

ロボットアームのダイレクト駆動における学習制御について

小山 純*・樋口 剛*
小杉 辰彦**・山田 英二***

Learning Control Method for Direct Drive of Robot Arm

by

Jun OYAMA* · Tsuyoshi HIGUCHI*
Tatsuhiko KOSUGI** and Eiji YAMADA***

Control method of direct-drive arm has been studied to realize the high speed and high accuracy position control. It is one of the important problem to reduce the speed fluctuation caused by the arm weight. In this paper, a learning control method based on the periodicity of the robot arm action is proposed to drive the motor with constant speed. Availability of this control method is conformed by experiment.

1. まえがき

ロボットにおける高速・高精度の位置決めや力の制御を実現するため、ダイレクトドライブ (D・D) 方式の開発が進められている。

従来のロボットアームでは、減速伝達機構が採用されているため、(a) ギアによる摩擦トルクが大きい、(b) アーム剛性が低く振動の原因となる、(c) ロータが高速回転するため慣性が大きくなり応答性が低くなる、などの欠点を持っていた。この欠点を克服するために開発が進められている D・D アームは高速・高精度な動作を可能にするが、アーム重量による非線形力の影響が顕著となるという問題が生ずる。従ってこのような変動負荷時の速度または位置制御に、従来から用いられている PID 制御を適用しても、かなりの速度変動が存在することになる。

本論文では、周期変動負荷をもつ電動機の定速度運転を実現するため、新しい“学習制御法”を提案し、実験による検証を行っている。

2. 学習制御法

作業ロボットは一般に、Fig. 1 に示すようにアー

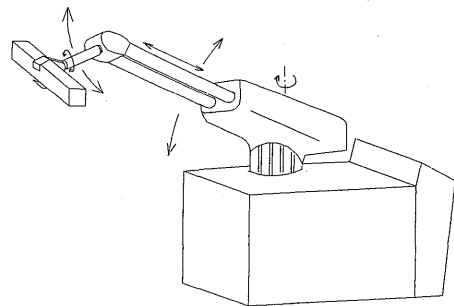


Fig. 1 Robot arm.

ムの先端に負荷となる物を掴み種々の運動を行う。このとき、掴む物体によってアームに加わる重力やアームの慣性モーメントは変化する。また、アームの位置によっても電動機が出力すべきトルクは一定でない。このようにシステムの特性は常に変化することになる。従って、このようなアームを D・D 駆動法で定速運転する場合においては、一般に用いられている PID 制御では精度が悪い。本研究では、モータに周期変動負荷が取り付けられた場合の制御についてトルク補償、学習補償を持つ制御方式を考案し実験を行った。

昭和62年9月30日受理

*電気工学科 (Department of Electrical Engineering)

**電気工学専攻 (Graduate Student, Electrical Engineering)

***電子工学科 (Department of Electronic Engineering)

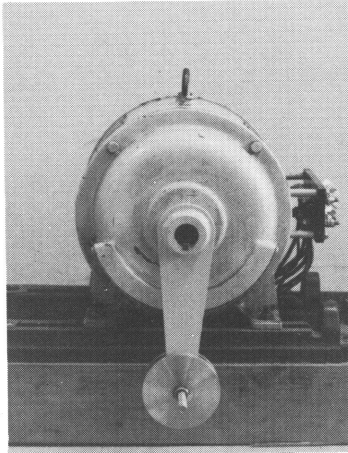


Fig. 2 Experiment device.

