWM 発生器の中にもメモリがあって、そのデータを変えることでいろんな使い方ができる（これが初期化や設定である）。割り込みをどんなふうにするか（たとえばタイマ割り込みの時間や割り込みプログラムの保存されている場所）を DSP に設定しておくことも必要である。その後、DSP を割り込み受付可能状態にする。それ以前は割り込みを受け付けない。PWM の動作開始で PWM ゲート信号発生器が動き始める。その後、波形を表示するため、電流や速度などのデータをホストコンピュータに送る仕事や逆にホストコンピュータからのデータを DSP に送る仕事を永遠に続けることになる（DSP をストップするまで）。

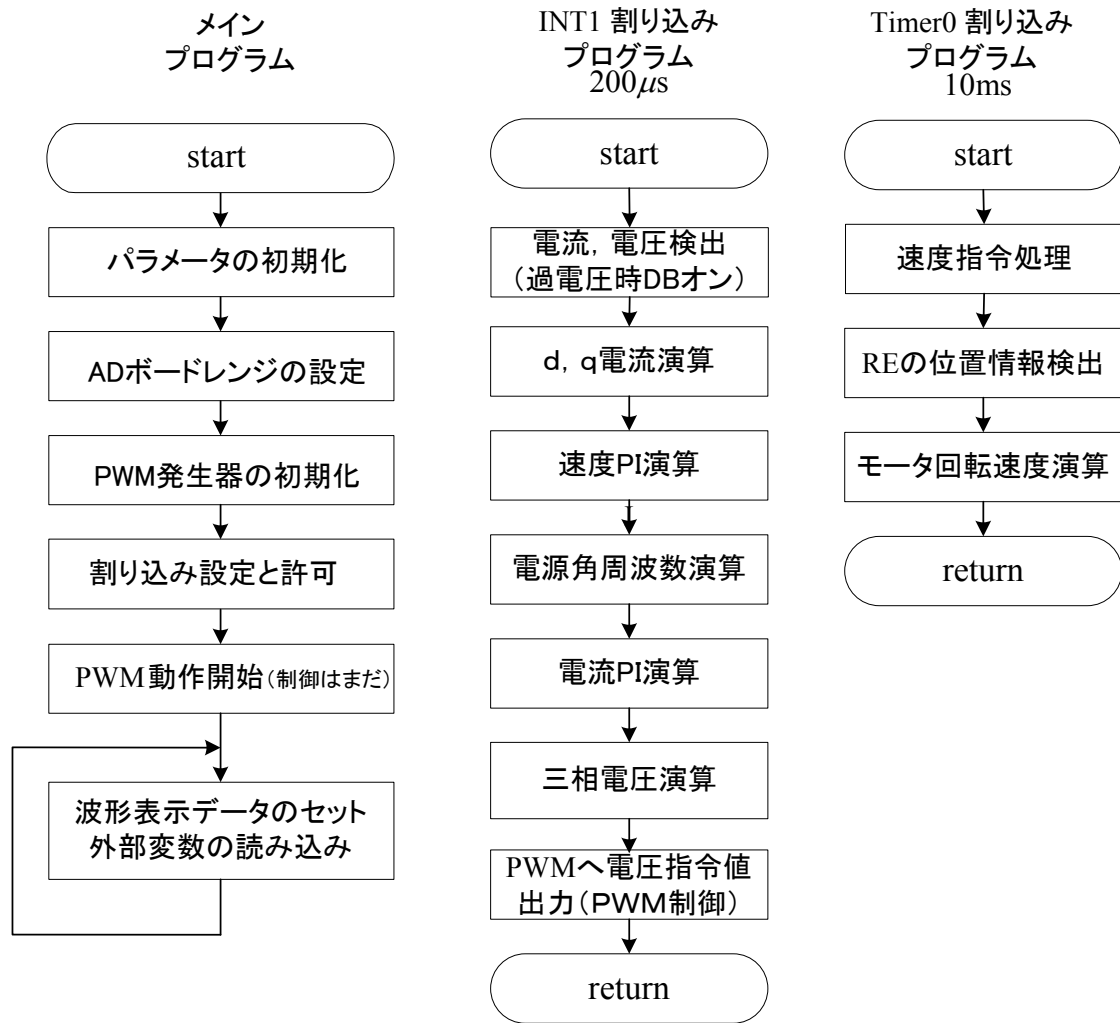


図 8-9 ベクトル制御プログラムのフローチャート

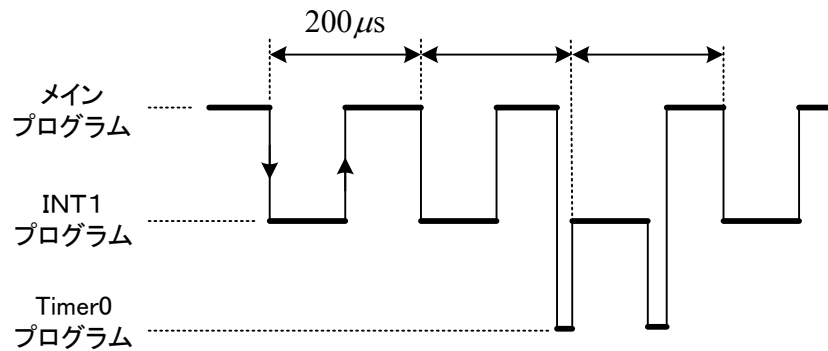


図 8-10 プログラムの実行シーケンス
(DSPは同時に2つのプログラムを処理できない！)

割り込み許可後、ゲート信号発生器から INT1 の信号が DSP に入ると、DSP はメインプログラムの仕事をやめて **INT1割り込みプログラム**を実行する。INT1 のプログラムがベクトル制御や速度制御の中心で、我々が作るプログラムの大部分がこれである。INT1 のプログラムが終わると、DSP は先ほど止めていたところからメインプログラム（波形表示データのセット）を繰り返し実行する。仕事をしていたら電話が鳴って（これが割り込み）、電話の用事を済ませてから元の仕事に戻るといったところである。