改良型CMACによるマニピュレータの軌道追従制御

学生員 横山 隆造 (明治大)

非会員 高瀬 英一郎 (明治大)

正員 三木 一郎 (明治大)

Continuous Path Control of Manipulator Using an Improved CMAC
 Ryuzo Yokoyama, Student Member, Ei-ichiro Takase, Student Member,
 Ichiro Miki, Member (Meiji University)

The Cerebellar Model Arithmetic Controller (CMAC) is a means to achieve neural networks, and this has the advantages that the convergence time is short and learning speed per step is fast, compared with those of standard back-propagation neural networks. In the conventional design method of CMAC, a learning gain and quantizing intervals are determined by trial and error, and these values are constant. Therefore, it is very difficult to improve both the learning speed and the accuracy of CMAC at once through the conventional design method.

In this paper, a conventional design method of CMAC has been improved to overcome this problem. Furthermore, a continuous path control system of manipulator using this CMAC is presented. The effectiveness is verified by experimental results.

キーワード：CMAC, マニピュレータ, 量子化間隔, 学習係数, 軌道追従制御

1. まえがき

現在の産業界では、工業用ロボットを使用した作業のオートメーション化が進められ、さらに、生産性の観点から、ロボットの高速化、高精度化が要求されている⁽¹⁾。しかし、ロボットを高速に動作させるに伴い、その動作には非線形性の影響が著しく現れてくる。これを解決する一手段として、制御系にニューラルネットワークなどを用いる研究が行われている⁽²⁾。

我々は、ニューラルネットワークの実現方法の一つであるCMAC(Cerebellar Model Arithmetic Controller)⁽³⁾⁽⁴⁾について研究を行ってきた^{(5)~(8)}。CMACは、ニューロン単体ではなく小脳の働きをモデル化したもので、バックプロパゲーション法を用いる階層型ニューラルネットワークと比べると、収束が速く、1ステップ当たりの学習時間が短いという利点を持っている。

J.S.Albus氏がCMACを数学的にモデル化して以来、CMACの様々な応用がなされており、近年、CMACの構造や学習方法についても研究されている。例えば、D.E.Thompson氏⁽⁹⁾はCMACの学習に用いる入力点の選定を行う学習方法を提案した。これは、CMACにおける汎化⁽¹⁰⁾の悪影響を生じさせることなく学習を行う方法である。

提案手法及び入力点に対してランダムに学習回数を決定する手法の2通りのCMACを数学関数の学習に適用し、シミュレーション結果よりこれらの学習方法の比較を詳細に行っている。また、伊賀氏ら⁽¹⁰⁾はCMACの欠点である記憶容量の使用規模に着目し、効率的にCMACメモリの利用を実現する手法について述べている。この手法では、入力により選択される顆粒細胞を階層化することで従来のCMAC⁽⁴⁾の使用時における精度と同程度の精度を保ちながら、CMACメモリの節約を行った。そして、これをマニピュレータ制御の数値シミュレーションに適用し、従来のCMACを使用した場合との比較を行った。前者の文献では、実際に制御分野への応用についての議論がなされておらず、また、後者は学習精度の向上等については触れられていない。

従来型CMAC⁽⁴⁾⁽⁵⁾の設計方法では、学習係数や量子化間隔などの係数を試行錯誤的に設定しており、相当な時間と手数がかかることから設計者の大きな負担となっている。また、これらの係数は一定値に設定されるため、学習速度と精度を共に良好なものにするのは非常に困難である^{(10)~(12)}。学習精度の向上を図るために、学習方法について検討を行うことは非常に重要であると考えるが、我々は上述の問題点を解決するため、基本的に学習方法は変更せず、学習速

図1 CMACのモデル
Fig.1. Model of CMAC.

図2 荷重の選択
Fig.2. Selection of weights.

