ソフトウェアサーボシステムにおけるモータの 運動制御アルゴリズム (第1報)*

—DD方式インデックステーブルのカム曲線制御—

古 屋 信 幸** 松 原 正 知***

An Algorithm of Motor Control by Software Servo System (1st Report)
—Motion Control of DD Indexing Table by Cam Curve—

Nobuyuki FURUYA and Masatomo MATSUBARA

In servo motor control, it is important to select suitable acceleration and deceleration curve to improve dynamic characteristics. So many kinds of motion curves have been proposed. But it is not so easy to control the motor motion by such motion curves. In this report, a new algorithm in the software servo system to generate any motion curve is proposed. In conventional pulse train motor control system, it is relatively easy to control the motor by any motion curve, using DDA (Digital Differential Analyzer) method, but it can not be applied to the software servo system directly, because the system is essentially different. The proposed new algorithm can produce any motion curve, based on the cam curve motion, and less deviation which is caused by numerical integral calculation. The algorithm is the same in the function as the DDA method in conventional pulse train motor control system. The algorithm is able to be applied to multi axes robot control, but in the first report, it is applied to the motion control of 1-axis DD (Direct Drive) Indexing table and some experimental results are reported.

Key words: motion control, servo motor, software servo system, DD motor, indexing table

1. は じ め に

ロボットあるいは一般の駆動機構において、モータの運動曲線が残留振動等の動特性に大きな影響を及ぼすことは明らかであり、そのために数々の運動曲線が提案されている。ロボットの制御の場合、運動曲線を初期位置、持上げ位置、降下位置、目標位置で区分し、それらを3次式で補間する方法¹⁾、カム曲線に基づく運動曲線を与える方法²⁾等がある。また、DCモータの伝達関数を補償して出力曲線の特性を改善する方法も提案されている³⁾。

このように、適切な運動曲線に基づくモータの運動制御が、機械システムの運動特性の改善に重要であるにもかかわらず、実際のロボットに使用されているサーボモータの制御では、せいぜい台形状の加減速制御が行われているに過ぎない。第1の理由としては任

意のパターンの加減速制御を行う際に比較的複雑なアルゴリズムが必要となり、その結果として処理プログラムが長くなることがある。もう一つの理由としては、ソフトウェアサーボシステムでは従来のパルス列による加減速制御のアルゴリズムを、そのままの形で直接利用できないことが挙げられる。

従来のパルス列によるハードウェア中心のサーボシステムではパルス列の周期を加減速曲線に従って変えることで比較的簡単に加減速制御を実現することができた。複数の制御軸がある場合でも、DDA (Digital Differential Analyzer) 等を使って各軸間を同期させながら加減速制御することが可能であった。それに対し、最近のロボットの制御の主流であるソフトウェアサーボ方式においては、パルス列を介在させずに、直接、位置や速度を制御するので、パルス列方式で確立されてきた方式は、そのままの形では利用できない。

本研究においては、ソフトウェアサーボ方式の制御の下で、従来のパルス列方式と同じようにモータの運動を任意の運動曲線に従って制御するアルゴリズムについて検討を行った。このアルゴリズムは複数の自由

* 原稿受付 平成4年2月3日

** 正 会 員 山梨大学工学部 (甲府市武田4-3-11)

*** 学生会員 山梨大学大学院 (現、ミノルタ(株): 豊川市光明町2-41)

Fig. 1 Motion pattern generated by cam curve

度を有するロボットの制御に拡張することもできるが、本報告ではダイレクトドライブモータ（以下DDモータと略す）を使用した1軸のインデックステーブルの制御に適用したので、その結果について報告する。

2. 制御アルゴリズム

2.1 カム曲線に基づく加減速制御の考え方

数多くの種類のカム曲線が提案されているが、そのなかの多くは、統一的な式で表すことができる⁴⁾。その式によると、図1に示すように変位曲線、速度曲線、加速度曲線が得られる。ここでは速度制御型のソフトウェアサーボを前提として、速度曲線からデータを作成する。速度曲線は出力するサンプリングタイムで最初に離散化してデータテーブル化しておき、モータ駆動時にDAコンバータから出力する。出力する速度指令はサンプリング周期間で一定値となり、図2に示すように、0次ホールド回路を介して出力することに等価となる。最小のサンプリング時間は計算機の処理速度と、処理手続きの長さによって決まるが、その値より長くする場合には、計算のループに無駄時間のループを挿入すればよい。カム曲線を用いた運動制御を行うに当たって、まず、一定の回転角を基準値に設定しておき、これを基準ストロークと呼ぶ。基準ス

Fig. 2 Digitized velocity pattern

トロークに対して、割り付けるカム曲線と、その曲線の実際の最大速度と割り付時間を決定すれば、基準となる速度指令パターンが決まるが、データ化しておくのはこの基準ストロークの場合のデータだけとする。