INT1 の割り込み信号は 200 μ s ごとに来るので、INT1 のプログラムの実行時間は 200 μ s 以下にしておかないと正常に動作しなくなる（暴走する）。電話が 10 分おきにかかってくるとして、話が 10 分以内に終わらないと次の電話に出られないことと同じである。**Timer0 割り込み**は、10ms ごとに DSP に信号が入り、実行される。このタイマーは DSP 自身で作っている。ちょうど自分の欲求で 1 時間ごとにお茶を飲むのと似ている。INT1 を実行中に Timer0 割り込みが入っても、優先順位があって INT1 が終わってからでないと Timer0 は受け付けられない。一番優先順位が低いのがメインプログラムで、これは 2 つのプログラムが走っていないときに実行される。電話中はお茶を飲まず、お茶を飲んでいても電話があれば出て、それらの時間以外で仕事をするといったところ。Timer0 割り込みプログラムは、比較的ゆっくりやればよい制御等に利用する。

実験の手順の説明

上記 3 つの実験プログラムを C 言語で作成し、ホストコンピュータに入れる。DSP の電源を入れて、ホストコンピュータと DSP が通信できる状態にする。DSP 側では ROM のブートプログラムが動く。次に、C 言語のプログラムを**アセンブル**して機械語（DSP にとって C は理解できず機械語のみ分る）に直し、Myway プラス（株）が作ったプログラムとも一緒にする（**リンク**という）。これを、ホストコンピュータから DSP へ