Fig. 2 に実験装置の外観を示す。出力 2 kw の直流電動機に、先端に 1 kg の重りを持つ長さ 20 cm のアームを取り付けている。このとき、アームの重心は、モータの軸と一致していないため、アームの位置によって、モーターが出力すべきトルクは不均一となる。

Fig. 3 における運動方程式は次のようになる。

$$J \frac{d^2\theta}{dt^2} = K_t I_t - mgl \sin\theta \quad (1)$$

J : 慣性モーメント

θ : アームの位置

$K_t I_t$: モータの出力トルク

m : アームの重量

l : モータ軸から、アームの重心までの距離

右辺第 2 項が、アームによる変動負荷の影響である。

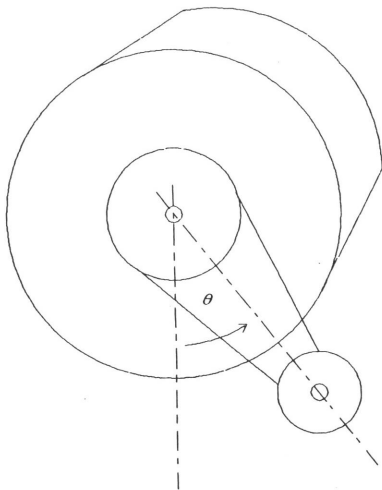


Fig. 3 Analytical model.

2. 1 PID 制御

PID 動作は、次式で与えられる。

$$y = -k_p \left(z + \frac{1}{T_i} \int z dt + T_d \frac{dz}{dt} \right) \quad (2)$$

ただし、 y : 操作量

Z : 偏差信号

第一項は比例 (proportional) 動作、第二期は積分 (integral) 動作、第三項は微分 (derivative) 動作である。フィードバック制御の出発点として P 動作を考える。これは、制御量が目標値からずれていれば、そのずれ (制御偏差) に比例した操作量を制御対象に加える制御動作である。しかし、P 動作では、定位置的な制御対象の場合、目標値のステップ状変化に対する制御量の変化には Fig. 4 に見るようにオフセット (定常位置偏差) が残ってしまう。そこでオフセットをなくすために I 動作を導入して PI 動作を用いる。ところが、PI 動作にするとオフセットはなくなるが、立ち上がり時間が長くなって、速応性が損なわれる。そこで速応性を向上するために PID 動作を用いる。

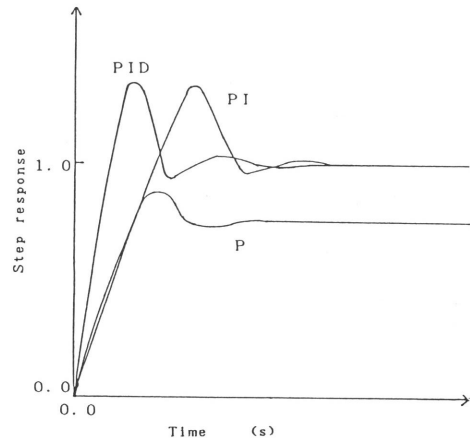


Fig. 4 Step response.

ここで、(2)式をデジタル化すると以下の式を得る。

$$I_{pid}(t) = I_{pid}(t-1) + K_i (S_t - Z_{t-1}) - K_p (Z_t - Z_{t-1}) + K_f (S_t - S_{t-1}) - K_d \{ (Z_t - Z_{t-1}) - (Z_{t-1} - Z_{t-2}) \} + K_s \{ (S_t - S_{t-1}) - (S_{t-1} - S_{t-2}) \} \quad (3)$$

ただし K_p : フィードバック比例定数

K_f : フィードフォワード比例定数

K_d : フィードバック微分定数

- K_s : フィードフォワード微分
定数
- K_i : 積分定数
- I_{pid} : 操作量
- Z_t : 実測値
- S_t : 指令値

まれている部分がマイクロ CPU によってデジタル的に行われる。CPU には演算の他に、記憶を行う機能があるため、これを利用して、適当な補償を行うことが考えられる。(1)式では、定常負荷は存在せず、変動負荷のみであるため、従来のような PID 制御では、速度変動が大きく実用的でない。従って、ここでは、トルク補償及び学習補償を追加した制御方法を考える。

Fig. 5 に PID 制御のブロック図を示す。点線で囲

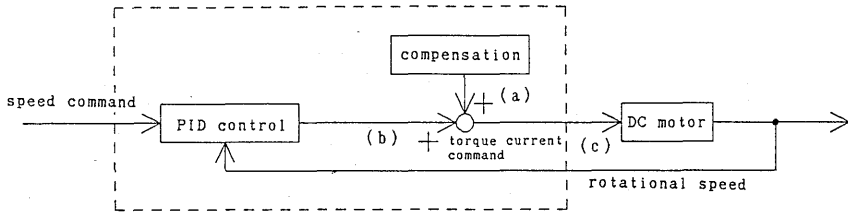


Fig. 5 Block diagram of PID control.

