2-Delay 入力制御方式を用いた離散時間学習制御法による マニピュレータの軌道制御 — 実機による検証*

早川聡一郎**・鈴木 達也**・大熊 繁**

Trajectory Control of Manipulators Based on Discrete Time Learning Control with 2-Delay Input Control Technique — Experimental Results*

Soichiro HAYAKAWA**, Tatsuya SUZUKI** and Shigeru OKUMA**

A learning control based on repetitive operations of robotic manipulators is one of the most promising method to realize high speed and high precision control for robotic manipulators. The joint servo system usually has an unstable zero in discrete-time domain. This fact leads to unstable learning, when a sampling period is small. In previous paper, we have proposed a new learning algorithm based on so-called 2-delay input method. By using the 2-delay input method we can stabilize the unstable zero and realize the stable learning even if the sampling period is small. However, concrete algorithm for parameter design and experimental results have not been shown yet. In this paper, we propose new design algorithm of two parameters in 2-delay input system especially to suppress ripple of the trajectory and show some experimental results to verify effectiveness of the proposed method.

1. はじめに

産業用ロボットの高速高精度化を実現する手法の一つとして、繰り返し動作に基づく学習制御法がある^{1)~4)}。学習制御法は時間の関数である過去の試行結果を離散化して、メモリに記憶し次の制御に用いる。したがって、その安定性を議論する際にはサンプル周期との関係に注目しながら、離散時間系特有の性質を考慮に入れて議論する必要がある。通常、マニピュレータのダイナミクスは遠心力、コリオリ力等の非線形項を有するが、実用化されているマニピュレータのほとんどは50~100程度の減速比を持つ減速機を有しており、したがって、各関節ごとに非干渉化された線形システムとみなしてもほとんど問題がない。線形システムを対象とする場合、離散時間システムの特徴の一つとして、短いサンプル周期で離

散化した際にパルス伝達関数が不安定零点を有することが挙げられる⁸⁾。一般に、サンプル周期を長くすることで不安定零点問題は解決されるが、サンプル点間の誤差は広がる。したがって高精度な動作を実現するためにはサンプル周期は短い方が望ましい。著者らは、これまでマニピュレータの各関節のサーボシステムを離散時間線形システムとみなし、そのシステムが不安定零点を有する場合、学習が不安定になりロボットの軌道制御に適用する際に大きな問題となることを理論・実験の両面から指摘してきた^{5), 6)}。またこの問題に対する有効な解決法として、線形近似システムに基づいた2-delay 入力制御方式⁹⁾を用いた学習制御系を提案した⁷⁾。2-delay 入力制御方式を用いることにより、制御対象が2入力1出力となるため、学習オペレータの選び方によって、不安定零点問題を回避できるが、参考文献7)ではシミュレーション結果が示されるにとどまっている。

本論文では、まず第一に著者らが参考文献7)において提案した2-delay 入力制御方式を用いた学習制御系が、不安定零点を安定化し、短いサンプル周期でも安定な学習を実現できることを実機により検証する。またそ

* 原稿受付 1994年10月6日

** 名古屋大学 工学部 School of Engineering, Nagoya University; Furocho, Chikusa ward, Nagoya city, Aichi 464-01, JAPAN

Key Words: discrete time learning control, unstable zero, 2-delay input control technique, robotic manipulator, ripple.

の際に、参考文献7)では議論されていない学習オペレータの設定とサンプル点間でのリップルの関係を調べ、リップルの抑制が可能な学習オペレータの設計法を実験を通して明らかにする。

以下、2.で2-delay 入力制御方式とその学習制御系の安定条件について述べ、3.では2-delay 入力制御方式を用いた学習制御系の構成法について述べる。4.では学習オペレータの設計法について述べ、5.では学習オペレータのパラメータの設計指針、特にさらにリップルを抑制するためのパラメータの設計指針を数値解析例をもとに示す。6.で実験結果を示し、2-delay 入力制御方式を用いた学習制御系が不安定零点問題に対して有効に働くことを示す。また5.で示した解析結果を元にパラメータを設定した場合の実験結果を示し、リップルの抑制効果を確認する。

