ファジィ・ニューロを利用した複腕ロボットによる
摩擦補償を考慮した対象物の滑らせ移動制御*木口量夫^{*1}, 福田敏男^{*2}Friction Compensation for Control of Drawing an Object
by a Dual-Arm Robot Using Fuzzy-Neuro

Kazuo KIGUCHI and Toshio FUKUDA

Manipulating an object is one of the most important tasks of robots. Numerous studies have been carried out on object manipulation algorithms and optimizing control force. When robots try to manipulate an object on some environments, in other words, when robots try to draw an object, they must take the effect of friction into account in order to control the object smoothly. In this paper, a friction compensation method is proposed for control of drawing an object using a specialized neuron which is connected to a fuzzy neural network. The specialized neuron learns the friction effect only when the object does not move against applied force. The fuzzy neural network is used for object trajectory control. The controller consists of a main fuzzy neural network controller which includes a neuron for friction compensation for object trajectory control and sub-fuzzy neural network controllers for manipulator force control to the object. Furthermore, modified Delta-Bar-Delta learning rate adaptation method is introduced for the main neural network controller. Computer simulation was performed to evaluate these proposed methods.

Key Words: Learning Control, Neural Network, Fuzzy Set Theory, Manipulator, Robot, Learning Rate Adaptation, Friction Compensation

1. はじめに

ロボットマニピュレータの高度な作業内容の一つとして複腕での対象物の操りが挙げられる。これまでに複腕ロボットマニピュレータによる対象物の操りについて多くの研究がなされており、いろいろな制御法や操り力の決定法が提案されてきた^{(1)~(8)}。制御法では、マスタ・スレイブマニピュレータの概念を利用したもの⁽¹⁾や、複数のマニピュレータのうち6自由度を位置制御に用い、残りの自由度を力制御に用いるハイブリッド制御タイプのもの⁽²⁾、あるいは対象物を直接制御するもの^{(3)~(7)}などがある。このうち各マニピュレータを力発生器とみなして対象物を直接制御する方法が、対象物操るものに最も自然な方法と思われ、著者らもファジィ・ニューラルネットワークを利用し、対象物の質量などの特性があいまいであったり未知であるときの制御法を提案してきた⁽⁶⁾⁽⁷⁾。

ファジィ・ニューラルネットワークは、あいまいな情報の扱いができ厳密な数学モデルを使うことなく制

御できるファジィ制御と、経験から学習できる能力をもつニューラルネットワークの両方の長所を合せたもので、前に挙げた著者らの研究例を含め、ロボット制御の分野でも積極的に取り入れられている^{(9)~(12)(15)}。例えば、ロボットマニピュレータで動特性の未知な対象物に目標どおりの力を与えるためには環境適応能力のあるニューラルネットワークを利用することなどが有効であるが、適応形のニューラルネットワーク制御器でも瞬時に適応できるわけではなく、適応するまでの間オーバーシュート等の危険性がある。そこで、ファジィ・ニューラルネットワークを利用した制御では、あらかじめファジィ制御の方法を用いてオーバーシュートしないような制御器を設計し、それをニューラルネットワークでモデル化し、バックプロパゲーション法で学習させることにより制御器を調整するようなことが実現できる⁽¹²⁾。

複腕ロボットマニピュレータで何らかの環境に接している対象物をその環境に接したまま動かす場合(例えば対象物を床上で引きずる場合)、対象物にクーロン摩擦が作用する。本稿では、このように対象物を環境に接したまま動かす制御を“滑らせ移動制御”と呼ぶことにする。既報では、ファジィ・ニューラルネットワークを用いることにより制御器がクーロン摩擦等の