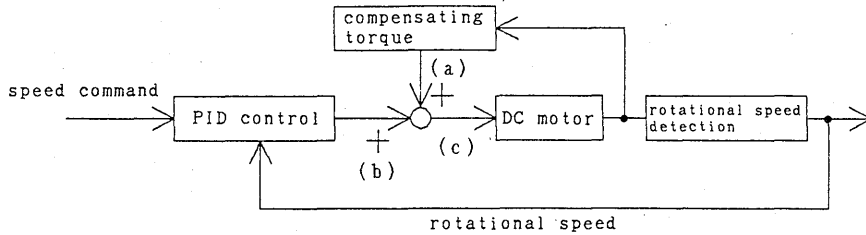


Fig. 6 Block diagram of compensating torque.

2.2 トルク補償

式1の右辺第2項を考慮したトルク電流を流すことにより、変動負荷の影響を軽減することが可能となる。これがトルク補償と呼ばれるものである。このブロック図を Fig. 6 に示す。 I_t を I_{t1} と I_{t2} に分け、

$$J \frac{d^2\theta}{dt^2} = K_t (I_{t1} + I_{t2} \sin\theta) - mgl \sin\theta$$

$$= K_t I_{t1} + (K_t I_{t2} - mgl) \sin\theta \quad (4)$$

$K_t I_{t2} = mgl$ となるような振幅を選び、 $I_{t2} \sin\theta$ なるトルク補償を行うことにより、右辺第2項の影響を

取り除くことができる。 I_{t1} は、PID 制御の操作量であり、 $K_t I_{t2}$ は実験的に決定する必要がある。

2.3 学習補償

一般に、ロボットアームは繰り返し動作を行うことが多い。この周期性を利用して前周期の操作量を記憶し、次の周期の制御を行うときの補正值として使うと周期を繰り返す毎に制御精度が向上する。本稿ではこの方法を学習制御法と呼んでいる。ブロック図を Fig. 7 に示す。

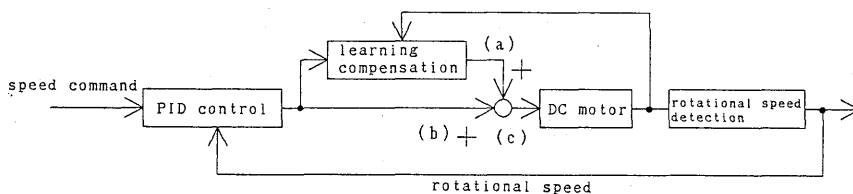


Fig. 7 Block diagram of learning compensation.

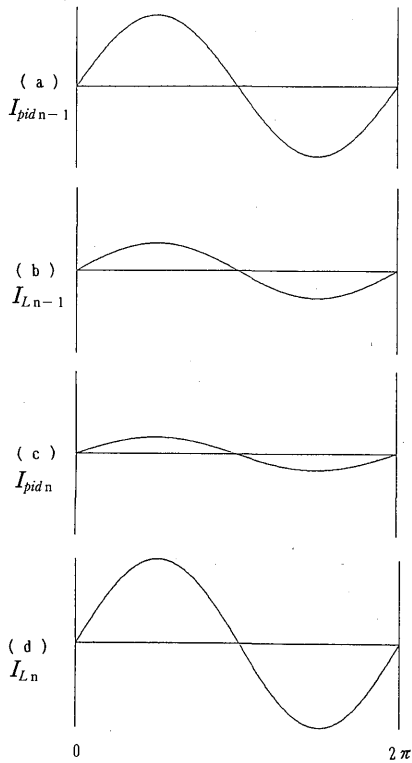


Fig. 8 Explanation of learning compensation.
 (a) PID control (b) Learning compensation
 (c) PID control
 (d) Learning compensation