度及び精度が共に良好となるように係数を自動的に変化させる手法を用いた改良型CMACについて検討し、その有効性をシミュレーションにより確認してきた。

本論文では、この改良型CMACの一設計法について述べ、これを用いたマニピュレータの軌道追従制御系のシミュレーションおよび実験を行い、改良型CMACの有効性、さらに本制御システムの妥当性について明らかにする。

なお、本手法ではCMACの入力空間における入力間隔に着目し、これを用いて量子化間隔の設定を行うが、前述のように基本的に学習方法の変更は行わないため、各入力変数に対して同一の量子化間隔を設定する。したがって、本手法では、この同一の量子化間隔を前提として、同じパターンとなる入力変数を用いてCMACの入力空間が構成される場合を対象とする。

2. CMACの設計法

<2・1> 従来のCMAC

小脳皮質内の構造や、機能に関する知見が得られるに伴い、小脳皮質内の情報処理機構の数学的モデルが提案された。このモデルをJ.S.Albus氏が発展させ抽象化し、さらに数学的にも定式化したものがCMACである⁽⁹⁾。この概念図を図1に示す。CMACは、同図に示す一連の写像関係により定義され、入力空間内の入力、荷重の選択という形で

表1 学習係数の影響

Table 1. Effect of learning gain.

学習係数値	最終的な誤差	学習速度
大	大	速い
小	小	遅い

表2 量子化間隔の影響

Table 2. Effect of quantizing interval.

量子化間隔	最終的な誤差	学習速度
粗	大	速い
密	小	遅い

発火する細胞を決定する。次に、荷重の選択方法を図2に示す。同図は単純な2変数の場合であり、量子化関数を3とした例である。入力をそれぞれ量子化間隔3で量子化して荷重の表を作成する。例えば、 $S=(3,3)$ という入力ベクトルは荷重の表からWBb, WEe, WHhの3つの荷重を選択する。そして、これらの荷重の総和がCMACの出力となる。この出力が、与えられた教師信号と異なる場合、その偏差を選択された荷重の個数で除し、さらに学習係数を乗じてこれを修正量とする。そして、これをそれぞれ選択された荷重に加え、荷重の調整を行う。以上の手続きにより、CMACの学習が進行する。ここで、修正量 ΔW の決定は次式で行われる。

$$\Delta W = g \frac{\hat{P} - P}{C} \quad \dots\dots\dots (1)$$

ただし、 $\hat{P}$ は教師信号、 P はCMACの出力、 C は選択された荷重の数、 g は1回の学習過程の中で行われる修正量を決める学習係数である。

ところで、CMACの学習機能は、学習係数や、量子化間隔に大きく影響される^{(10)~(12)}。表1、表2にそれぞれ学習係数、量子化間隔が基本的な学習機能に与える影響を示す。学習係数は、学習速度と精度に大きなかわりを持つパラメータである。表1からわかるように、学習係数を大きく設定すると、学習速度は速く、収束するまでの学習回数は少なくて済むが、最終的な誤差は大きくなる。逆に、学習係数を小さく設定すると、最終的な誤差は小さくなるが、学習速度は遅くなり、学習回数が多くなる。これは(1)式からわかるように、学習係数を大きく設定すると荷重の修正量が大きくなり、小さく設定すると荷重の修正量が小さくなるために生じる現象である。また、表2より量子化間隔の粗密によっても誤差、学習速度が影響される。これは、量子化間隔を設定すると、選択される荷重の

数が決定されるため、(1)式で用いられるCが決まり、荷重の修正量に影響することにより生じる問題である。

以上で述べた学習係数及び量子化間隔の設定は、従来のCMACでは試行錯誤的に行われるが、互いに荷重の修正量に影響するため、設定には膨大な時間と手数がかかる。また、最終的な誤差をできるだけ小さく、しかも学習速度を速くすることは非常に難しい問題であることがわかる。

