

ニューラルネットによるインピーダンスパラメータの繰返し学習†

辻 敏夫*・西田 正孝*・伊藤 宏司**

Iterative Learning of Impedance Parameters for Manipulator Control Using Neural Networks

Toshio TSUJI*, Masataka NISHIDA* and Koji ITO**

Impedance Control is one of the most effective control methods for the manipulators in contact with their environments. The characteristics of force and motion control, however, are determined by the desired impedance of end-effectors which must be designed according to the given tasks. In the present paper, we propose a method to regulate impedance parameters of the manipulator's end-effector using neural networks. Three kinds of the Back Propagation typed neural networks are prepared corresponding to position, velocity and force control of the end-effector. Firstly, the neural networks for position and velocity control are trained using iterative learning of the manipulator during free movements. Then, the neural network for force control is trained for contact movements. During the contact movement, the virtual trajectory of the end-effector is modified to reduce the force control error. Computer simulations shows that the method can regulate not only stiffness and viscosity but also inertia of the end-effector, and can realize smooth transition from free to contact movements by regulating impedance parameters before contact.

Key Words: impedance control, robot manipulator, neural networks, trajectory control, force control, iterative learning, virtual trajectory

1. ま え が き

対象物に接触して作業を行う場合、対象物からの拘束にしたがって位置だけでなく力の制御が必要となる。インピーダンス制御は、対象物からみたマニピュレータの手先の動特性を望ましい値に設定しようとするもので、位置と力の制御を必要とする接触作業に有効である¹⁾。しかしながら、このためには作業内容、目的、対象物の特性に応じて、適切なインピーダンスを設定する必要がある、その設定法を一般的に記述することは容易ではない。

この目標インピーダンスの設定法として、たとえば羅らは、制御理論の立場から、環境のダイナミクスを考慮

したモデルマッチング法を提案している²⁾。これは、まず自由空間内における手先の追値制御特性を設計し、その上で接触空間内でのコンプライアンス性を保つように規範モデルを選択するというものである。しかしながら、この方法では適切な規範モデルの選択が必要であり、また、環境のパラメータがある程度既知でなければならない。一方、松尾らは、接触状態におけるコンプライアンスの満たすべき条件を線形制約条件として定式化し、適当な目的関数を与えることでコンプライアンス設定問題を線形計画問題に帰着させるという方法を提案している³⁾。この方法では、最適なコンプライアンスを求めることが可能であるが、線形計画法を用いているため線形で時不変なコンプライアンスしか求めることができない。また、剛性、慣性を含めたインピーダンスの設定に拡張することも難しい。これに対して本稿では、ニューラルネットを用いて学習的に目標インピーダンスを設定する方法について考察する。

最近、ニューラルネットをインピーダンス制御に適用した研究がいくつか報告されている^{4)~6)}。たとえば Gomi

† 第1回インテリジェントシステムシンポジウムで発表 (1991・10)

* 広島大学工学部 東広島市鏡山 1-4-1

** 豊橋技術科学大学 豊橋市天伯町雲雀ヶ丘 1-1

Faculty of Engineering, Hiroshima University,
Higashi-Hiroshima

** Toyohashi University of Technology, Toyohashi
(Received April 9, 1992)
(Revised July 17, 1992)

らはインピーダンス制御系の非線形補償にニューラルネットワークによるフィードバック誤差学習を適用し、モデル化や同定困難な制御対象のインピーダンス制御を可能にしている⁴⁾。この方法は、ニューラルネットワークを制御対象の補償器として用いたもので、インピーダンスパラメータそのものを学習しているわけではない。一方、Asada は、力フィードバック制御器にニューラルネットワークを用いることにより、非線形なコンプライアンス(粘性のみ)を学習的に実現できることを示した⁵⁾。また Cohen らは、手先の剛性行列、粘性行列をニューラルネットワークで表現し、速度と力の評価に基づいて学習的にインピーダンスパラメータを調節する方法を提案している⁶⁾。彼らは、インピーダンスだけでなく、目標速度を変化させることにより作業特性を向上できることを示しているが、慣性パラメータは扱っておらず、また対象とする運動を拘束運動のみに限定している。

以上の従来の研究に対して、本稿では慣性、粘性、剛性のすべてのインピーダンスパラメータを調節することができ、また、自由運動と拘束運動をともに取り扱うことが可能な手先インピーダンス調節法を提案する。この方法は、ニューラルネットワークによりインピーダンスパラメータを学習的に設定できるだけでなく、目標軌道を積極的に修正することで、自由運動から拘束運動への滑らかな移行を実現することができる。以下、2章でインピーダンス制御の定式化を行い、3章で学習による手先インピーダンスの調節法について説明する。4章ではシミュレーション実験により本手法の有効性を検討する。

2. インピーダンス制御

一般に対象物に接触しているマニピュレータ(関節自由度 n) の運動方程式は、

$$M(\theta)\ddot{\theta} + h(\theta, \dot{\theta}) + g(\theta) = \tau - J^T(\theta)F_{int} \quad (1)$$