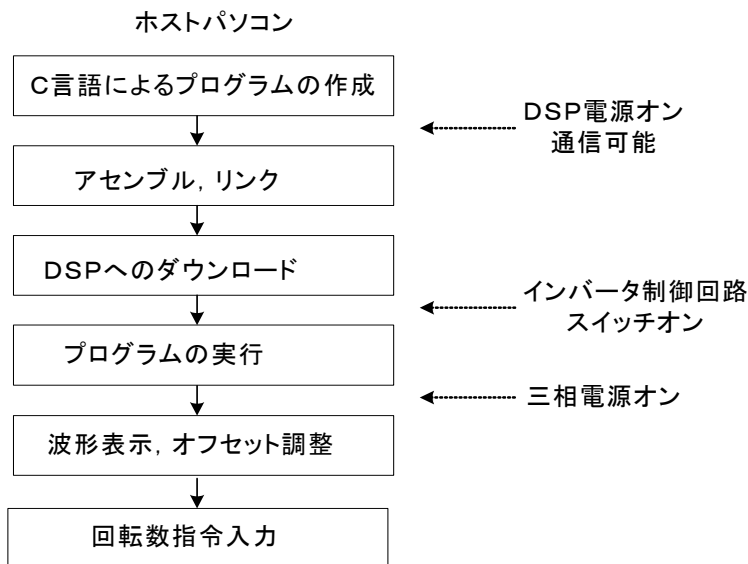


図 8-11 実験の手順

送り、DSP はメモリ RAM にこれらのプログラムを保存する（**ダウンロード**という）。このあとホストからコマンドを入力すると DSP が読み込み、それに応じた仕事をしてくれる。これらの通信やコマンドの処理プログラムは Myway プラス（株）が作ったもので、パソコンと DSP 両方に必要である。

○ 同期電動機の世界制御システム

図 8-12 にベクトル制御を利用した同期電動機の世界制御システムを示す。位置センサにより磁極位置が判るので、それに基づいて座標変換を行い電流制御することでベクトル制御が達成される。すべり周波数演算がない点が誘導機の世界制御との差である。