<2・2> 改良型CMACの一設計法

従来型のCMACでは、学習係数と量子化間隔を定数としているため、前節で述べたような問題点を生じる。そこで、学習係数と量子化間隔を自動的に変化させることによって、この問題を解決することを考えた。以下にそれぞれの設定方法について述べる。

(1) 学習係数の設定方法

学習係数は、表1の影響を考慮に入れて自動的に変化させることを考える。まず、学習の初期段階では大きな学習係数を用いることによって、学習を促進する。しかし、大きな学習係数を継続して用いると精度が悪化するため、ある程度学習が進んだ段階で係数を小さなものに切り替えるという手法をとる。

係数の切り替えは、現在の学習係数で学習が進行しなくなったときに行う。学習がある程度進行してくると、学習1ステップ中に選択される荷重は、それまでの学習で少なくとも1度は選択された荷重である。よって、係数の切り替えを行う時の状態は<3・1>節でその詳細を後述するが、現在の学習ステップにおいて、それまでに選択された荷重以外に新たな荷重は選択されず、教師信号との誤差が次式で表せるときである。

$$\left. \begin{aligned} \Delta(n-1) - \Delta(n) &\leq 0 \\ \Delta(n) &= \sum |\delta(i)| \end{aligned} \right\} \dots\dots\dots (2)$$

ただし、nは学習回数、 $\delta(i)$ は1サンプリングごとの誤差、 $\Delta(n)$ は総誤差、つまり学習1ステップ中におけるサンプリングごとの誤差の総和である。

また、学習係数gの変化方法は(3)式のように変形したシグモイド関数を用いて行われ、学習が進まなくなった時点における総誤差をパラメータとして変化させる。

$$g(\Delta(n)) = \frac{1}{\{1 + \exp(-\beta\Delta(n) + \alpha)\}} \dots\dots\dots (3)$$

ただし、 α 、 β は定数である。

(2) 量子化間隔の設定方法

量子化間隔は、表2の影響を考慮に入れて設定する。量子化間隔が一定の場合、入力等間隔ならば汎化作用⁽³⁾が均等に働き、さほど問題はない。しかし、入力が等間隔でない場合は、一定の量子化間隔では良好な学習を行うことができなくなる。そこで、入力間隔が変化しても平等な汎化作用を生じることができるよう、以下に示す方法で量

子化間隔を変化させる。まず、(4)式により入力の差分を求め、(5)式で示すように整数Rで量子化した値F(t)を使用し、最終的に(6)式により量子化間隔を設定する⁽³⁾⁽⁸⁾。この手法により、入力の間隔に応じて自動的に量子化間隔C(t)を設定できる。

$$f(t) = \left[\frac{dS(t)}{dt} \right]_{t=kT} \doteq \left| \frac{S(kT+T) - S(kT)}{T} \right| \dots\dots (4)$$

$$F(t) = \left[\gamma \frac{f(t)}{R} \right] + 1 \dots\dots\dots (5)$$

$$C(t) = F(t)R \dots\dots\dots (6)$$

ただし、tは時間、(4)式において、S(t)は入力、Tはサンプリングタイム、kはサンプリング回数である。また、(5)式における γ は係数、Rは量子化の値、

$\left[\gamma \frac{f(t)}{R} \right]$ は $\gamma \frac{f(t)}{R}$ を越えない整数を表す。

ここで、本設定法による荷重の選択方法について一例を挙げて説明する。例えば入力関数S(t)が、次式のように時間の経過につれてその傾きが大きくなる場合における量子化間隔の設定法について考える。

$$S(t) = t^4 \dots\dots\dots (7)$$

ただし、tは時間[s]である。

いま、(4)式において $T=1.0[s]$ 、(5)式において $\gamma=1$ 、 $R=4$ とすると、 $t=0.0, 1.0, 2.0, 3.0$ におけるそれぞれの量子化間隔C(t)は(4)式～(6)式より以下のようになる。