* 原稿受付 1995年6月19日。

^{*1} 正員、新潟工業短期大学(〒950 21 新潟市上新栄町5 13-7)。^{*2} 正員、名古屋大学工学部(〒464 01 名古屋市千種区不老町1)。

外乱の補償を学習していくため、徐々に対象物を目標軌道どおり制御ができるようになることを示した⁽⁷⁾が、制御器が学習するまで少し時間がかかり、その間の対象物の動きも滑らかなものではなかった。そこで本稿では、力が与えられているにもかかわらず対象物が動かない場合だけ学習する摩擦補償用ニューロンと、対象物が運動しているときだけ学習するファジィ・ニューラルネットワークを組合せた制御器を提案する。今回提案する“状況に応じて学習を切替える制御器”により、対象物にクーロン摩擦が作用していても摩擦補償用ニューロンで即座に摩擦を補償し、その後はファジィ・ニューラル制御器により対象物の運動を滑らかに制御することが可能となる。

ニューラルネットワークのバックプロパゲーション法において学習係数は、学習速度や安定性にかかわる重要なファクタであり、これまでも適切な学習係数を見つけるために多くの研究がなされてきた。例えば、Jacobs は四つのヒューリスティックな学習係数調整法を挙げたうえで、ニューラルネットワークのすべてのウェイトがそれぞれ別個の学習係数をもつことを主張しており⁽¹³⁾、Arabshahi らは学習係数をファジィ制御で調整することを提案している⁽¹⁴⁾。学習係数をファジィ制御で調整する方法は効果的であるが、学習係数の有効範囲をあらかじめ把握しておく必要があると思われる。したがって、適切な学習係数の範囲が予測できない場合は、Jacobs が提案しているような、学習係数を線形的に増加させたり指数的に減少させたりする Delta Bar Delta 学習法が効果的であると思われる。しかし Delta Bar Delta 学習法では学習係数が最適値より小さい場合、学習係数は常に一定の割合で増加するが、著者らは学習状況により増加の割合も変化させたほうがより効果的であると考え、学習状況に応じ学習係数が小さい場合の増加割合を変化させる新しい Delta Bar Delta 学習法も提案し、対象物操り用のファジィ・ニューラルネットワークと摩擦補償用ニューロンの学習に利用した。

これらの制御方法の効果は、コンピュータによるシミュレーションにより確認した。

2. 対象物の軌道制御

本稿では問題を簡単にするため、対象物の平面の並進運動のみを扱うものとする。何らかの環境に接しながら並進運動する対象物の運動方程式は次式のように表される。

$$m\ddot{x} + F_c = f_{ext} + \sum_{i=1}^n f_i \quad \cdots \cdots \cdots (1)$$

Fig.1 Grasped object by 2DOF robots

ここで、

m : 対象物の質量

x : 対象物重心の基準座標系での位置ベクトル

f_{ext} : 外部から対象物重心に働く力

f_i : マニピュレータ i で対象物重心に加える力

F_c : クーロン摩擦力

とする。本稿では、図1に示すような平面2DOFロボットマニピュレータで対象物を机上平面で滑らせながら操るようなモデルを考え、対象物は机で支えられるものとして対象物の重力補償は行わなかった。また、マニピュレータは対象物と点接触し、任意の力を加えられるものとする。

対象物操りのための目標力(マニピュレータで発生させる合力から内力のための力を除いたもの)は、各マニピュレータで平等に分担するものとする。ただし、各マニピュレータの中で目標操り力を加えることができないものがある場合、つまり対象物操りのための目標力がそのマニピュレータにより対象物を引く形になる場合のものを除き、残りのマニピュレータで平等に分担するものとする。また、対象物とマニピュレータとの接触は点接触とし、適当な把持力を加えている限り滑ったり離れたりしないものとする。本稿では適当な把持力が既知であると仮定し、常に一定の把持力が作用しているものとした。

式(1)より、目標どおりの加速度を発生するには、外部から働く力と対象物の把持に必要な力が既知であり摩擦力も補償されているとすると、後は対象物の質量がわかれば各マニピュレータで発生させる力が決まる。しかし本稿では対象物が未知であると仮定しているためマニピュレータで対象物に加える力は決まらない。そこで、まず熟練者の制御知識を用いて後件部が定数値をとる対象物操り用ファジィ制御器を設計し、それをニューラルネットワークでモデル化し、バックプロパゲーション法で学習する制御器を考える。ファジィ・ニューラルネットワークを利用して制御することにより「少し右に目標位置から外れていた少し左に押してやる」というような制御ができ、押してやる力が適切でなかったらバックプロパゲーション法によ