2. 2-Delay 入力制御方式と学習の安定条件

2.1 2-Delay 入力制御方式

2-delay 入力制御方式は Fig. 1 のようにサンプル点から mT ($0 < m < 1$) だけ遅れた時点において入力を変更する制御方式をとる。もとのシステムを1入出力系と仮定すると、2-delay 入力制御方式を用いた制御系の状態方程式は以下のように求められる。

$$\mathbf{x}(n+1) = \mathbf{A}\mathbf{x}(n) + \mathbf{b}_1 u_1(n) + \mathbf{b}_2 u_2(n) \quad (1)$$

$$y(n) = \mathbf{c}\mathbf{x}(n) \quad (2)$$

ただし

$$\mathbf{A} = e^{\mathbf{A}_c mT} \quad (3)$$

$$\mathbf{b}_1 = \int_{(1-m)T}^T e^{\mathbf{A}_c t} \mathbf{b}_c dt, \quad \mathbf{b}_2 = \int_0^{(1-m)T} e^{\mathbf{A}_c t} \mathbf{b}_c dt \quad (4)$$

$$\mathbf{c} = (1 \ 0 \ 0) \quad (5)$$

である。ここで $\mathbf{A}_c$, $\mathbf{b}_c$ は、連続系での制御対象のシステム行列である。 $u_1(n)$, $u_2(n)$ に関しては Fig. 1 を参照されたい。

$u_1(n)$, $u_2(n)$ を独立した入力とみなすことによって、システムは2入力1出力系とみなせる。前半部の入力

Fig. 1 2-delay input system

$u_1(n)$ から出力 $y(n)$ への伝達関数を G_1 , 後半部の入力 $u_2(n)$ から出力 $y(n)$ への伝達関数を G_2 とすると、それらはそれぞれ以下のように与えられる。

$$G_1(z) = \mathbf{c}(z\mathbf{I} - \mathbf{A})^{-1}\mathbf{b}_1 \quad (6)$$

$$G_2(z) = \mathbf{c}(z\mathbf{I} - \mathbf{A})^{-1}\mathbf{b}_2 \quad (7)$$

この二つの伝達関数はほとんど全ての m に対して無限遠点以外に共通な不安定零点を持たないことが参考文献9)で証明されている。

2.2 2-Delay 入力制御方式を用いた学習制御系の安定条件

2-delay 入力制御方式を用いることから、本論文では Fig. 2 で示すような2入力1出力の制御対象を考える。

Fig. 2 Two inputs single output system

$$Y_k(z) = \mathbf{G}(z) \mathbf{U}_k(z) = (G_1(z) \ G_2(z)) \begin{pmatrix} U_{1,k}(z) \\ U_{2,k}(z) \end{pmatrix} \quad (8)$$

ただし、 $\mathbf{G}(z)$ は制御対象のパルス伝達関数を表わし、 $U_{1,k}(z)$, $U_{2,k}(z)$ はそれぞれ二つの入力の z 変換を表わす。また、 $Y_k(z)$ は出力の z 変換を表わす。このとき各試行における入力は次式によって修正される。

$$\mathbf{U}_{k+1}(z) = \mathbf{U}_k(z) + \mathbf{H}(z) \mathbf{E}_k(z) \quad (9)$$

ただし、

$$\mathbf{E}_k(z) = Y_d(z) - Y_k(z) \quad (10)$$

である。ここで、添え字 k , $k+1$ は試行回数を表わす。また $\mathbf{H}(z) = (H_1(z) \ H_2(z))^T$ はパルス誤差修正関数を表わし、 $H_1(z)$ が $U_{1,k}(z)$ の修正量を、また $H_2(z)$ が $U_{2,k}(z)$ の修正量を決定する。なお、添字 T は転置を表わす。今、(8), (9) 式から以下の式が与えられる。

$$\mathbf{E}_{k+1}(z) = (1 - \mathbf{G}(z) \mathbf{H}(z)) \mathbf{E}_k(z) = \mathbf{S}(z) \mathbf{E}_k(z) \quad (11)$$

(11) 式より誤差の L_2 ノルムが試行ごとに減っていくための条件を考えると、それは次式で与えられる。

$$\begin{aligned} \|\mathbf{S}\|_\infty &= \sup_{0 \leq \omega \leq \omega_N} |\mathbf{S}(e^{j\omega T})| \\ &= \sup_{0 \leq \omega \leq \omega_N} |1 - \mathbf{G}(e^{j\omega T}) \mathbf{H}(e^{j\omega T})| < 1 \quad (12) \end{aligned}$$