$$\{C(0.0), C(1.0), C(2.0), C(3.0)\} = \{4, 4, 8, 12\} \dots\dots (8)$$

このときの時間に対する量子化間隔の変化の様子を図3に、入力により選択される荷重を図4に示す。図4は、0から8までの1次元入力空間において、(7)式の各時間におけるS(t)を入力とした場合であり、コード化した際に名前をつけたアルファベットは荷重を示す。例えば、入力S(0.0)は(8)式より量子化間隔が4であるので、図4における入力空間の0から3までを従来のCMACの場合と同様に、1/4ずつシフトして量子化する。したがって、この領域はA,C,D,F,G,I,Jの荷重でCMACメモリが構成され、この入力により選択される荷重はA,D,F,Iとなる。そして、これらの総和がCMACの出力となり、荷重の調整は(1)式を用いて行われる。この例の場合には全体の荷重は図4のように決定される。次に、各入力間で共通に選択される荷重の数を求める。例えば入力の傾きが $f(0.0)=1$ のとき、入力S(0.0)とS(1.0)によりそれぞれ選択される荷重のうちA,D,Iの3つの荷重が共通に選択されることがわかる。同様に他の傾きについて共通に選択される荷重の数を、ま

図3 量子化間隔

Fig.3. Quantizing interval.

図4 荷重

Fig.4. Weights.

表3 共通な荷重の数

Table.3. Numbers of common weights.

入力傾き	本手法	従来法(1)	従来法(2)
$f(0.0) = 1$	3	3	7
$f(1.0) = 2$	2	2	6
$f(2.0) = 4$	4	0	4

た、比較のために従来法(1)、(2)として量子化間隔をそれぞれ4及び8に設定した場合のその数を表3に示す。同表において、従来法(1)では入力傾きが $f(2.0)=4$ のとき、共通に選択される荷重は0となり、汎化作用は全く働かない。また、従来法(1)よりも量子化間隔が大きい場合である従来法(2)では、 $f(0.0)=1$ のとき、共通に選択される荷重の数多く、汎化作用が大きく働くことが分かる。しかしこの場合、表2で示したように学習速度は速いが最終的な精度は悪化するおそれがある。したがって、この例をとってみても、従来法に比較して本手法の方が、入力間隔の大小によって共通に選択される荷重の数にばらつきが少なく、平等な汎化作用が得られると考えられる。この効果は、入力間隔がさらに大きくなると、より顕著に現れる。

なお、(3)式の α 、 β 及び(5)式の γ 、 R の値の選定方法は4章で詳述する。

3. 改良型CMACを用いたマニピュレータの軌道追従制御

マニピュレータの軌道追従制御に、前章で述べた改良型CMACを直接コントローラとして用いた。これは、CMACには次のような特長があり、制御システムへの使用に適していると考えられるからである。

- ・非線形写像が行える。
- ・事例から学習でき、モデリングが行える。
- ・汎化作用を有する。

以下に、本研究で用いたCMACの実際の設計方法と制御システムについて述べる。

<3・1> CMACの設計方法

CMACは多入力1出力であるので、それぞれのアクチュエータに1つのCMACを用いた。CMACの入力は、目標角度 $\theta_d(t+T)$ とマニピュレータの実角度 $\theta(t)$ とした。また、直接コントローラとして用いるため、出力は制御電圧である。

(1) 学習係数の設定

(2) 式における $\delta(i)$ には、サンプリングごとの角度偏差を用いる。従って、 $\Delta(n)$ は学習1回ごとの角度偏差の総和になる。(3)式における β は、学習係数 $g=1$ で学習が終了した時点で(3)式中の指数関数が、 α のみの簡単な関数となるように $\beta=2\alpha/\Delta_f$ と設定した(4章参照)。ただし、 Δ_f は学習係数 $g=1$ で学習が終了した時の総誤差である。最終的に、学習係数は

$$g(\Delta(n)) = \frac{1}{1 + \exp\left(-\frac{2\alpha}{\Delta_f} \Delta(n) + \alpha\right)} \quad \dots\dots\dots (9)$$