と表わされる。ただし、 $M(\theta) \in R^{n \times n}$ は慣性行列、 $h(\theta, \dot{\theta}) \in R^n$ はコリオリ力・遠心力、 $g(\theta) \in R^n$ は重力トルク、 $\theta \in R^n$ は関節角度、 $\tau \in R^n$ は関節トルク、 $J(\theta) \in R^{m \times n}$ はヤコビ行列、 $F_{int} \in R^m$ は対象物からマニピュレータの手先に作用する力で、 m は作業空間の自由度、 $(\)^T$ は転置行列を表わす。また、 F_{int} はつぎのようにモデル化する。

$$F_{int} = M_e \ddot{X} + B_e \dot{X} + K_e(X - X_e) \quad (2)$$

ここで、 $M_e, B_e, K_e \in R^{m \times m}$ はそれぞれ対象物の慣性、粘性、剛性、 $X \in R^m$ はマニピュレータの手先位置、 $X_e \in R^m$ は対象物の平衡点での位置を表わす。

(1)式に対して、

$$\tau = h(\theta, \dot{\theta}) + g(\theta) + J^T(\theta)F_{int} + M(\theta)J^{-1}(\theta)[F_{act} - \ddot{X}] \quad (3)$$

のような非線形補償を行うとマニピュレータのダイナミ

クスはつぎのように単純化される²⁾。

$$\ddot{X} = F_{act} \quad (4)$$

ただし、 $F_{act} \in R^m$ は作業空間で表現した制御入力である。

一方、マニピュレータの目標手先インピーダンスは、作業空間上で

$$M_d \ddot{X} + B_d \dot{X} + K_d X = F_d - F_{int} \quad (5)$$

と記述されるものとする¹⁾。ただし、 $M_d, B_d, K_d \in R^{m \times m}$ はそれぞれ、目標慣性行列、目標粘性行列、目標剛性行列、 $dX = X - X_d$ で、 $X_d \in R^m$ は目標軌道(本稿では仮想軌道とも呼ぶ)、 $F_d \in R^m$ は目標手先力を表わす。このとき、目標インピーダンス((5)式)を実現する制御入力 F_{act} は、

$$F_{act} = -M_d^{-1}(B_d \dot{X} + K_d X) + M_d^{-1}(F_d - F_{int}) + \ddot{X}_d \quad (6)$$

となる。

Fig. 1 にインピーダンス制御系のブロック線図を示す。マニピュレータが対象物と接触していない自由運動では、 $F_d = F_{int} = 0$ であるから、図中の力制御ループは働かず、位置・速度制御のみが機能する。一方、対象物と接触している場合は、力制御が同時に行われ、位置と力の制御を統一的に扱うことが可能となる。しかしながら、自由運動時の手先の追値特性と拘束運動時の力制御特性は、インピーダンスパラメータ(M_d, B_d, K_d)によって決定される。そこで本稿では、ニューラルネットワークを用いて学習的にこれらを調節することを考える。

3. インピーダンスパラメータの繰返し学習

3.1 インピーダンス学習制御系の構成

Fig. 2 にニューラルネットワークを組み込んだインピーダンス学習制御系の構成を示す。ニューラルネットワークは位置・速度制御を行う NNTC (Neural Networks for Trajectory Control) と力制御を行う NNFC (Neural Networks for Force Control) の二つの部分からなる。NNTC は、インピーダンスパラメータ $M_d^{-1}K_d$ と $M_d^{-1}B_d$ に対応し、NNFC は、 M_d^{-1} に対応している (Fig. 1 参照)。

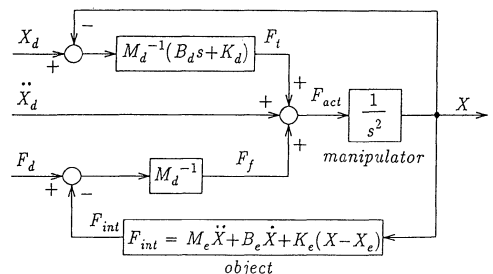


Fig. 1 Impedance control system in the operational space

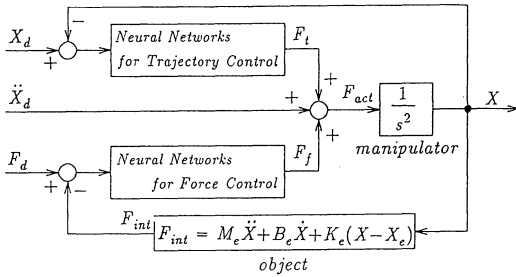


Fig. 2 Impedance control system using neural networks

ニューラルネットの学習は、つぎのような手順で行う。まず、マニピュレータを自由空間で動作させ、手先の追従誤差を減少させるように **NNTC** を学習する。そして、**NNTC** の学習が終了した後、マニピュレータに接触作業を行わせる。このとき、**NNTC** は固定し、手先の力制御誤差を減少させるように **NNFC** および目標軌道 X_d , \dot{X}_d を修正する。**NNFC** だけでなく、目標軌道の修正を行うことにより、より効果的な学習が期待できる。