ただし $\|S\|_\infty$ は伝達関数の H^∞ ノルムを表わす。(12)式において $G(z)H(z)$ に不安定零点が存在すると安定条件が満たされなくなることは、複素関数論の最大絶対値の原理から容易に証明できる⁵⁾。

3. 2-Delay 入力制御方式を用いた学習制御系の構成

本論文で用いるロボットマニピュレータの各関節のデジタルサーボ系のブロック図を Fig. 3 に示す。Fig. 3 中の Motor and Robot Dynamics のブロックの詳細を Fig. 4 に示す。Fig. 4 において、 τ_L はアームの駆動トルクを表わす。ほかのパラメータについては Table 1 を参照されたい。Fig. 3 で示すように Fig. 4 のブロックに対して、次式で与えられる入力電圧値を印加するものとする。

Table 1 Constants of D.C.M.

Link No.	1	2
Inertia J_m [kgm ²]	1.980	0.618
Damping coefficient D_m [Nms/rad]	7.675	2.775
Elec. - mot. force constant K_E [Vs/rad]	10.602	10.220
Torque constant K_T [Nm/A]	10.150	10.392
Armature resistance R_a [Ω]	2.3	3.2
Armature inductance L_a [mH]	1.8	3.2
Gear ratio	50	50

Fig. 3 Block diagram of digital servo system

Fig. 4 Block diagram of motor and robot dynamics

$$v_a(n) = (K_p(\theta_d(n) - \theta(n)) + K_v(\dot{\theta}_d(n) - \dot{\theta}(n)) + \xi_1(n))K_{cu} \quad (0 \leq t < mT) \quad (13)$$

$$v_a(n) = (K_p(\theta_d(n) - \theta(n)) + K_v(\dot{\theta}_d(n) - \dot{\theta}(n)) + \xi_2(n))K_{cu} \quad (mT \leq t < T) \quad (14)$$

$$\left(K_{cu} = \frac{R_a}{K_T}\right)$$

ただし、 n はサンプル点を表わし、 $\xi_1(n)$ 、 $\xi_2(n)$ は学習によって獲得するサンプル点間における前半部と後半部のフィードフォワード補償入力である。ここで、(1)式における $u_1(n)$ 、 $u_2(n)$ を次のようにみなす。

$$u_1(n) = K_p\theta_d(n) + K_v\dot{\theta}_d(n) + \xi_1(n) = V(n) + \xi_1(n) \quad (0 \leq t < mT), \quad (15)$$

$$u_2(n) = K_p\theta_d(n) + K_v\dot{\theta}_d(n) + \xi_2(n) = V(n) + \xi_2(n) \quad (mT \leq t < T), \quad (16)$$

$$(V(n) = K_p\theta_d(n) + K_v\dot{\theta}_d(n))$$

このように変更されたシステムへの入力の関係図を Fig. 5 に示す。また、 $y(n)$ としてサンプル点上のモータの関節角 $\theta_m(n)$ を考える。このとき (9) 式の学習アルゴリズムは、次式で表わされる学習アルゴリズムと等価となる。

$$\Xi_{k+1}(z) = \Xi_k(z) + H(z)E_k(z) \quad (17)$$

ただし、 $\Xi(z)$ は $\xi(z) = (\xi_1(z) \ \xi_2(z))^T$ の z 変換である。これは (15)、(16) 式の第 1、2 項、すなわち $V(n)$ は試行によって変化せず、 $\xi_1(n)$ 、 $\xi_2(n)$ が等価的に前

節までの $u_1(n)$ 、 $u_2(n)$ として考えられるからである。これは Fig. 1 と Fig. 5 を比較することでも容易に確認できる。

また、(8) 式の $G(z)$ としては、フィードバックループにより安定化されたシステムを考え、そのパルス伝達関数は次式で与えられる。

$$G(z) = \begin{pmatrix} G_1(z) \\ G_2(z) \end{pmatrix}^T = \begin{pmatrix} c(zI - A_f)^{-1}b_1 \\ c(zI - A_f)^{-1}b_2 \end{pmatrix}^T \quad (18)$$