となる。学習1ステップ終了時に毎回、CMACに使用するメモリ数及び(2)式を計算し、その結果メモリ数の増加がとまり、かつ現在の学習終了時の総誤差が、学習における1ステップ前の総誤差以上になった時にはじめて学習係数値の切り替えを行う。そして、一度設定された係数値は、再び(2)式の条件が成立するまで一定値に保たれる。学習係数の初期値は、学習の初期段階において速度に重点をおき学習を行わせるため、学習係数の最大値である1とした。

(2) 量子化間隔の設定

CMACへの入力が角度であるため、入力の差分は目標角速度 $\omega_d(t+T)$ で置き換えることができる。従って、 $t=t_s$ のときの量子化間隔は、(5)式及び(6)式より

$$C(t_s) = \left\lceil \gamma \frac{\omega_d(t_s + T)}{R} \right\rceil + 1 \quad \dots\dots\dots (10)$$

となる。

ただし、 γ は角速度の大きさを量子化間隔レベルの大きさに変換するために用いられる係数である。

検出角速度がノイズ等の影響を受けた場合、これを用いて設定を行った量子化間隔では、荷重の選択が不安定となり良好に学習を行うことが困難となる。本研究では基本的にはCMACの学習方法は変更しないので、目標角速度を用いた(10)式により、各入力変数に対して同一である量子化間隔の設定を行う。

<3・2> 制御システム

CMACの学習には教師信号が必要となるため、修正量 ΔW は、 $t=t_s$ のとき角度偏差と角速度偏差を用いて次式のように決定した。

$$\Delta W = g \frac{g_1(\theta_d(t_s) - \theta(t_s)) + g_2(\omega_d(t_s) - \omega(t_s))}{C(t_s)} \quad \dots\dots\dots (11)$$

ここで、 g_1 、 g_2 は係数であり、それぞれの偏差を電圧レベルにするものである。

本論文では、修正量に角速度を用いているため、変換回路を通して角速度を求めている。図5に改良型CMACを用いた制御システムを示す。

<3・3> 実システムの構成

実験に用いたマニピュレータのモデルを図6に示す。また、DCモータの定格およびマニピュレータの諸定数を表4、表5にそれぞれ示す。実システムは、以下に説明する

図5 制御システム
Fig.5. Control system.

ものから成り立っている。

● マニピュレータ

本実験では、ダイキン工業社のMR1を使用した。これは、平行4リンク機構を持つものであり、アクチュエータはDCモータである。また、角度検出装置はポテンシオメータである。

● PC-9801BX

本実験では、制御装置としてPC9801BXにCPUアクセラレータを装着して用いている。また、CMACは大きなメモリを使用するため、RAMを13312Kバイトに増設している。

● アクチュエータ駆動回路

駆動回路として、PWMチョッパを試作し、これを用いた。チョッパ部には、MOSFETモジュールを使用した。

4. 結果及びその検討

従来型CMACと改良型CMACを用いて、マニピュレータの軌道追従制御のシミュレーション及び実験を行った。 θ_1 、 θ_2 の目標軌道は図7(a)とし、図7(b)に示す台形型速度パターンを用いている。制御時間は2.5[s]とし、制御周期は10[ms]とした。CMACの修正量に

図6 マニピュレータのモデル
Fig.6. Model of manipulator.

表4 DCモータの定格
Table 4. Rating of DC motor.

出力	11[W]
電圧	42[V]
電流	0.5[A]
回転数	3000[rpm]

表5 マニピュレータのパラメータ
Table 5. Parameters of manipulator.