2.2 駆動ストロークの変化に対する処理

1種類の基準となる加減速パターンだけでストローク（この場合は回転角）の変更に対処するために次のような条件を設定する。まず、基準ストロークよりも短いストロークでは最大速度をストロークに比例して下げ、出力の時間は一定とする。従って基準ストロークより短いストロークであれば、ストロークの長さによらずに割出し時間は一定となる。基準ストロークよりも長いストロークでは加速と減速の区間の中間に等速区間を設けて対処する。割出し時間を変更する場合には、テーブルを作成する段階で分割数と、各分割区間での出力データを変更すればよい。

図3において S_L を基準ストロークとして、それよりも短いストロークを S_s で表す。また、それぞれのストロークに対応する速度を V_L , V_s として、その最大値を V_{Lmax} , V_{smax} とすると、ストロークの長さに比例した速度指令は式(1)により求められる。図4に示すように、プログラムのループ回数 i での速度データを $V(i)$ 、プログラム1回のループ時間、すなわちサンプリング時間を Δt とすると i 回目のプログラムループまでに進んだ距離（回転角）は、式(2)で与えられる。この式から第 i ループ時の位置が算出できるので、モータのエンコーダからのフィードバックパルスのカウント値との偏差を計算し、ゲイン定数を乗じて速度指令に加算することにより位置フィードバックも構成することができる。しかし、実際の速度指令値はDAコンバータの分解能により、また、ストロークは位置検出のためのエンコーダの分解能によって量子化されるために、速度および位置指令値はそれぞれの分解能を最小単位とする整数値となる。従って、ここではこれらの比例演算を整数の演算のみ

Fig. 3 Displacement and velocity patterns for small stroke

Fig. 4 Digitized velocity pattern by sampling time

ここで、 S_r ：位置決め時の累積位置誤差、 $[]$ ：切捨て、 $R(i)$ ： i 回目の演算の剰余。

このようなアルゴリズムに従って速度指令値を生成すると、モータ駆動時の物理的外乱を無視すれば、数パルス分の移動距離でも速度指令の出力終了時には、目標移動距離との位置誤差は ± 1 パルス以内に収まる。速度指令の誤差も、DACの分解能の ± 1 ビット以内に収まることが確かめられる。実際のサンプリング時間の精度も制御の精

で実行するアルゴリズムを提案する。

$$V_s(i) = \frac{S_s V_L(i)}{S_L} \quad (1)$$

$$S(i) = \sum_{k=1}^{k=i} V(k) \Delta t \quad (2)$$

式(1)をそのまま実行した場合、実際の速度指令出力は式(3)のようになる。速度指令の出力のプログラムのループ回数を n とすれば、出力終了時には式(3)で切り捨てられた剰余 $R(i)$ による誤差が累積し、位置決め制御に入るときに式(4)で表される位置誤差を生ずる。そこで、式(3)によって生じた剰余を式(5)のように、次のループの速度データに加えてゆくことで、剰余の切捨て誤差の累積を防ぐことができる。以上のアルゴリズムに基づいて作られた制御プログラムのフローチャートを図5に示す。

$$V_s(i) = \left[\frac{S_s V_L(i)}{S_L} \right] \quad (3)$$

$$S_r = \sum_{i=1}^n R(i) \Delta t \quad (4)$$

$$V_s(i+1) = \left[\frac{S_s V_L(i+1) + R(i)}{S_L} \right] \quad (5)$$

Fig. 5 Flowchart of motion control in software servo system

Fig. 6 Control system of DD indexing table

度に影響を及ぼすと考えられるが、今回はプログラムを実際に行い、実験的にサンプリング時間を測定して計算に用いた。

3. DD モータ制御システム

本研究では THK 社製のインデックステーブルを用いた。従来のインデックステーブルは、カム等の純機械的な機構によって構成されていたが、低速でも高トルクの得られる DD モータが開発されたことによりサーボ制御の可能なインデックステーブルが実現できるようになった。このようなインデックステーブルは制御システムとしてはサーボモータと等価である。

今回の研究で使用した装置を図 6 に示す。DD モータの制御には、CPU 80386 (16/20 MHz) を使用したパーソナルコンピュータ (PC-9801 RL) でソフトウェアサーボシステムを構築し、システムバスに増設した 12 bit DA コンバータから、DD モータドライバに直接速度指令を入力する。また、モータの位置決め用エンコーダからのパルス列をモータドライバを介して、増設した 24 bit カウンタに入力し、そこからコンピュータに取り込んで位置フィードバックを行っている。

速度フィードバックループはモータドライバ内でハードウェア的に構成されており、その点では完全なソフトウェアサーボシステムとはいえず、セミソフトウェアサーボ方式である。テーブルの位置決めのためのエンコーダの分解能は 1 回転が 655 360 パルスで、最高回転速度は 2 rps である。モータドライバから速度信号も取り込んでいるが、今回は制御に利用していない。制御プログラムの記述には C 言語 (TURBO-C) を使用したので、1 回の速度指令出力のプログラ