$$I_{L_n}(\theta) = I_{L_{n-1}}(\theta) + I_{pid_{n-1}}(\theta) \times K_I$$

学習補償は、マイクロ CPU のメモリを利用して、操作量を記憶する方法である。ここで記憶の方法に時間の関数として記憶する方法と、位置の関数として記憶する方法の2通りがある。負荷変動は位置の関数であるため位置の関数として記憶を行っている。従ってトルク電流指令は、PID 制御の操作量である I_{pid} と学習補償の操作量である $I_{L\theta}$ の和となる。

学習制御の説明を Fig. 8 に示す。n-1 周期目の PID 制御の指令値 (図中 a) が学習補償量に加えられ n 周期目の学習補償量となる。ここで K_L が掛けられるが、これは学習における重みである。

ところで、PID 制御の操作量は、時間遅れをもっているが、この問題も学習補償により解決する。遅れ時間を、 Δt 、位置を $\Delta\theta$ 、角速度を ω とすると、

$$\Delta\theta = \omega\Delta t \tag{5}$$

なる関係がある。これを考慮した式は、

$$I_{L_n}(\theta) = I_{L_{n-1}}(\theta) + I_{pid_{n-1}}(\theta + \Delta\theta) \times K_I \tag{6}$$

となる。

3. 実験装置

Fig. 9 にシステム構成図を示す。本システムは、4 象限チョップにより駆動される直流モータ (Fig. 2) と、16 bit CPU を用いたゲート制御回路から構成

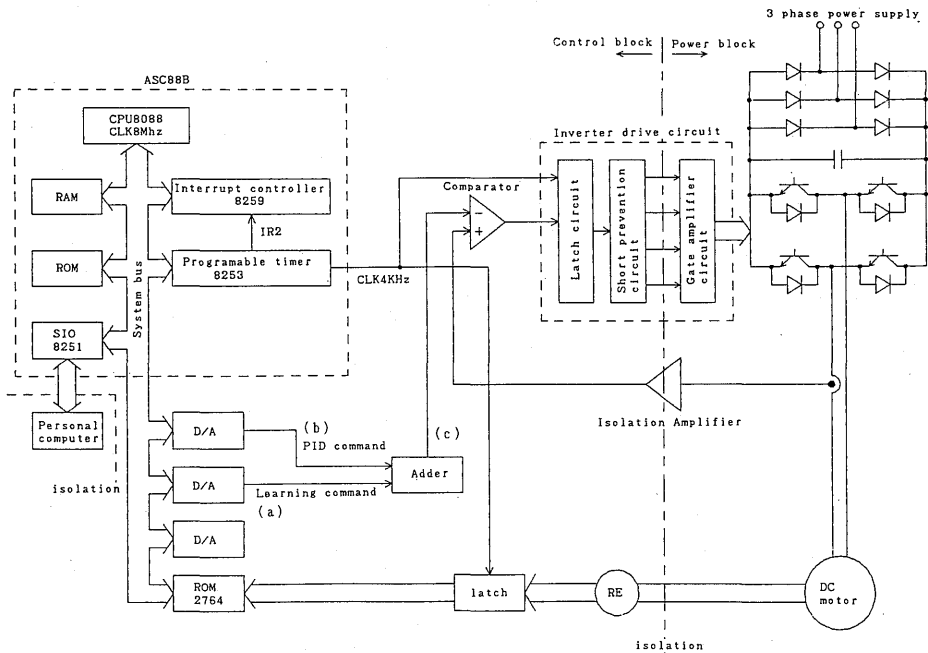


Fig. 9 System configuration.