	Link1	Link2
リンク長	0.128[m]	0.050[m]
質量	0.028[kg]	0.017[kg]

用いる諸係数は $\alpha=2.31$ 、 $\gamma=50$ 、 $R=5$ 、 $g_{11}=0.5$ 、 $g_{12}=0.25$ 、 $g_{21}=g_{22}=0.1$ とした。

ここで、学習係数 g を示す(3)式における α 、 β の決定法について述べる。＜3・1＞節で述べたように、 β を(9)式に示したように $2\alpha/\Delta_f$ と設定すれば、学習係数 $g=1$ で学習を開始し、学習が進まなくなった時点ではじめて学習係数の切り替えを行う際、 $\Delta_f=\Delta(n)$ となるので(9)式の指数部は $-\alpha$ となり、 g は α のみで決定されることになる。本研究では実際にCMACの設計を行う前に、総誤差の変化による学習係数値のデータを収集した。すなわち、(9)式における総誤差 Δ_f を任意の大きさに決め、 $\Delta(n)$ を可変にして任意の α を入力した。こうして、いくつかの α に対する学習係数値のデータを獲得し、このデータより学習係数が図8に示すように、誤差の減少に伴いできるだけなめらかに減少する、目安となる α を選定した。同図は、本研究で用いた任意の総誤差 Δ_f の範囲内において、 $\Delta(n)$ をパラメータとして(9)式より決定される学習係数値の変化の様子を示す。次

図7 目標軌道

Fig.7. Desired orbit.

図8 学習係数の変化

Fig.8. Change of learning gain.

図9 量子化間隔の変化

Fig.9. Change of quantizing interval.

に、その α を微小に変化させて再度、総誤差の変化による学習係数値のデータを集める。前述のように、学習係数の初期値 $g=1$ における学習終了時に(9)式を用いて学習係数値の切り替えを行うが、この値が0.9となるような α (本研究では $\alpha=2.31$)を設定した。本研究では、はじめての切り替え後の係数値を比較的大きな値に設定すれば学習速度を向上させることができると考え、 $g=0.9$ と設定した。

量子化間隔の設定においても(10)式に任意の γ 、 R の値を代入し、時間の推移に対する量子化間隔のデータを収集した。入力間隔の大小にかかわらず汎化作用を均等に働かせるために、(10)式により量子化間隔を自動的に設定するが、その汎化の程度に関わりをもつ近傍範囲(同一とみなす範囲)を次のように決定する。本研究では図7(a)に示す目標角度を設定した場合、角速度が最大であるときに入力間隔が最も大きくなり、この値は0.3[deg]となる。そこで、近傍範囲を0.6[deg]に設定すれば、最も大きな入力間隔をもつ隣りあう2つの入力の間においても十分に汎化を作用させることができると考えた。本研究では入力の分解能を0.02[deg]としているので、最も入力間隔が大きいところの量子化間隔は30となる。また同様に、角速度が0以外の最も小さい時の入力間においても近傍範囲を決めると0.1[deg]となるので、この量子化間隔は5となる。(10)式より最小の量子化間隔は、量子化の値 R に等しくなることから、この場合 $R=5$ となる。そして、これを用いて量子化間隔が30となる $\gamma\omega_d(t)$ を量子化すると、図9のように量子化間隔が6段階に変化する。したがって、(10)式に任意の γ 、 R の値を代入して獲得した量子化間隔のデータから、 $R=5$ の時に同図に示す量子化間隔となる γ (本研究では $\gamma=50$)を選定した。ここで図9は、本研究で用いた時間の推移によって変化する量子化間隔 $C(t)$ を示す。また、同図において $\gamma\omega_d(t)$ は、係数 γ を乗ずることにより角速度の大きさを量子化間隔レベルの大きさに変換したものである。上述したこれらのデータは、実際にシステムを動かす前に(9)式、(10)式を用いて容易に獲得できるため、CMACの設計に要する時間及び手数は、従来と比較してかなり軽減することができる。