3.2 自由運動の学習

Fig. 3 に **NNTC** の構成を示す。**NNTC** は、位置制御を行う **NNPC** (Neural Networks for Position Control) と、速度制御を行う **NNVC** (Neural Networks for Velocity Control) の二つの部分に細分化される。

各ニューラルネットは、マニピュレータの手先位置 $X \in R^m$ を入力とし、インピーダンスパラメータ $M_d^{-1}K_d \in R^{m \times m}$ (**NNPC**), $M_d^{-1}B_d \in R^{m \times m}$ (**NNVC**) を出力とする多層型ニューラルネットである。ただし、入力ユニットには線形出力関数を、中間層、出力層にはシグモイド型の関数を用いる。すなわち、各ユニットの入出力を x_i, y_i とすると、

$$x_i = \begin{cases} I_i & (\text{入力層}) \\ \sum y_j w_{ij} & (\text{中間層, 出力層}) \end{cases} \quad (7)$$

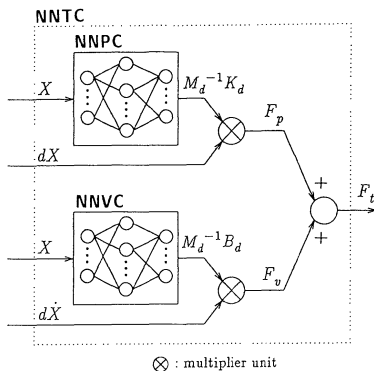


Fig. 3 Neural network for trajectory control

$$y_i = \begin{cases} x_i & (\text{入力層}) \\ \frac{1}{1 + e^{-x_i}} & (\text{中間層}) \\ \frac{U}{1 + e^{-(x_i + \theta)}} & (\text{出力層}) \end{cases} \quad (8)$$

である。ただし、 w_{ij} はユニット j からユニット i への重み係数、 U, θ は正の定数で、それぞれ、ネットワークの出力の最大値、しきい値を表わす。

ここで、ネットワークの出力を

$$O_p = (o_{p1}, o_{p2}, \dots, o_{pm})^T \quad (9)$$

$$O_v = (o_{v1}, o_{v2}, \dots, o_{vm})^T \quad (10)$$

のように表現する。ただし、 $o_{pi}, o_{vi} \in R^m$ はそれぞれ $M_d^{-1}K_d, M_d^{-1}B_d$ の第 i 行に対応したベクトルで、 $O_p, O_v \in R^{m^2}$ はそれぞれ **NNPC**, **NNVC** の出力をたてに並べたベクトルである。これにより、自由運動時のマニピュレータの制御入力 F_{act} は、

$$\begin{aligned} F_{act} &= F_t + \dot{X}_d = F_p + F_v + \dot{X}_d \\ &= - \begin{bmatrix} o_{p1} \\ o_{p2} \\ \dots \\ o_{pm} \end{bmatrix} dX - \begin{bmatrix} o_{v1} \\ o_{v2} \\ \dots \\ o_{vm} \end{bmatrix} d\dot{X} + \dot{X}_d \end{aligned} \quad (11)$$

となる。ここで、 $F_p, F_v \in R^m$ はそれぞれ、**NNPC**, **NNVC** から計算された制御入力である。

ネットワークの学習は、誤差関数 E_1

$$E_1 = \frac{1}{2} \sum_{t=0}^{N_f} \{E_p(t) + E_v(t)\} \quad (12)$$

を減少させるように行う。ただし、 $E_p(t) = (X_d(t) - X(t))^T (X_d(t) - X(t))$, $E_v(t) = (\dot{X}_d(t) - \dot{X}(t))^T (\dot{X}_d(t) - \dot{X}(t))$ で、 $N_f = t_f / \Delta t$ はデータ数を表わす。 t_f は制御の終了時刻、 Δt はサンプリングタイムである。このとき **NNPC**, **NNVC** の重み係数 $w_{ij}^{(p)}, w_{ij}^{(v)}$ を誤差の最急降下方向に修正する。

$$\Delta w_{ij}^{(p)} = -\eta_p \frac{\partial E_1}{\partial w_{ij}^{(p)}}, \Delta w_{ij}^{(v)} = -\eta_v \frac{\partial E_1}{\partial w_{ij}^{(v)}} \quad (13)$$

$$\frac{\partial E_1}{\partial w_{ij}^{(p)}} = \frac{\partial E_1}{\partial F_p(t)} \frac{\partial F_p(t)}{\partial o_{pi}(t)} \frac{\partial o_{pi}(t)}{\partial w_{ij}^{(p)}} \quad (14)$$

$$\frac{\partial E_1}{\partial w_{ij}^{(v)}} = \frac{\partial E_1}{\partial F_v(t)} \frac{\partial F_v(t)}{\partial o_{vi}(t)} \frac{\partial o_{vi}(t)}{\partial w_{ij}^{(v)}} \quad (15)$$