される。ここでは、電流ベクトル制御としての回路構成及び信号の流れを説明する。

CPU は電動機の変圧子に接続された12 bit アブソリュート型ロータリエンコーダにより500ms毎に回転

子位置を検出し、同時に回転速度を計算する。計算した回転速度は指令速度と比較され、デジタル的にPID 制御が行われトルク電流指令が演算される。トルク電流指令は学習指令 (図中 a) と PID 指令 (図

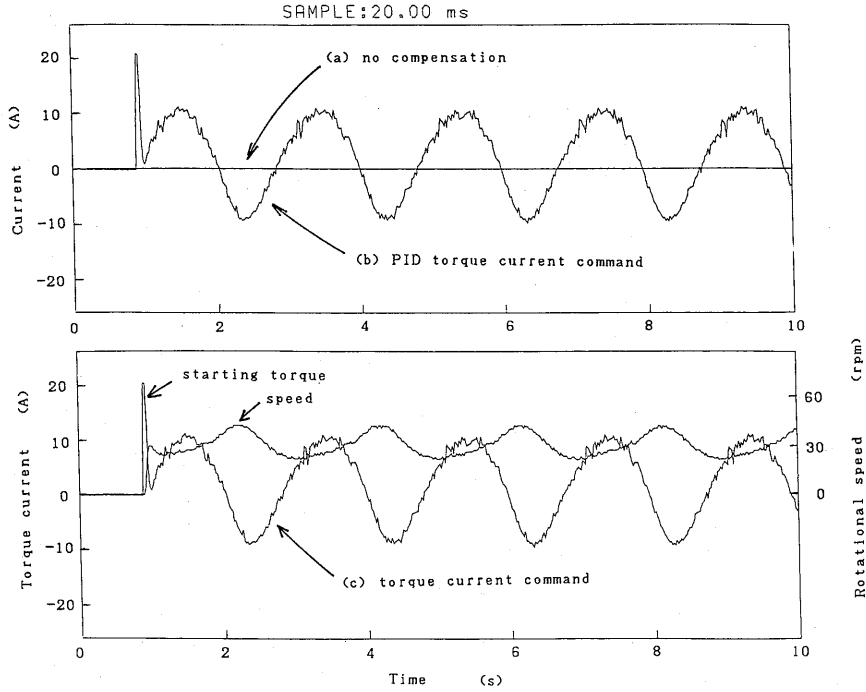


Fig. 10 Oscirogram of PID control.

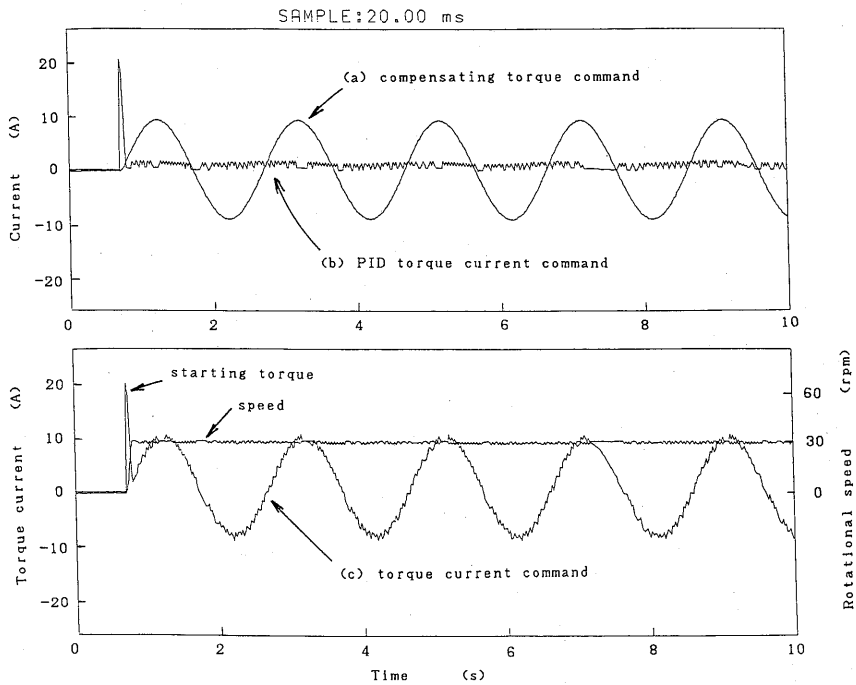


Fig. 11 Oscirogram of compensating torque.