る学習で修正するということが可能となる⁽⁶⁾。

摩擦力が補償されていない場合も、制御開始後しばらくすればバックプロパゲーション法による制御器自身の学習により対象物を目標どおり制御できるようになる⁽⁷⁾。しかし、制御器が学習するまで少し時間がかかり、その間の対象物の動きも滑らかなものではなかった。そこで本稿では、摩擦補償用ニューロンで摩擦力の補償を行い、対象物操り用ファジィ・ニューラルネットワークで対象物の滑らせ移動制御を行う制御器を用い、状況に応じて学習を切換えながら制御を行う方法を提案する。本制御器により、より早く対象物の運動を滑らかに制御することが可能となる。

3. 制御器の構成

本制御器は対象物を滑らせ移動制御するために必要な力指令を出力するメインの制御器 (Object Controller) と、各マニピュレータを力発生器と考え、動特性の未知な環境に対しメイン制御器からの力指令どおりの力を各腕から出力させるための二つのサブの制御器 (Arm Controllers) の複数の制御器から構成されている。このシステムのブロック線図を図2に示す。メインとサブの制御器から成る構成自体は既報の制御器と同じである⁽⁶⁾⁽⁷⁾。メイン制御器への入力是对象物の目標位置とフィードバックされた実際の位置情報であり、メイン制御器からの出力つまり各マニピュレータで発生させる目標力と力センサからフィードバックされた実際の力情報がサブ制御器への入力である。この各サブ制御器からの出力が各マニピュレータへの入力となる。

本稿で提案するメイン制御器は、力が与えられているにもかかわらず対象物が動かない場合だけ学習する摩擦補償用ニューロンと、対象物が運動しているときだけ学習する対象物操り用ファジィ・ニューラルネットワークとを組合せた制御器であり、対象物の運動状況により制御器内の学習する部分を切替えるということにも特徴がある。この学習の切替えは対象物の位置情報をもとにして、力が与えられているにも関わらず対象物が動かない場合は対象物操り用ファジィ・ニューラルネットワークの学習係数を零に、対象物が運動しているときは摩擦補償用ニューロンの学習係数を

Fig. 2 Block diagram of the system

零にすることにより行われる。メイン制御器からの出力は、対象物操り用ファジィ・ニューラルネットワークからの出力と摩擦補償用ニューロンからの出力を合せたものである。この組合せにより、対象物にクーロン摩擦が作用していても摩擦補償用ニューロンで即座に摩擦補償を学習し、その後は摩擦補償用ニューロンで摩擦を補償したまま、対象物操り用ファジィ・ニューラルネットワークにより対象物の運動を滑らかに制御することが可能となる。摩擦補償用ニューロンが学習しているときは、対象物操り用ファジィ・ニューラルネットワーク部では学習はしないが初期設定どおりの出力は出すことになっている。このメイン制御器の対象物操り用ファジィ・ニューラルネットワークの詳細を図3に示す。ここで図中の Σ のニューロンは入力の和を出力し、 Π のニューロンは入力の積を出力するものである。

対象物の滑らせ移動制御のためのメイン制御器では入力情報を基に各マニピュレータで発生すべき目標力を出力し、その目標力どおりの力を各マニピュレータで発生するようサブ制御器である各マニピュレータ力制御器でフィードバック制御する。各マニピュレータから目標どおり発生された力により対象物が運動し、その情報が対象物の制御器へフィードバックされる。

この章ではおもにメイン制御器について説明し、サブ制御器については4章で説明することとする。

3.1 摩擦補償用ニューロン この節では対象物

Fig. 3 Architecture of the neural network (object controller)

の滑らせ移動制御のためのメイン制御器内の摩擦補償用ニューロンについて説明する。このニューロンは、摩擦補償が必要な方向(1あるいは-1)を入力とし、その入力にウェイト値を掛けたものを摩擦補償力として出力する。したがって、ある程度学習した後のウェイト値はクーロン摩擦力とほぼ同じ大きさになっている。この出力と対象物操り用ファジィ・ニューラルネットワークからの出力とを合せたものがメイン制御器からの出力となる。