η_p, η_v はニューラルネットの学習率を表わす。ここで、

$$\frac{\partial F_p(t)}{\partial o_{pi}(t)}, \frac{\partial F_v(t)}{\partial o_{vi}(t)} \text{ は (11) 式から計算でき、} \frac{\partial o_{pi}(t)}{\partial w_{ij}^{(p)}},$$

$$\frac{\partial o_{vi}(t)}{\partial w_{ij}^{(v)}} \text{ は誤差逆伝播法により計算することができる}$$

$$\text{が、} \frac{\partial E_1}{\partial F_p(t)}, \frac{\partial E_1}{\partial F_v(t)} \text{ はマニピュレータのダイナミクスが}$$

$$\text{存在するために直接計算することができない。そこで本稿ではマニピュレータを繰返し動作させ、betterment}$$

$$\text{process}^7) \text{を用いて } \frac{\partial E_1}{\partial F_p(t)}, \frac{\partial E_1}{\partial F_v(t)} \text{ を近似することにする}$$

る。

betterment process とは、前回のシステムの出力を用いて次の入力繰り返し修正することにより、システムの出力を目標出力に近づけていく方法である⁷⁾。なお、マニピュレータの非線形補償を行っていない場合、システムは非線形となり betterment process の適用は難しい。この場合は広義ニュートン法^{8),9)}などを用いればよい。DP 型の betterment process を用いると、 $k+1$ 回目の試行時の制御入力 $F_{act}^{k+1}(t)$ は k 回目の試行時の制御入力 $F_p^k(t)$, $F_v^k(t)$ より、

$$F_{act}^{k+1}(t) = F_p^{k+1}(t) + F_v^{k+1}(t) + \dot{X}_d(t) \quad (16)$$

$$F_p^{k+1}(t) = F_p^k(t) - \Gamma_p dX^k(t) \quad (17)$$

$$F_v^{k+1}(t) = F_v^k(t) - \Gamma_v d\dot{X}^k(t) \quad (18)$$

のように書くことができる。ただし、 $\Gamma_p, \Gamma_v \in R^{m \times m}$ はそれぞれ、位置、速度誤差の修正ゲインである。適切な修正ゲイン Γ_p, Γ_v を用いる限り、誤差関数 E_1 は指数関数的に減少していくことが保証されている⁷⁾。

ここで、(17), (18)式に着目すると、いずれも右辺第 2 項が誤差関数 E_1 を時刻 t において減少させる方向を与えていることに気付く、そこで、

$$\begin{aligned} \frac{\partial E_1}{\partial F_p(t)} &\equiv [\Gamma_p dX^k(t)]^T, \\ \frac{\partial E_1}{\partial F_v(t)} &\equiv [\Gamma_v d\dot{X}^k(t)]^T \end{aligned} \quad (19)$$

を採用し、誤差の最急降下方向((13)式)を計算することにする。以上の方法をまとめると、自由運動時の学習法は以下になる。

[step 1] インピーダンス制御: NNPC, NNVC の重み係数 $w_p^{(p)}, w_v^{(p)}$ の初期値を設定し、与えられた目標軌道に対してインピーダンス制御を 1 試行行う。

[step 2] betterment process: step 1 で生じた位置、速度誤差から (17), (18)式を用いて $F_p^{k+1}(t)$, $F_v^{k+1}(t)$ を計算する。そして、この制御入力 $F_{act}^{k+1}(t)$ に対するシステムの出力誤差 $dX(t)$, $d\dot{X}(t)$ を記憶しておく。

[step 3] NNPC, NNVC の学習: step 2 で求めた $F_p^{k+1}(t)$, $F_v^{k+1}(t)$, $dX(t)$, $d\dot{X}(t)$ から (13), (14), (15), (19)式を用いてニューラルネットを学習し、学習誤差が極小となるまで重み係数 $w_p^{(p)}, w_v^{(p)}$ を更新する。

以上の手続きを誤差関数 E_1 ((12)式)が最小となるまで繰り返し適用する。学習が終了したとき、NNPC, NNVC はそれぞれ最適なインピーダンスパラメータ $M_d^{-1}K_d$, $M_d^{-1}B_d$ をネットワークの出力 $O_p(t)$, $O_v(t)$ として表現していることになる。

3.3 拘束運動の学習

NNTC の学習終了後、マニピュレータに接触作業を行わせ、力制御の学習を行う。ただし、このとき NNTC の

重み係数は固定しておく。Fig. 4 に NNFC の構成を示す。

ニューラルネットは、NNPC, NNVC と同一構造の多層型ニューラルネットで、マニピュレータの手先位置 $X(t) \in R^m$ および、力制御誤差 $dF(t) \equiv F_d(t) - F_{int}(t)$, $dF(t-1) \in R^m$ を入力とし、インピーダンスパラメータ M_d^{-1} を出力する。 dF は接触情報をニューラルネットに与えるための入力である。各ユニットの入出力は、(7), (8)式と同様とする。