ただし、

$$A_f = A - b_f f \quad (19)$$

$$b_f = \int_0^T e^{A_d t} b_c dt \quad (20)$$

$$f = (K_p \ K_0 \ 0) K_{cu} \quad (21)$$

である。

以上より、2. で示した安定条件をもと

Fig. 5 Modified 2-delay input signal

に Fig. 3 のデジタルサーボ系に対して、 ξ を更新していく学習アルゴリズムを付加した場合の学習の安定性が議論できる。

4. 学習オペレータの設計法

学習オペレータの設計に際してまず学習の試行回数を減らすことを考える。学習の収束速度を早くするためには $S(z)$ のノルムを小さくすれば良いことから、(12) 式より $G(z)H(z) = 1$ となるように $H(z)$ を選べば良い。

$$G_1(z)H_1(z) + G_2(z)H_2(z) = 1 \quad (22)$$

しかし、実際には制御対象の正確な伝達関数 $G(z)$ は得られないことから、Fig. 4 の Motor and Robot Dynamics のブロックにおいて τ_L を無視し、それを Fig. 3 に代入したものを考え、以下、線形近似伝達関数 $G_M(z) (= (G_{M1} \ G_{M2}))$ と表記し、設計に使用することにする。産業用ロボットのほとんどは大きな減速比を持つ減速機を有していることから τ_L の影響を無視して考えても有効な解析結果が得られる。以上より、設計の際には (22) 式の代わりに次式を考える。

$$G_{M1}(z)H_1(z) + G_{M2}(z)H_2(z) = 1 \quad (23)$$

この (23) 式を解くことで試行回数の少ない学習オペレータを得ることができる。この式の解は無数にあるが、ここでは設計の簡便さと二つの入力の関係の明確化の観点から、二つの学習オペレータを独立に設計するのではなく、二つの入力の大きさの比を一定とした補償入力を生成する学習オペレータの設計法を考える。すなわち学習オペレータ $H(z)$ に対して以下のような拘束を考える。

$$H_1(z) = (1 - \alpha)H(z) \quad (24)$$

$$H_2(z) = \alpha H(z) \quad (25)$$

これより (17) 式は以下のように変更される。

$$\begin{pmatrix} E_{1,k+1}(z) \\ E_{2,k+1}(z) \end{pmatrix} = \begin{pmatrix} E_{1,k}(z) \\ E_{2,k}(z) \end{pmatrix} + \begin{pmatrix} (1 - \alpha)H(z) \\ \alpha H(z) \end{pmatrix} E_k(z) \quad (26)$$

ここで α は二つの入力の比を決めるパラメータであり、振幅比と呼ぶ。またこのとき、(23) 式は以下のように変更される。

$$(1 - \alpha)G_{M1}(z)H(z) + \alpha G_{M2}(z)H(z) = 1 \quad (27)$$

上式を $H(z)$ について解けば学習オペレータが得られる。

$$H(z) = ((1 - \alpha)G_{M1}(z) + \alpha G_{M2}(z))^{-1} \cong G_{sum}^{-1} \quad (28)$$

合成伝達関数 G_{sum} に不安定零点が存在しないようにパラメータ α と m を適切に定めることで安定な学習が実現できる。5. においてこの二つのパラメータの設計指針を示す。

5. 二つのパラメータの設計指針

5.1 伝達関数の零点の安定領域

合成伝達関数 $G_{sum}(z)$ は以下の式で表わされる。

$$G_{sum}(z) = (1 - \alpha)G_{M1}(z) + \alpha G_{M2}(z) \quad (29)$$

また $G_{M1}(z)$, $G_{M2}(z)$ の分子多項式を、

$$G_{M1n}(z) = a_1(z^2 + b_1z + c_1) \quad (30)$$

$$G_{M2n}(z) = a_2(z^2 + b_2z + c_2) \quad (31)$$

とする。以上より (29) 式の分子多項式 $G_{sumn}(z)$ は次のように求まる。

$$\begin{aligned} G_{sumn}(z) &= (1 - \alpha)G_{M1n}(z) + \alpha G_{M2n}(z) \\ &= ((1 - \alpha)a_1 + \alpha a_2)(z^2 + u(\alpha)z + v(\alpha)) \end{aligned} \quad (32)$$