この摩擦補償用ニューロンは、力が与えられているにもかかわらず対象物が動かない場合のみ学習するものとし、対象物が運動しているときは学習しないものとする。したがって、対象物が運動し始めた後学習が対象物操り用ファジィ・ニューラルネットワークに切替わっても、摩擦補償用ニューロンはそれまでに学習した力を一定に出力し続け、対象物の運動方向が反転したら摩擦補償用ニューロンへの入力の正負も反転するので、運動の方向にかかわらず一定の力で摩擦補償を行うことができる。

3・2 対象物操り用ファジィ・ニューラルネットワーク
メイン制御器内の対象物操り用ファジィ・ニューラルネットワークは、力が与えられているにもかかわらず対象物が動かない場合は学習せず、対象物が運動しているときだけ学習するというものを除けば、基本的に既報⁽⁶⁾⁽⁷⁾で紹介したものと同じであるのでこの説明は簡単にする。

前件部のファジィ変数は5種類(*PB*, *PS*, *ZO*, *NS*, *NB*)とし、入力変数は目標位置との誤差量 E とその変化量 $\dot{E}$ の2種類とする。ファジィ変数は、式(2)で表されるシグモイド関数と式(4)で表されるガウシアン関数の両方を用いることにより単層のニューラルネットワークで前件部を構成するものとする。

$$f_s(u_s) = \frac{1}{1 + e^{-u_s}} \quad (2)$$

$$u_s(x) = w_0 + w_1 x \quad (3)$$

$$f_G(u_G) = e^{-u_G^2} \quad (4)$$

$$u_G(x) = \frac{w_0 + x}{w_1} \quad (5)$$

ここで、 w_0 はしきい値、 w_1 は結合荷重である。ガウシアン関数では、これらの値はそれぞれメンバシップ関数の中心値と偏差値を表す。詳細は既報⁽⁶⁾⁽⁷⁾を参照されたい。前述のとおりこの出力と摩擦補償用ニューロンからの出力を合せたものがメイン制御器からの出力である。

3・3 学習則 バックプロパゲーション法を用いて対象物操り用ファジィ・ニューラルネットワークの

Fig. 4 Learning algorithm of the object controller

前件部と後件部の両方のウェイト値の微調整、あるいは摩擦補償用ニューロンの学習を各サンプリングごとに行うものとする。評価関数は次式で示される対象物位置の二乗誤差値とする。

$$Y = \frac{1}{2} (x_d - x)^2 \quad (6)$$

ただし、対象物操り用ファジィ・ニューラルネットワークと摩擦補償用ニューロンは同時に学習することではなく、対象物が動いているときは対象物操り用ファジィ・ニューラルネットワークが学習し、力が与えられているにもかかわらず動かないときは摩擦補償用ニューロンが学習する(図4参照)。すなわち、対象物の位置情報をもとにして、力が与えられているにもかかわらず対象物が動かない場合は対象物操り用ファジィ・ニューラルネットワークの学習係数を零に、対象物が運動しているときは摩擦補償用ニューロンの学習係数を零にすることにより学習の切り換えが行われる。

対象物操り用ファジィ・ニューラルネットワークの前件部と後件部の各部分でのウェイト値の変化量は次のとおりである。

$$\text{後件部: } \Delta w_r = \eta (x_d - x) y_l y_k \quad (7)$$

$$\text{前件部: } \Delta w_i = \eta \sum_j \frac{\partial e_{jk}}{\partial y_j} f'(u) u' \quad (8)$$

ここで η は学習係数、 x_d は目標位置、 x は実際の対象物の位置、 y_j は前件部出力、 e_{jk} は前件部出力誤差、 $f'(u)$ と u' は式(2)、(4)で表されるアクティベーション関数とその入力の微分である。これにより、メンバシップ関数の定義や制御器からの出力が調整される。