ここで、ネットワークの出力を NNFC と同様に、

$$O_f = (o_{f1}, o_{f2}, \dots, o_{fm})^T \quad (20)$$

と表現する。ただし、 $o_{fi} \in R^m$ は M_d^{-1} の第 i 行に対応したベクトルで、 $O_f \in R^{m^2}$ は NNFC の出力をたてに並べたベクトルである。このとき、(6), (11)式から拘束運動時のマニピュレータの制御入力 F_{act} は、

$$\begin{aligned} F_{act} &= F_t + F_f + \dot{X}_d \\ &= - \begin{bmatrix} o_{p1} \\ o_{p2} \\ \vdots \\ o_{pm} \end{bmatrix} dX - \begin{bmatrix} o_{v1} \\ o_{v2} \\ \vdots \\ o_{vm} \end{bmatrix} d\dot{X} + \begin{bmatrix} o_{f1} \\ o_{f2} \\ \vdots \\ o_{fm} \end{bmatrix} dF + \dot{X}_d \end{aligned} \quad (21)$$

となる。ただし、 $F_f \in R^m$ は、NNFC から計算された制御入力を表す。

NNFC の学習は、評価関数 E_2

$$\begin{aligned} E_2 &= \frac{1}{2} \sum_{t=0}^{N_f} \tilde{E}_2(t) \\ &= \frac{1}{2} \sum_{t=0}^{N_f} \left\{ \frac{1}{2} \sum_{i=N}^N h^2(i) [E_p(t+i) + E_f(t+i)] \right\} \end{aligned} \quad (22)$$

を減少させるように行う。ここで、

$E_f(t+i) = dF^T(t+i)dF(t+i)$ で、 $t > N_f$ および $t < 0$ のとき $E_p(t) = E_f(t) = 0$ とする。また、 $h(i)$ は窓関数¹⁰⁾である。Fig. 5 に窓関数の一例を示す。

時刻 t での誤差関数 $\tilde{E}_2(t)$ は、時刻 $t-N$ から $t+N$ までの位置誤差、力誤差の重み付き 2 乗和を与える。したがって、もし (22) 式の誤差関数 E_2 を最小(または極小)化できれば、 N 時刻までの未来の誤差を、あらかじめ

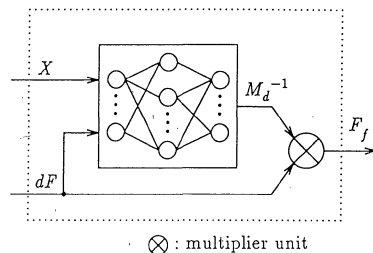


Fig. 4 Neural network for force control

時刻 t で考慮することが可能になる。このことは、自由運動から拘束運動への切り換え時などに有効である。本稿では繰返し制御を採用して学習を行っているため、 k 回目の繰返し試行時の時刻 t 以後の誤差を、 $k+1$ 回目の試行時に利用することができる。

NNFC の重み係数 w_{ki}^f の誤差 E_2 に対する最急降下方向は、

$$\Delta w_{ki}^f = -\eta_f \frac{\partial E_2}{\partial w_{ki}^f} \quad (23)$$

となる。 η_f は学習率である。まず、(22)式を変形して、 w_{ki}^f で偏微分すると、

$$\frac{\partial E_2}{\partial w_{ki}^f} = \sum_{i=-N}^N h^2(i) \frac{\partial E_3}{\partial F_{act}(t+i)} \frac{\partial F_{act}(t+i)}{\partial O_f(t+i)} \frac{\partial O_f(t+i)}{\partial w_{ki}^f} \quad (24)$$

を得る。ただし、 $E_3 = \frac{1}{2} \sum_{i=0}^{N_f} [E_p(t+i) + E_f(t+i)]$ とおいた。

(24)式の $\frac{\partial F_{act}(t+i)}{\partial O_f(t+i)}$ は(21)式から計算でき、

$\frac{\partial O_f(t+i)}{\partial w_{ki}^f}$ は誤差逆伝播法により計算できるが、

$\frac{\partial E_3}{\partial F_{act}(t+i)}$ は直接計算できない。そこで、NNTCと同様に betterment process を適用することを考える。

いま、窓関数の長さ N がデータ長 N_f に比べて十分短いと仮定しよう。このとき、

$$\begin{aligned} & \frac{1}{2} \sum_{i=0}^{N_f} [E_p(t+i) + E_f(t+i)] \\ & \approx \frac{1}{2} \sum_{i=-N}^N [E_p(t+i) + E_f(t+i)] \end{aligned} \quad (25)$$