ただし、

$$\begin{aligned} u(\alpha) &= \frac{(1 - \alpha)a_1b_1 + \alpha a_2b_2}{(1 - \alpha)a_1 + \alpha a_2} \\ &= \frac{a_1b_1 - a_2b_2}{a_1 - a_2} + \frac{a_1a_2(b_1 - b_2)}{\alpha - \frac{a_1}{a_1 - a_2}} \end{aligned} \quad (33)$$

$$\begin{aligned} v(\alpha) &= \frac{(1 - \alpha)a_1c_1 + \alpha a_2c_2}{(1 - \alpha)a_1 + \alpha a_2} \\ &= \frac{a_1c_1 - a_2c_2}{a_1 - a_2} + \frac{a_1a_2(c_1 - c_2)}{\alpha - \frac{a_1}{a_1 - a_2}} \end{aligned} \quad (34)$$

となる。一般的に伝達関数 $G(z)$ が 2 次の分子多項式を持つ場合、その係数 $(u(\alpha), v(\alpha))$ で表わされた点 $P(\alpha)$ が Fig. 6 で示されるような三角形内にあれば、 $G(z)$ は不安定零点を持たないことが知られている。

また、(33), (34) 式から点 $P(\alpha)$ の軌跡は $u-v$ 平面上で直線となることがわかる。この直線は α を ∞ に変化させた場合、 $((a_1b_1 - a_2b_2)/(a_1 - a_2), (a_1c_1 - a_2c_2)/(a_1 - a_2))$ に収束し、 $\alpha \approx a_1/(a_1 - a_2)$ の場合は、直線

Fig. 6 Stable area

の両端に発散する。この直線と三角形が交わっていれば、(29) 式の不安定零点を安定化するような α が存在することになる。また直線と三角形が交わる場合、その二つの交点を u 座標の小さいものから $P(\alpha_l)$, (α_r) とすると、それぞれの交点での α は

$$\alpha_l = \frac{-a_1}{a_1 - a_2} \times \left(\frac{a_2((b_1 - b_2) + (c_1 - c_2))}{(a_1 - a_2) + (a_1 b_1 - a_2 b_2) + (a_1 c_1 - a_2 c_2)} - 1 \right) \quad (35)$$

$$\alpha_r = \frac{a_1}{a_1 - a_2} \times \left(\frac{a_2((b_1 - b_2) + (c_1 - c_2))}{(a_1 - a_2) + (a_1 b_1 - a_2 b_2) + (a_1 c_1 - a_2 c_2)} + 1 \right) \quad (36)$$

と表わされる。

以上よりサンプル周期に対して m を横軸、 α を縦軸とし、伝達関数が安定となるパラメータの組み合わせを示す図を描ける。例としてサンプル周期 10ms の場合

Fig. 7 m v.s. magnitude ratio α

の実験に用いるロボットマニピュレータの 1 軸の m と α_r , α_l の関係を表わすグラフを Fig. 7 に示す。実線が α_l を示し、破線が α_r である。実線の上側でかつ、破線の下側である領域が零点を安定とする領域である。この領域内より二つのパラメータ m , α を選べば短いサンプル周期でも安定な学習を実現できる。

5.2 パラメータ設計指針

ここでは m と α の二つのパラメータを安定領域内から選ぶ指針について述べる。サンプル点上の誤差の収束を考えるだけならば、安定領域からパラメータを任意に選べば良い。しかし高精度の軌道を実現するためには、サンプル点間の応答についても考慮すべきである。本学習制御法はサンプル点間誤差を学習に反映できないが、学習オペレータのパラメータ設計段階で学習が収束した後に過大なサンプル点間誤差を生じさせないようにパラメータを設計することは可能である。

Fig. 8 に α と m を変化させた場合の学習が収束した後の 1 軸のサンプル点間誤差の数値計算結果を示す。この数値計算はサンプル周期を 10ms とし次節の実験条件にできるだけ近い条件下で行った。ただし学習が収束した後のサンプル点間誤差の計算に関しては制御対象を線形システムとみなし、与えられた目標入力に追従するように $U(z) = G_M^{-1}(z) Y_d(z)$ として得られた補償入力をもとに計算している。またこのときのサンプル点間の目標軌道は各サンプル間を直線補完したものを使用した。図において $0.5 < \alpha < 1$ 付近の領域はどのようなパラメータの組み合わせをもってしても安定化は不可能な領域であることからプロットが行われていない。この結果を見ると、 α の値が負の値側に変移する場合にサンプル点間誤差が広がっているのが確認できる。($\alpha > 1$ の領域は存在する m の値が少ないことから議論の対象外とする。)