摩擦補償用ニューロンの入力が1のときのウェイト値の変化量は次式で表される。

$$\Delta w_f = \eta (x_d - x) \quad (9)$$

入力が-1のときは式(9)の変化量に-1を掛けたものがウェイト値の変化量となる。

摩擦補償用ニューロンの学習は対象物が動いていないときだけ行われるので、評価関数が減少するように

学習するというよりは、対象物が動き出すまでウェイト値を大きな値に変更するものである。

メイン制御器内の対象物操り用ファジィ・ニューラルネットワークと摩擦補償用ニューロンのすべてのウェイト値は、それぞれ別個の学習係数により調整される。

4. 学 習 係 数

ニューラルネットワークのバックプロパゲーション法において学習係数は、学習速度や安定性にかかわる重要なファクタである。学習係数が固定されている一般的なバックプロパゲーション法では、誤差関数曲面のこう配が緩い場合学習速度が遅くなる。そこで、学習速度を上げるためには学習係数を大きくすればよいのだが、学習係数が大きすぎると不安定になる危険性がある。この問題を解決するため、これまでに適切な学習係数を見つけるための多くの研究がなされてきた⁽¹³⁾⁽¹⁴⁾。例えば、Jacobs は四つのヒューリスティックな学習係数調整法を挙げたうえで、ニューラルネットワークのすべてのウェイトがそれぞれ別個の学習係数をもつことを主張しており⁽¹³⁾、Arabshahi らは学習係数をファジィ制御で調整することを提案している⁽¹⁴⁾。学習係数をファジィ制御で調整する方法は効果的であるが、学習係数の有効範囲をあらかじめ把握しておく必要があると思われる。したがって、適切な学習係数の範囲が予測できない場合は、Jacobs が提案しているような、学習係数を線形的に増加させたり指数的に減少させたりする Delta-Bar-Delta 学習法のほうが効果的であると思われる。Delta-Bar-Delta 学習法では、時刻 t での学習係数の調整量 $\Delta\eta(t)$ は次に示すように定義される。

$$\Delta\eta(t) = \begin{cases} \kappa & \text{if } \bar{\delta}(t-1)\delta(t) > 0 \\ -\phi\eta(t) & \text{if } \bar{\delta}(t-1)\delta(t) < 0 \\ 0 & \text{otherwise} \end{cases} \quad \dots\dots\dots(10)$$

$$\delta(t) = \frac{\partial Y(t)}{\partial w(t)} \quad \dots\dots\dots(11)$$

$$\bar{\delta}(t) = (1-\theta)\delta(t) + \theta\delta(t-1) \quad \dots\dots\dots(12)$$

ここで、 κ と ϕ は定数、 $\delta(t)$ は時刻 t における評価関数 Y のウェイト w に対する偏微分、 $\bar{\delta}(t)$ は θ をベースとした現在と過去の微分の指数的平均である。しかし、このような Delta-Bar-Delta 学習法では学習係数が適切な値より小さかった場合、学習係数は常に一定の割合で増加するため、学習係数とその増加割合が適切な学習係数に比べ小さい場合は効率が悪いという問題がある。

本稿では、状況により学習を切替える制御器に加え、学習状況に応じ学習係数が最適値より小さい場合の増加割合を変化させる新しい Delta-Bar-Delta 学習法も提案し、対象物操り用のファジィ・ニューラルネットワークと摩擦補償用ニューロンの学習に利用する。新しい Delta-Bar-Delta 学習法では、学習係数が最適値よりも小さいとき、一定の割合で増加させるのではなく、ウェイトの変化分に対し出力誤差の変化分が小さいニューロンの学習係数は大きな割合で、ウェイトの変化分に対し出力誤差の変化分が大きいニューロンの学習係数は小さな割合で増加させる。すなわち、誤差関数曲面のこう配が緩いものには大きな割合で、誤差関数曲面のこう配が急なものには小さな割合で学習係数を増加させるものとする。ただし、学習係数を減少させる場合は既存の Delta-Bar-Delta 学習法と同様に指数的に減少させるものとする。したがって本稿では、新しい Delta-Bar-Delta 学習法として、学習状況により式(10)での κ の大きさを下記の例のように調整する方法を提案し、メイン制御器の対象物操り用ファジィ・ニューラルネットワークと摩擦補償用ニューロンの両方の学習係数の調整に利用した。