と近似できるから、(25)式の右辺に対する最急降下方向を betterment process により計算することが可能となる。3.2節と同様に、DP型の betterment process を用いると、 $k+1$ 回目の試行時に対して、

$$\begin{aligned} \frac{\partial E_3}{\partial F_{act}(t+i)} & \approx [\Gamma_p dX^k(t+i) - \{\Gamma_f dF^k(t+i) \\ & + \Phi_f d\dot{F}^k(t+i)\}]^T \end{aligned} \quad (26)$$

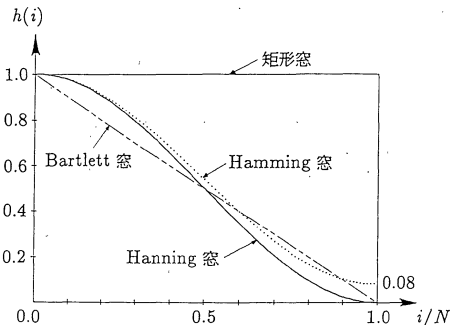


Fig. 5 Examples of the window functions $h(i)^{(10)}$

を得る。 $\Gamma_p, \Gamma_f, \Phi_f \in R^{m \times m}$ は betterment process の修正ゲインである。これにより誤差の最急降下方向(23)式を計算することが可能となる。

しかしながら、(26)式には力誤差の微分 $dF(t+i)$ が含まれており、NNFCのみでこれを学習するのは非常に困難である。そこで、(26)式から求めた誤差を NNFCだけでなく、仮想軌道へも誤差逆伝播することを考える。これは、(14)、(15)式のように誤差の修正量を NNTCの学習に用いるのではなく、制御系に対する入力である $X_d(t)$ 、 $\dot{X}_d(t)$ の修正に用いることを意味する。

仮想軌道の修正則は NNFC とほぼ同様に導出できるが、 $X_d(t)$ の修正を行うので(22)式に含まれる位置誤差 $E_p(t+i)$ は意味をもたない。そこで、位置誤差を評価関数から除いた

$$E_4 = \sum_{i=-N}^N \left\{ \frac{1}{2} \sum_{i=-N}^N h^2(i) E_f(t+i) \right\} \quad (27)$$

を用いる。このとき、誤差の最急降下方向は、

$$\Delta X_d(t) = -\eta_d \left[\frac{\partial E_4}{\partial X_d(t)} \right]^T \quad (28)$$

$$\begin{aligned} \frac{\partial E_4}{\partial X_d(t)} & = \sum_{i=-N}^N h^2(i) \frac{\partial E_5}{\partial F_{act}(t+i)} \frac{\partial F_{act}(t+i)}{\partial X_d(t+i)} \\ & \times \frac{\partial X_d(t+i)}{\partial x_d(t)} \end{aligned} \quad (29)$$

で与えられる。 $E_5 = \frac{1}{2} \sum_{i=0}^{N_f} E_f(t+i)$ である。ここで、誤差関数(27)式は、時刻 $t-N$ から $t+N$ までの力制御誤差の重み付き 2 乗和を時刻 t で評価しているの、 $X_d(t)$ の修正量はこの区間でほぼ一定と考えることができる。

そこで、 $\frac{\partial X_d(t+i)}{\partial x_d(t)} = I_m$ (m 次の単位行列)と仮定すると、

$\frac{\partial E_5}{\partial F_{act}(t+i)}$ は betterment process を用いて計算でき、

$\frac{\partial F_{act}(t+i)}{\partial X_d(t+i)}$ は(21)式から計算できるから、(28)式を用いて $X_d(t)$ を修正することが可能となる。 $\dot{X}_d(t)$ についても同様に学習則を導出できる。

以上をまとめると、拘束運動時の学習法は以下のようになる。

[step 1] インピーダンス制御：NNFC の重み係数 w_{ki}^f の初期値を設定し、初期目標軌道と学習後の NNTC を用いてインピーダンス制御を 1 試行行う。

[step 2] betterment process：step 1 で生じた誤差から betterment process を実行し、 $F_{act}^{k+1}(t)$ を計算する。そして、この制御入力に対するシステムの出力誤差 $dX(t)$ 、 $dF(t)$ 、 $d\dot{F}(t)$ を記憶しておく。

[step 3] NNFC、仮想軌道の学習：step 2 で求めた出力誤差から(23)、(28)式を用いて学習誤差が極小となるまで重み係数 w_{ki}^f 、仮想軌道 $X_d(t)$ 、 $\dot{X}_d(t)$ を更新する。

以上の手続きを誤差関数 E_2 ((22)式)が最小となるまで繰り返し適用する。学習が終了したとき、NNFC は最適なインピーダンスパラメータ M_d^{-1} をネットワークの出力 O_f として表現していることになる。

4. シミュレーション実験

本手法の有効性を検討するため、2 次元運動のシミュレーション実験を行った ($m=2$)。ただし、マニピュレータの手先のダイナミクスは (3) 式のように単純化されていると仮定する。