Fig. 7 Experimental system

ムループには、ほぼ 1 ms を要した。駆動プログラムの一部を C 言語に代えてアセンブリ言語を使用すれば、さらに高速な処理が可能になると考えられる。

4. 実験結果

以上の制御アルゴリズムに従って実際にインデックステーブルを駆動する実験を行い、目的どおりの結果が得られているか実験による検証を行った。測定および評価は DD モータからのフィードバックパルス、コンピュータに取り込み回転速度に換算し、加減速制御に用いたカム曲線から計算される指令速度入力との偏差を計算することによって行った。また、テーブル上には加速度計を取り付けて加速度の実測も行った。

図 7 に実験システムの写真を示す。負荷としてはテーブル上に腕とおもりを取り付けた。おもりを交換

(a) Modified trapezoid

(b) Modified sine

Fig. 8 Input velocity curve and output velocity curve of DD motor indexing table

(a) Modified trapezoid

(b) Modified sine

Fig. 9 Deviation of input and output velocity

ることにより、負荷の変動に対して、どの程度まで入力曲線に追従が可能であるかも検討した。

図8は実際に加減速に用いた曲線の指令値とフィードバックパルスから計算されたDDモータの速度出力を示したものである。負荷の大きさはモータ自体の慣性モーメントにテーブルの慣性モーメント、アームおよびおりの慣性モーメントをすべて加えて、モータ単体の慣性モーメントを基準として表し、4.05倍まで変えて実験した。図8は負荷が4.05倍の時の結果である。回転角度は 180° で割出し時間は0.5s、出力のサンプリング時間はほぼ1msである。

図9は動作時間中の速度指令とモータの回転速度の偏差を、最大出力速度に対する割合で示したものである。(a)は変形台形曲線、(b)は変形正弦曲線で駆動

した場合の結果である。どちらの結果も最大速度に対して、1%から2%の誤差の間で追従していることがわかる。また、速度偏差が大きく生じるところは、加速を開始した直後などの加速度の変化率の大きい時点で発生する傾向にある。誤差の大きさと傾向は、負荷を変えて繰返し実験しても変化が見られなかったことから、例えば摩擦や制御システムの遅れ等の負荷変動の影響を受けない要因により生ずると考えられる。

図10(a), (b)はインデックステーブル上に取り付けた加速度計による加速度の測定結果である。加速度計の取付位置はテーブル中心から108mmの位置である。また、出力は100Hzのローパスフィルタを介して、記録計に記録している。(a)は変形台形曲線を用いた場合、(b)は変形正弦曲線を用いてモータを駆動

Fig. 10 Comparison between measured and calculated acceleration

した場合の結果である。それぞれの図の下に示した(c), (d)の図はカム曲線から計算によって求めた加速度である。両者を比較することにより、目的とする入力曲線に対して、よく追従していることがわかる。

5. 考 察 と 結 論

DD モータを用いたインデックステーブルのソフトウェアサーボシステムの基での動作制御のアルゴリズムを提案し、従来のカムを用いたインデックス装置と同等のカム曲線に従った運動制御の可能なことが確認できた。この結果、DD モータを使用したインデックステーブルにおいて、従来使用されてきたカム式のインデックス装置と同等の運動特性を実現できた。

本研究で用いたシステムで生じた制御誤差は負荷の変動による影響があまり見られなかったことから摩擦等により生ずると考えられるが、これらの影響はフィードフォワード制御的な補償等により、改善できると考えられる。

ソフトウェアサーボシステムが広く普及しているロ

ボットの制御システムにおいても、ここで述べた制御アルゴリズムを応用することは可能であり、今後はロボットのようにさらに制御軸数の多いサーボシステムにおける応用を検討する。

終わりに、本実験システムをご提供いただいたTHK(株)および、数々の便宜を図っていただいた沖電気工業(株)に謝意を表します。

参 考 文 献

- 1) R. Paul : Modeling, Trajectory Calculation, and Servoing of a Computer Controlled Arm, Memo AIM-177; Stanford Artificial Intelligence Laboratory, Palo Alto, Calif., (1972).
- 2) 古屋信幸, 相馬勝男, 陳 栄進, 牧野 洋: 選択的コンプライアンス構造をもつ組立用ロボットの開発研究(第2報)——SCARA 制御装置のハードウェアとソフトウェア, 精密機械, 49, 7 (1983) 835.
- 3) 神崎一男, 小林伸明: 直流サーボモータによるカム曲線の出力(第1報)——逆伝達関数補償法の適用, 精密工学会誌, 55, 12 (1989) 2215.
- 4) 牧野 洋: ユニバーサルカム曲線とその応用, 山梨大学工学部研究報告, 28 (1978) 48.