例：

$$\kappa = \begin{cases} 10\kappa & \text{if } 0.2 > \delta(t) \\ \kappa & \text{if } 0.2 \leq \delta(t) \leq 1.0 \\ 0.1\kappa & \text{if } 1.0 < \delta(t) \end{cases}$$

また、 κ の大きさを連続で変化させる方法も考えられる。例えば上の例で $\delta(t)$ が 0.2 より小さいときに、

$$\kappa = (20 - 95\delta(t))\kappa \quad \text{if } 0.2 > \delta(t)$$

というようにも設定可能である。

ただし、力が与えられているにもかかわらず対象物が動いていない場合は対象物操り用ファジィ・ニューラルネットワークの学習係数が零、対象物が運動しているときは摩擦補償用ニューロンの学習係数が零に設定される。

5. マニピュレータの制御

メイン制御器で出力された目標力どおりの力を各サブ制御器によりマニピュレータで発生できなければ対象物は予定どおり運動しない。したがって、目標力どおりの力(メイン制御器からの出力)を各マニピュレータで発生できるということが重要となる。そこで本稿では既報⁽⁶⁾⁽⁷⁾と同じく、ファジィ制御の方法を用いて適当な力制御器を設計し、それをニューラルネットワークでモデル化し、バックプロパゲーション法で学習するファジィ・ニューラルネットワーク力制御器⁽¹²⁾を利用する。ここでのマニピュレータの制御法について

は既報でも述べているので、本稿では簡単に説明するものとする。

5.1 ファジィ・ニューラルネットワーク力制御

通常ロボットマニピュレータのファジィ制御では、入力情報として誤差量とその変化量が考えられるが、力制御の場合は通常測定値にノイズが多く含まれているため誤差の変化量を利用することが難しい。そこで誤差の変化量の代わりにマニピュレータが環境を押す速度を利用することが考えられるが、マニピュレータが環境を押す速度が早い遅いのかの概念はマニピュレータの形態により異なる。つまりその形態での力制御方向の慣性行列の大きさにより速度の概念が変わることになる。そこで、著者らはマニピュレータの速度ではなく力制御方向のマニピュレータ運動量を入力情報として利用することを提案してきた⁽¹²⁾。対象物軌道制御の際の力制御は通常の力制御と違い環境自身が運動するので、運動量算出に使う速度は力制御方向のマニピュレータと対象物との相対速度を使用する。

サブ制御器からの出力は基準座標系でのマニピュレータへの力指令とする。したがって、図1のモデルで X 方向と Y 方向の2方向の力制御器が必要である。

サブ制御器のファジィ・ニューラルネットワークの学習則は、次に示す評価関数以外前項のメイン制御器のニューラルネットワークの学習則と同じである。

$$Y = \frac{1}{2}(f_d - f)^2 \quad \text{..... (13)}$$

ただし、サブ制御器の学習係数の調整は行わず、一定の定数を使用した。

5.2 標準環境モデルによる補償 本サブ制御器において対象物の動特性を設計時に固めに想定しておけば、ほとんどの場合オーバーシュートすることもなく目標どおりの力を制御できる。しかし、もし対象物の動特性が想定していたものよりかなり固かった場合は大きなオーバーシュートが発生する等の問題が生じる。そこでサブ制御器内に設計時に想定した環境モデルを置き、環境が想定したものよりも固かった場合に誤差のメンバシップ関数調整を行う方法を利用する⁽⁶⁾⁽⁷⁾。

Fig. 5 Environment model

環境を図5に示すようにモデル化すると、環境からの反力つまりマニピュレータにより加えられた力 F は、環境表面の変位を x_e とすると次式のように表される。

$$F = M_e \ddot{x}_e + D_e \dot{x}_e + K_e x_e \quad \text{..... (14)}$$

ファジィ制御則の設計時に標準となる M_e , D_e , K_e の各係数を想定し、サブ制御器内で式(14)を各サンプリングごとに計算することにより設計時に想定した環境での環境反力がわかる。もし想定した環境よりも実際に測定した反力が固かった場合は、オーバーシュティングなどの問題を抑制するため、環境の固さに応じ制御器への入力情報である目標値との誤差値に補正係数をかけてやることにより補償を行う。補正係数 H の算出式は次のとおりである。