Fig. 8 Magnitude ratio α v.s. trajectory error

また m の値が小さいほど α の変化により誤差が変動しやすいことも分かる。この結果からパラメータの設定に際しては、 m の値はできる限り大きく取り、 α の値はその m の値で安定条件を満たす最大のものを使用すれば良いことがわかる。これに対して、例えば $m = 0.2$ で $\alpha > 0$ の領域のパラメータを選定しても問題が無いように思われるが、実際にはパラメータ誤差などから差異を生じるため、 α の微小な変動で誤差が大きく変わるような設定は避けるべきである。また、あまり m の値を小さくし、 α の値を負にすると、システムに加えるべき電圧が大きなものとなり、入力飽和を引き起こす。したがって、 m の値はできるだけ大きな方が望ましく、また $\alpha \ll -1$ として入力過多となってしまうことは避けるべきである。さらに線形近似モデルと実機との誤差を考えると、実機に適用する場合には二つのパラメータの設定には実際にはある程度余裕を見込む必要がある。あまり安定領域の境界付近でパラメータを設定すると、線形近似モデルには不安定零点が存在しなくとも、実機では存在する場合があります、学習が不安定になることがある。以上より、パラメータの設定法に関するフローチャートを示すと Fig. 9 のようになる。

以上の解析結果の妥当性を 6. にて実験結果より示す。

6. 実験結果

6.1 実験に用いたシステム構成

5. までに示した理論を実験によって検証する。実験には Fig. 10 に概略が示されるような 2 自由度スカラ

Fig. 10 SCARA type 2 d.o.f. manipulator

Fig. 9 Flow chart to design the two parameters

型ロボットマニピュレータを用いた。またリンク 1 の長さ $l_1 = 0.25$ m, リンク 2 の長さ $l_2 = 0.16$ m である。ほかのシステムパラメータを Table 1 に示す。このマニピュレータを DSP ボードを内蔵したコンピュータ (PC 9801 RX) により制御する。コンピュータは学習制御部分を主に受け持ち、ほかに DSP とのデータの受け渡しと制御タイミングの管理を行っている。2-delay 入力方式を用いない 1 入出力システムを実現する場合は、サンプル周期より遥かに短い間隔で発生する割り込みをカウントすることでサンプル周期を計り、サンプル周期 (一定カウント) ごとに DSP にフィードバック制御演算を行わせ、そのときのロボットの状態を取り込む処理を行った。今回入力切り換えタイミングパラメータ m によって指定された時間で入力を切り換える必要があるため、

Table 2 Feedback parameters

	K_p [Nm/rad]	K_v [Nm/rad]
joint 1	10.0	0.9
joint 2	10.0	0.5

サンプル点から指定された割り込み回数にて入力を切り換える処理を付加することで2-delay 入力制御を実現した。

6.2 実験条件

6.1 のようなシステム構成である2自由度スカラ型ロボットマニピュレータに対して、中心が $x = 0.2 \phi\text{m}$, $y = 0.2 \phi\text{m}$, 半径 $r = 0.10\text{m}$ の円軌道を角速度 20rad/s , 手先速度台形パターンとして2秒で描かせる実験を行った。フィードバック定数は Table 2 に示される値を用いた。

6.3 不安定零点の安定化の実験結果

サンプル周期を 10ms とし 6.2 の実験条件のもとで実験を行った結果を Fig. 11 に示す。(a) が従来の学習アルゴリズムの結果であり、不安定零点の影響から学習

(a) Conventional system

(b) 2-delay input system

Fig. 11 Experimental results

が安定に行なわれていない。ちなみにこのときシステムの不安定零点は1軸が -1.28 , 2軸が -1.16 となっている。(b) が2-delay 入力制御方式を用いた学習制御系の結果であり、システムの不安定零点の影響を打ち消し、安定な学習が実現できているのが確認できる。二つのパラメータは $m = 0.8$, $\alpha = 0.0$ と設定した。