また、リンク j の修正量に用いる i 個の係数 g_{ij} は、CMACの学習係数と同様に $0 < g_{ij} \leq 1$ ($i=1,2, j=1,2$)の範囲内で決定する。 g_{11} は角度偏差を電圧レベルにする係数であるが、学習速度や精度への影響を考慮して範囲内における中間値を目安に設定した。そして、 g_{12} はリンク長に比例した値を、 g_{21} 、 g_{22} は範囲内における微小な値をそれぞれ目安として設定を行った。 g_{21} 、 g_{22} は実際にはオーバーシュートを抑制する目的で設定される係数である。

従来型と改良型CMACを用いて軌道追従制御のシミュ

レーションを行った。リンク1とリンク2各々について学習回数に対する総誤差を求めた結果を図10に示す。また、図10における学習200回終了後の総誤差の値を表6に示す。これらの図及び表より、従来型に比べ改良型の方が学習速度、精度とも良い結果を示していることがわかる。

図11はシミュレーション上で改良型CMACに200回の学習を実行させ、そのCMACを用いて得られた軌道応答の結果である。図12は改良型CMACによるそのときの軌道誤差の時間的推移を示している。これらの結果より、マニピュレータは高精度で目標軌道に追従していることがわかる。

図13および図14は軌道追従制御の実験結果である。シミュレーション時と同じ条件で行った実験である

が、目標軌道に良好に追従しているもののリプルが観測される。この原因として、まず、本研究で用いたマニピュレータでは減速比の小さなギヤが用いられているが、多少のバックラッシュが存在していると考えられる。また、その他の要因によるノイズの影響があると考え、マニピュレータを初期姿勢で固定し、チョッパから供給される正負の方形波電圧の平均値を0[V]とした電圧を印加した状態で、ポテンシオメータにより角度の検出を行った。この実験結果を図15に示す。同図では、わずかなリプルが存在しているが、チョッパをオフした状態で測定するとノイズがほとんど現れないことから、これはチョッパの正負の出力方形波電圧によるモータのトルクリプルであることが考えられる。また、ポテンシオメータによる角度検出値をA/D変換して計算機に取

図10 総誤差
Fig.10. Total error.

表6 学習200回終了後の総誤差

Table 6. Total error after completing 200times learning.

	Link1	Link2
改良型CMAC	531.46[deg]	563.11[deg]
従来型CMAC	619.97[deg]	653.50[deg]

図11 軌道応答

Fig.11. Orbital response.

図12 軌道誤差

Fig.12. Orbital error.

り込み、その値をグラフ化すると図16となった。同図では、図15に比べかなりリブルが多く、かつ大きくなっている。この原因は計算機の介在が影響していることから、ポテンシオメータからの測定電圧を計算機内に取り込む際、A/D変換ボードを通す過程において、計算機内部でノイズが重畳されたと考えられる。したがって、前述のバックラッシュの影響と共に、このようなノイズを含むデータがCMACに入力される可能性が高く、これを用いて出力制御電圧を計算するため、図13及び図14に示すように実験結果に誤差が生じたと考えられる。今後、さらに精度を高めるためには、これらの点に

ついて引き続き検討する必要がある。

しかしながら、これらの結果を全体的にみると、改良型CMACを用いた本制御システムは十分な精度をもって軌道追従制御を行っていると考えることができる。

以上の結果より、改良型CMACの有効性が実験的にも明らかになった。また、改良型CMACを用いた本制御システムにより、高精度なマニピュレータの軌道追従制御を実現できることが分かった。

5. あとがき

マニピュレータの軌道追従制御にCMACを用いるにあた

(a) Link1

(b) Link2

図13 軌道応答
Fig.13. Orbital response.

(a) Link1

(b) Link2

図15 ポテンシオメータの出力
Fig.15. Output of potentiometer.

(a) Link1

(b) Link2

図14 軌道誤差
Fig.14. Orbital error.

(a) Link1

(b) Link2

図16 ノイズの影響
Fig.16. Effect of noise.