中b)に分けて D/A コンバータから出力され OP アンプにより加算される。

4. 実験結果

Fig. 10に PID 制御のみの場合に速度指令を 0 から 30 rpm に変化したときのオシログラムを示す。回転指令 30 rpm に対し ± 9 rpm の速度変動がある。

Fig. 11 は PID 制御に加えトルク補償を行った場合である。速度変動は ± 1 rpm と大幅に改善されることがわかる。しかし、トルク補償では、正弦波の振幅と位相を決定する必要がある上、実際にアームが使われるときは、物体を持っている時と持っていない時とで重量が変化するため実用性に乏しいという欠点がある。また、システムの変化は全てソフトウェアで補償しなければならず多関節ロボットへの応用は難しい。

Fig. 12に、学習制御を行った場合を示す。始動から最初の 2 秒は、始動トルク (始動時のスパイク状の

トルク)の影響を受けないように学習を禁止している。それ以降は学習補償となっているが、1 周期目は、学習のみであり出力は現れていない。従って、最初の 4 秒では、PID 制御のみであり、 ± 9 rpm の速度変動がある。これに対し学習の後は ± 1 rpm と速度変動が大幅に改善される。本補償は、トルク補償のように煩雑でなく、しかも十分な精度が得られることが明らかとなった。

5. むすび

学習補償を利用した周期変動負荷時の制御について一定速度でモータを回転させる実験を行い良好な結果を得た。今後、提案した学習補償が周期性を持つ制御に汎用的に利用できるか検討するため、任意の 2 地点を往復するような位置制御に対して、実験による検証を行う予定である。

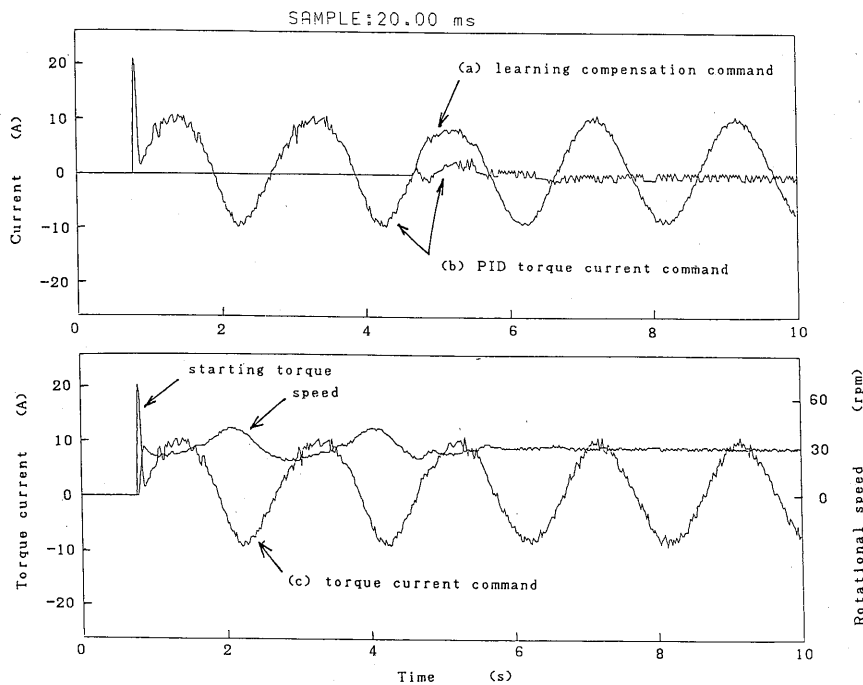


Fig. 12 Oscirogram of learning compensation.