Fig. 12 に1軸の試行ごとの誤差の変移を示す。縦軸はサンプル点上の誤差の二乗積分値の平均値を、横軸は試行回数を表わす。比較のために同条件で従来の1入出力系に対して学習を行わせた結果を合わせて示す。従来のシステムでは学習が発散してしまっているが、2-delay 入力制御方式を用いた場合には学習が5回程で収束しているのがわかる。以上よりサンプル周期が短い場合でも2-delay 入力制御方式の学習制御系を用いることによって学習を安定に行うことができることが確認された。

なお、6.3の結果については、2軸においても1軸と同様の結果が得られている。また、サンプル周期が $T = 5\text{ms}$ の場合についても安定な学習が行われていることが確認されている。

Fig. 12 Convergence of trajectory error in case of $T = 10\text{ms}$

6.4 サンプル点間応答の実験結果

つぎにサンプル点間の応答改善の結果を示す。サンプル点間の軌道データを計測することは本システムでは困難であるため、サンプル周期を 30ms とし、そのリップルの大きさを評価することにした。目標軌道は 6.3 と同じとし、 $\alpha = 0$ とし m の値を 0.5 , 0.8 の二つの場合について実験を行った。サンプル周期 30ms の時の1軸の α に対する軌道誤差を Fig. 13 に示す。サンプル周期が 30ms の場合は2-delay 入力方式を用いない場合でもシステムが不安定零点を持たないことから $\alpha = 0.5$ (1入出力系と等価) の結果が得られており、その時の結果は m によらない。

Fig. 13 Magnitude ratio α v.s. trajectory error

Fig. 14 Experimental results

Fig. 14 に実験結果を示す。(a) が $m = 0.5$ のときの 10 回目の試行結果であり、軌道にリップルが発生しているのが確認される。(b) が $m = 0.8$ のときの 10 回目の試行結果であり、(a) の結果に比べてリップルが抑えられていることが確認される。これより m の値が大きいほどリップルが抑制されていることが確認できる。こ

れは 5.2 の数値計算結果とよく一致しており、 m の値が大きい方がサンプル点間応答が良くなることが確認された。これにより解析結果を適用してパラメータ設定を行うことの妥当性が確認できた。

7. おわりに

本論文では 2-delay 入力制御方式を用いた学習制御系の有効性を実機により検証し、その場合の学習オペレータのパラメータの設計法を示した。さらにサンプル点間のリップルを抑制するためのパラメータの設定指針について述べ、実験によりその妥当性を示した。

今回提案した設計法は、入力を途中で一度切り換えるだけなので計算量的にはほとんど問題がない。この方法は従来の 1 入出力系での逆システムに基づいた学習アルゴリズムを用いる場合よりもはるかにサンプル周期を短くすることが可能となることから、実用化が大いに期待できる手法であると考えられる。

参考文献

- 1) 川村, 宮崎, 有本: 学習制御方式のシステム論的考察; 計測自動制御学会論文集, Vol. 21, No. 5, pp. 445~450 (1985)
- 2) 美多, 加藤, 青木: 反復制御とロボットアームの軌道制御への応用; 計測自動制御学会論文集, Vol. 21, No. 11, pp. 1223~1230 (1985)
- 3) 石原, 阿部, 竹田: 離散時間繰り返し制御系の一設計法; 計測自動制御学会論文集, Vol. 22, No. 1, pp. 43~49 (1986)
- 4) 杉江, 小野: 学習制御に関する一考察; システムと制御, Vol. 13, No. 2, pp. 129~135 (1987)
- 5) 鈴木, 安江, 大熊, 内川: ロボットマニピュレータの離散時間学習制御法; 日本ロボット学会誌, Vol. 8, No. 2, pp. 179~187 (1990)
- 6) 早川, 曾根, 鈴木, 大熊: 離散時間学習制御法によるマニピュレータの軌道制御—実機による検証; 電気学会論文誌 D, Vol. 113, No. 5, pp. 655~664 (1993)
- 7) 早川, 近藤, 鈴木, 大熊: 2-Delay 入力制御方式を用いた離散時間学習制御法によるマニピュレータの軌道制御; ロボット学会論文誌, Vol. 12, No. 6, pp. 872~878 (1994)
- 8) K. J. Astrom, P. Hagander and J. Sternby: Zeros of Sampled System; Automatica, Vol. 20, No. 1, pp. 31~38 (1984)
- 9) 美多, 千田: 多入出力 2-Delay デジタル制御の提案と応用; 計測自動制御学会論文集, Vol. 24, No. 5, pp. 467~474 (1988)