$$H = 1/(kF_m/F_{est}) \quad \text{..... (15)}$$

ここで、 k は補正のための経験値 (≥ 1)、 F_m は実際に測定された力、 F_{est} は標準環境モデルで算出した力である。この補正係数 H をかけてやることにより、設計時に定義されたメンバシップ関数を環境の固さに応じ横に広げてやったような形状に変更するのと同じ効果が得られる。この効果により実際にはまだ誤差が大きい時点で誤差が小さいときのようにきめ細かく制御されるようになる。

6. シミュレーション

本制御器の有効性を評価するため、図1に示したモデルを用いコンピュータによるシミュレーションを行った。対象物の質量は8 kg とし、 X 方向と Y 方向へ同時に目標軌道どおりに並進運動するよう制御されるものとする。また、対象物には X 方向 Y 方向ともに10 N のクーロン摩擦が作用するものとする。シミュレーションでのサンプリング時間はすべて1 ms とした。既報⁽⁶⁾⁽⁷⁾においてすでにサブ制御器により目標どおりの力が得られることを確認しているため、本稿ではサブ制御器の目標力に対する追従性のシミュレーション結果は省略するものとする。

まず最初に、メイン制御器において、提案した新しい学習係数調整法の効果を確認するため、摩擦補償用ニューロンによるクーロン摩擦補償なしの場合について、従来のDelta-Bar-Delta 学習係数調整法を用いた対象物滑らせ移動制御の結果と本稿で提案した新しいDelta-Bar-Delta 学習係数調整法を用いた対象物滑らせ移動制御の結果とで比較を行った。マニピュレータの可動範囲を考え、対象物の目標軌道は X 方向で $0.05 \sin(0.25 \pi t)$ m, Y 方向で $0.025 \sin(0.25 \pi t)$ m とした。また、対象物の動特性は設計時に想定した

 Fig. 6 Simulation result-effect of modified Delta Bar Delta rule (X -direction position)

 Fig. 8 Simulation result-effect of the neuron for friction compensation (X -direction position)

 Fig. 7 Simulation result effect of modified Delta Bar Delta rule (Y -direction position)

 Fig. 9 Simulation result-effect of the neuron for friction compensation (Y -direction position)

モデルと同じものにした。このときの X 方向のシミュレーション結果を図 6 に、 Y 方向のシミュレーション結果を図 7 に示す。これらの結果より、飛躍的な向上は見られないものの、新しい学習係数調整法を用いたほうがより滑らかに誤差も少なく目標軌道に近づいていくことが確認できる。

次に、メイン制御器における摩擦補償用ニューロンによるクーロン摩擦補償効果を確認するための対象物滑らせ軌道制御シミュレーションを行った。このとき、対象物操り用ファジィ・ニューラルネットワークと摩擦補償用ニューロンのバックプロパゲーション法による学習の学習係数の調整は、学習係数が零に切換えられているとき以外、本稿で提案した新しい Delta-Bar-Delta 学習法により調整した。対象物の目標軌道は最初のシミュレーションのときと同じく X 方向で $0.05 \sin(0.25 \pi t)$ m、 Y 方向で $0.025 \sin(0.25 \pi t)$ m とした。このときの X 方向のシミュレーション結果は図 8 に、 Y 方向のシミュレーション結果は図 9 に示されている。ここでは、制御開始後約 0.3 秒間位摩擦補償用ニューロンが学習し、その後は対象物が運動し始め

るため、対象物操り用ファジィ・ニューラルネットワークが学習している。対象物操り用ファジィ・ニューラルネットワークが学習を始めるときには摩擦補償用ニューロンによる摩擦補償の学習がほぼ完了しているため、その後の対象物の動きは滑らくなる。摩擦補償用ニューロンによる摩擦補償量が多少多すぎたり少なすぎたりしても、対象物操り用ファジィ・ニューラルネットワークの適応能力において微調整が行われる。これらの結果より摩擦補償用ニューロンによるクーロン摩擦補償により、図 6 と 7 で見られたような振動もすることなく、制御開始後まもなくから滑らかに目標軌道に追従することがわかる。ただし、 X 方向と Y 方向では目標移動距離が異なり、摩擦補償用ニューロンの学習の際の条件が異なる(対象物操り用ファジィ・ニューラルネットワーク部からの出力が異なる)ため、若干の傾向の違いが見られる。