り、従来のCMACが持つ欠点を補う改良型CMACを示し、その妥当性を明らかにするために、シミュレーション及び実験を行った。その結果次のことが明らかになった。

(1) CMACの学習係数と量子化間隔を自動的に変化させる方式を採用することにより、設計に要する時間と手数を軽減することができた。また精度と学習速度を共に良好にすることができた。

(2) マニピュレータの軌道追従制御に改良型CMACを用いることにより、従来型CMACを適用したシステムよりも高精度なシステムを構成できた。

今後は、さらに精度向上を目指すと共に、本質的な問題である学習方法に対する検討や、作業座標上での制御などについても検討し、より一層の改善を進めていく予定である。

最後に、本研究の遂行に貢献された(株)キヤノンの牛久徹氏に感謝する。

(平成7年7月4日受付、平成8年5月30日再受付)

文 献

- (1) 末広, 大川, 黒須: 「低減速関節を有するマニピュレータの適応制御」, 日本機械学会論文集, Vol.57, No.544, pp.3901/3906, C, 1991
- (2) 辻, 西田, 高橋, 伊藤: 「神経回路網によるマニピュレータの能動的重力補償」, 平元電気学会産業応用大全, 193
- (3) J. S. Albus (小杉他訳): ロボティクス, 啓学出版, 1984
- (4) J. S. Albus: 「A New Approach to Manipulator Control: The cerebellar model articulation controller (CMAC)」, Transactions of the ASME Journal of Dynamic Systems, Measurement, and Control, Vol.97, No.3, pp.220/227, Sept. 1975
- (5) 牛久, 永山, 三木: 「CMACによるマニピュレータの軌道追従制御」, 平5電気学会大会, 714, 6-147
- (6) 牛久, 高瀬, 三木: 「CMACによるロボットマニピュレータ制御用CMACの一設計法」, 平5電気学会産業応用大会, T-35
- (7) 高瀬, 牛久, 三木: 「マニピュレータの軌道追従制御に用いるCMACの一設計法」, T.IEE Japan, Vol.114-D, No.12, pp.1302/1303, 1994
- (8) 高瀬, 横山, 三木: 「CMACによるマニピュレータの軌道追従制御の実験的検討」, 平6電気学会産業応用大会, T-44
- (9) D.E.Thompson, S.Kwon: 「Neighborhood Sequential and Ran-

dom Training Techniques for CMAC」, IEEE Trans. on Neural Networks, Vol.6, No.1, pp.196/202, 1995

- (10) 伊賀, 飯国: 「階層型メモリ構造をもつCMACの設計法」, 計測自動制御学会論文集, Vol.28, No.10, pp.1224/1230, 1992
- (11) 狩野: 「神経回路モデルCMACによるロボット・マニピュレータの学習制御」, 富士通(株)国際情報社会科学研究所第5回夏期シンポジウム報告書, pp.164/173, 1988
- (12) 狩野, 高山: 「神経回路モデルCMACによるロボット・マニピュレータの学習制御」, 第32回システムと制御研究発表講演会予稿集, pp.181/182, 1988

横山 隆造

(学生員) 昭和46年8月13日生。平成6年3月明治大学理工学部電気工学科卒業。同年4月同大学大学院理工学研究科博士前期課程に進学, 現在に至る。

高瀬 英一郎

(非会員) 昭和45年8月28日生。平成5年3月明治大学理工学部電気工学科卒業。同年4月同大学大学院理工学研究科博士前期課程に進学。現在、中部電力株式会社に勤務。

三木 一郎

(正員) 昭和25年9月20日生。昭和53年3月明治大学大学院博士課程単位修得退学。同年同大学工学部助手。専任講師、助教授を経て平成2年4月同理工学部教授, 現在に至る。昭和62年3月～昭和63年3月ケンタッキー大学, カリフォルニア大学デイビス校において客員研究員。工学博士。IEEE会員。