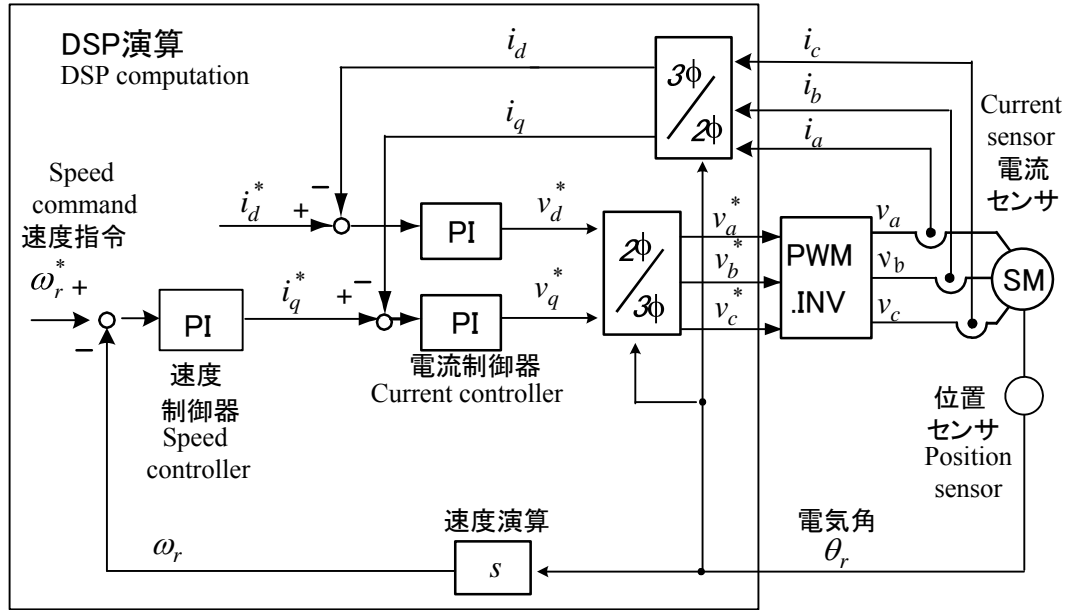


図 8-12 ベクトル制御を用いた同期電動機の世界制御システム

○ 電流の非干渉制御

図 8-1, 図 8-12 では、電流制御として PI 制御を考えた。この部分には、一般に**非干渉制御**(decoupling control)を追加して用いられている。非干渉制御は、 d 軸電圧を増加させると d 軸電流だけが增加するのではなく、 q 軸電流も変化する(干渉分と呼ぶ)ので、これを防ぐ目的がある。 q 軸電流についても同じことが言える。非干渉制御は、干渉分を演算して加えてやることで、PI 電流制御の出力電圧が、その軸の電流成分だけに作用するようにする。誘導電動機と永久磁石同期電動機の非干渉制御を用いた電流制御の例をそれぞれ図 8-13, 図 8-14 に示す。これは、ベクトル制御に使用する磁束の位相に同期した回転座標系の式から容易に得られる。(3-17)より

$$e_{sd} = (R_s + \sigma L_s p) i_{sd} - \omega^* \sigma L_s i_{sq} + \frac{M}{L_r} p \psi_{rd} - \frac{\omega^* M}{L_r} \psi_{rq} \quad (8-19)$$

$$e_{sq} = \omega^* \sigma L_s i_{sd} + (R_s + \sigma L_s p) i_{sq} + \frac{\omega^* M}{L_r} \psi_{rd} + \frac{M}{L_r} p \psi_{rq} \quad (8-20)$$

ベクトル制御が理想的であれば、 i_{sd}^* を一定に制御する場合

$$\psi_{rd} = M i_{sd}, \psi_{rq} = 0 \quad (8-21)$$

であるから、これを上式に代入して

$$e_{sd} = (R_s + \sigma L_s p) i_{sd} - \omega^* \sigma L_s i_{sq} \quad (8-22)$$

$$e_{sq} = \omega^* \sigma L_s i_{sd} + (R_s + \sigma L_s p) i_{sq} + \frac{\omega^* M}{L_r} \psi_{rd} = (R_s + \sigma L_s p) i_{sq} + \omega^* L_s i_{sd} \quad (8-23)$$

よって、図 8-13 の非干渉制御を行うと $e_{sd}^* = e_{sd}$, $e_{sq}^* = e_{sq}$ であれば、次式が成り立つ。

$$e_{sdc} = (R_s + \sigma L_s p) i_{sd} \quad (8-24)$$

$$e_{sqc} = (R_s + \sigma L_s p) i_{sq} \quad (8-25)$$

e_{sdc} , e_{sqc} から見ると、それぞれ d および q 軸電流のみが関係するので、電流制御が容易になる。実際の電流の代わりに電流指令を用いることも可能である。同期機についても、Park の式から同様に導ける。

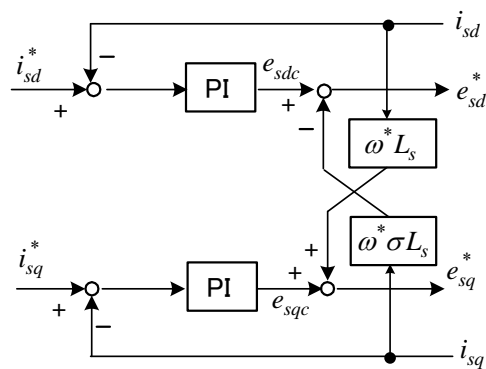


図 8-13 誘導機の電流制御
(非干渉制御付き)

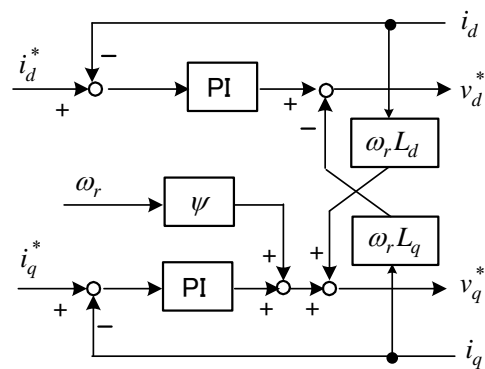


図 8-14 永久磁石同期機の電流制御
(非干渉制御付き)