4.1 自由運動

まず、自由運動時の追従特性が学習によってどのように変化するか調べた。用いたニューラルネットは、NNPC, NNVC とも、入力ユニット 2, 中間ユニット 10, 出力ユニット 4 の 3 層ネットワークで、 $\theta=6$, $U=500$ ((8) 式) とした。重み行列 w_i^p , w_i^v の初期値は $|w_i^p|$, $|w_i^v| < 0.25$ の一様乱数で、学習率 $\eta_p=2.0 \times 10^{-4}$, $\eta_v=5.0 \times 10^{-5}$, betterment process の誤差修正ゲイン Γ_p , Γ_v ((19) 式) は、文献 7) を参考にして

$$\Gamma_p = \frac{1}{2} \text{diag}[\min(o_{p11}(t)), \min(o_{p22}(t))] \quad (30)$$

$$\Gamma_v = \frac{1}{2} \text{diag}[\min(o_{v11}(t)), \min(o_{v22}(t))] \quad (31)$$

のように設定した。また、環境からの外乱の影響を考慮して、制御入力 F_{act} に正規乱数 (平均 0[N], 標準偏差 1[N]) を加えた。

Fig. 6 にマニピュレータの手先軌道の変化を示す。図中の試行回数は betterment process を適用した回数で、betterment process 1 回につき NNPC, NNVC を 100 回学習している。目標軌道 $X_d(t)$ は、境界条件

$$X_d(0) = (x_d(0), y_d(0))^T = (-0.3, 0.2)^T [\text{m}] \quad (32)$$

$$X_d(t_f) = (x_d(t_f), y_d(t_f))^T = (0.5, -0.2)^T [\text{m}] \quad (33)$$

$$\dot{X}_d(0) = (0, 0)^T [\text{m/sec}], \quad \dot{X}_d(t_f) = (0, 0)^T [\text{m/sec}] \quad (34)$$

$$\ddot{X}_d(0) = (0, 0)^T [\text{m/sec}^2], \quad \ddot{X}_d(t_f) = (0, 0)^T [\text{m/sec}^2] \quad (35)$$

に対して 5 次多項式¹⁾ で与えた。ただし、動作終了時刻 t_f は 1 sec とし、サンプリングタイム Δt は 0.001 sec とした ($N_f=1000$)。Fig. 6 から数回の試行回数でマニピュレータの手先は目標軌道に一致していることがわかる。

学習によるインピーダンスパラメータ $M_d^{-1}K_d$, $M_d^{-1}B_d$ の変化を Table 1 に示す。表中の $E[M_d^{-1}K_d]$, $E[M_d^{-1}B_d]$ は、それぞれインピーダンスパラメータの時間平均

$$E[M_d^{-1}K_d] = \frac{1}{1000} \sum_{t=0}^{1000} M_d^{-1}K_d(t) \quad (36)$$

$$E[M_d^{-1}B_d] = \frac{1}{1000} \sum_{t=0}^{1000} M_d^{-1}B_d(t) \quad (37)$$

を表わす。重み係数の初期値を用いて計算した学習前のインピーダンスパラメータに比べて、10 試行の学習後のインピーダンスパラメータは対角要素が増加し、非対角要素が減少している。

4.2 拘束運動

つぎに学習後の NNFC を用いて、Fig. 7 に示すように Fig. 6 の x 軸上に障害物を設定して拘束運動の学習を行った。対象物の特性は $M_e = \text{diag}[0, 0]$ [kg], $B_e = \text{diag}[0, 10]$ [N·sec/m], $K_e = \text{diag}[0, 1.0 \times 10^5]$ [N/m] ((2) 式) とし、簡単のため x 軸方向の摩擦は無視した。

用いた NNFC は、入力ユニット 6, 中間ユニット 10, 出力ユニット 4 の 3 層構造で、学習率は、 $\eta_f=6.43 \times$

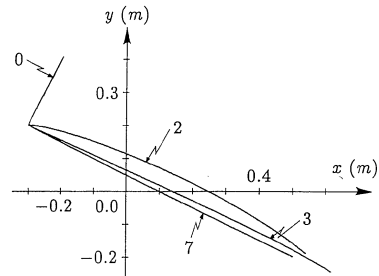


Fig. 6 End-point trajectories of the manipulator during learning of free movement

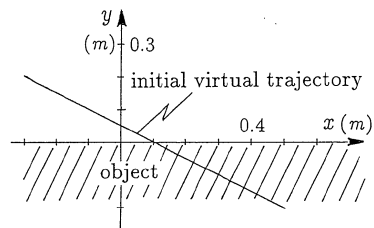


Fig. 7 Task space for contact movement

Table 1 Impedance parameters before and after learning for free movement

	$E[M_d^{-1}K_d]$	$E[M_d^{-1}B_d]$
before learning	$\begin{bmatrix} 1.58855 & 1.14292 \\ 1.00605 & 1.08828 \end{bmatrix}$	$\begin{bmatrix} 1.24700 & 1.62821 \\ 1.40561 & 1.25643 \end{bmatrix}$
after learning (10 trials)	$\begin{bmatrix} 495.19279 & 0.44871 \\ 0.44652 & 475.05515 \end{bmatrix}$	$\begin{bmatrix} 497.59935 & 0.41674 \\ 0.56942 & 494.30498 \end{bmatrix}$