7. ま と め

複腕ロボットによる対象物の滑らせ移動制御において、対象物操り用ファジィ・ニューラルネットワーク

と摩擦補償用ニューロンを組合せることによりクーロン摩擦が働いていても滑らかに対象物を運動させる方法を提案した。対象物操り用ファジィ・ニューラルネットワークは対象物が運動しているときだけ学習し、摩擦補償用ニューロンは力が与えられているにもかかわらず対象物が動かない場合だけ学習するという新しい方法を示した。また、ニューラルネットワークの学習係数を適切な値に調整する新しいDelta-Bar-Delta 学習法を提案した。

これらの方法により対象物を目標軌道どおり滑らかに滑らせ移動制御できることを説明し、コンピュータシミュレーションによりその有効性を確認した。

文 献

- (1) Nakano, E., Ozaki, S., Ishida, T. and Kato, I., Cooperative Control of the Anthropomorphic Manipulator MELARM, *Proc. of 4th International Symposium on Industrial Robots*, (1974), 251-260.
- (2) 内山勝・ドシェビエール, 両手ロボットの対称型運動学と非マスタスレーブ協調制御, 日本ロボット学会誌, 7-1(1989), 19-30.
- (3) 中村仁彦・永井清・吉川恒夫, 複数のロボット機構による協調的あやつりの力学, 日本ロボット学会誌, 4-5(1986), 489-498.
- (4) 小菅一弘・吉田英博・福田敏男・酒井勝・蟹谷清・針木和夫, 複数のロボットの分散協調制御, 第11回日本ロボット学会学術講演会予稿集, (1993), 705-706.
- (5) Schneider, S. A. and Cannon, R. H. Jr., Object Impedance Control for Cooperative Manipulation: Theory and Experimental Results, *IEEE Trans. on Robotics and Automation*, 8-3(1992), 383-394.
- (6) 木口景夫・福田敏男, ファジィ・ニューラルネットワークを利用した複腕ロボットによる未知対象物の制御, 機論, 61-589, C(1995), 3643-3649.
- (7) Kiguchi, K. and Fukuda, T., Object Manipulation Under Disturbance Using Fuzzy-Neural Network, *Mathematics and Computers in Simulation* (To be published), (1996).
- (8) Kerr, J. and Roth, B., Analysis of Multifingered Hands, *Int. J. Robotic Research*, 4-4(1986), 3-17.
- (9) 堀川慎一・古橋武・大熊繁・内川嘉樹, ニューラルネットワークによる学習型ファジィ制御器, 計測自動制御学会論文集, 27-2(1991), 208-215.
- (10) 湯軍・渡辺桂吾・中村政俊・古賀信之, ファジィ・ガウシアン・ニューラルネットワーク制御器とその移動ロボット車の制御への応用, 機論, 59-564, C(1993), 2290-2297.
- (12) Kiguchi, K. and Fukuda, T., Robot Manipulator Contact Force Control Application of Fuzzy-Neural Network, *Proc. of IEEE Int. Conf. on Robotics and Autom.*, (1995), 875-880.
- (13) Jacobs, R. A., Increased Rates of Convergence Through Learning Rate Adaptation, *Neural Networks*, (1988), 1, 295-307.
- (14) Arabshahi, P., Choi, J. J., Marks, R. J. II, Caudell, T. P., Fuzzy Control of Backpropagation, *Proc. of IEEE Int. Conf. on Fuzzy Systems*, (1992), 967-972.
- (15) 福田敏男・柴田崇徳・鶴田正俊・光岡豊一, 神経回路モデルによるロボットマニピュレータの位置・力制御の研究(第3報, ファジィルールを適用したニューラルネットワークの位置と力のハイブリッド制御への応用), 機論, 57-535, C(1991), 874-881.