10^{-11} ((23)式), $\eta_d = 6.0 \times 10^{-8}$ ((28)式)とした。誤差修正ゲイン Γ_f , $\Phi_f \in R^{2 \times 2}$ ((26)式)は

$$\Gamma_f = \frac{1}{2} \text{diag}[\min(o_{f11}(t)), \min(o_{f22}(t))] \quad (38)$$

$$\Phi_f = \frac{1}{1000} \Gamma_f \quad (39)$$

で、ほかのパラメータは自由運動のシミュレーションと同様である。

Fig. 8 に窓関数の長さ N を $N=100(0.1 \text{ sec})$ としたときの手先力の変化を示す。(24)式の窓関数 $h(i)$ には、Hanning 窓¹⁰⁾を用いた。目標手先力 F_d は $F_d = (0, -100.0)^T$ [N] で、接触直後(0.01 sec)は不連続性を避けるため、5次多項式を用いて設定した。学習前は、マニピュレータが目標軌道を追従しようとする、および、自由運動で学習した大きなインピーダンスパラメータを有しているため、大きな反力 F_{int} が発生している。しかし、数十回の繰返し学習で手先力がほぼ目標値と一致している。

Fig. 9 は窓関数の長さ N を $N=0, 100, 200$ としたときのそれぞれ 30 回繰返し学習後の仮想軌道 X_d である。いずれの場合も手先力 F_{int} は目標値 F_d にほぼ一致するが、学習後の X_d が大きく異なっている。 $N=0$ では、手先が対象物に接触する際に発生する大きな衝撃力を吸収するため、仮想軌道が急激に変化している。これに対して、 N を大きくすると、接触前から仮想軌道が変化し、自由運動から拘束運動への移行が滑らかに行われている。

Table 2 に拘束運動中のインピーダンスパラメータを示す。表中の $E[M_d]$, $E[B_d]$, $E[K_d]$ は、NNFC, NNTC の出力から、

$$E[M_d] = \frac{1}{N_f - N_c} \sum_{t=N_c}^{N_f} [M_d^{-1}(t)]^{-1} \quad (40)$$

$$E[B_d] = \frac{1}{N_f - N_c} \sum_{t=N_c}^{N_f} [M_d^{-1}(t)]^{-1} [M_d^{-1} B_d(t)] \quad (41)$$

$$E[K_d] = \frac{1}{N_f - N_c} \sum_{t=N_c}^{N_f} [M_d^{-1}(t)]^{-1} [M_d^{-1} K_d(t)] \quad (42)$$

により計算した。ただし、 N_c は接触時刻を表す。学習前の粘性行列、剛性行列 $E[B_d]$, $E[K_d]$ は、対角要素がともに大きな値となっているが、30 試行の学習後のインピーダンスパラメータは慣性、粘性、剛性とも接触面の法線方向 (y 軸方向) には小さく、接触面の接線方向 (x 軸方向) には大きな値となっている。学習により、イン

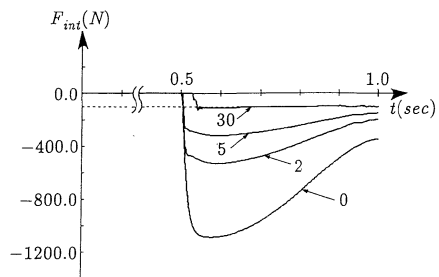


Fig. 8 End-point force patterns of the manipulator during learning of a constrained movement ($N=100$)

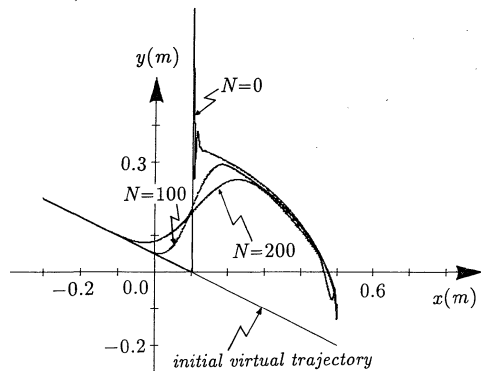


Fig. 9 Learning results of virtual trajectories for contact movements ($N=0, 100, 200$)

ピーダンスパラメータがうまく調節されていることがわかる。

5. まとめ

本稿では、ニューラルネットを用いてマニピュレータの手先インピーダンスを学習的に調節する方法を提案した。この方法は、マニピュレータを繰り返し動作させながら、追値制御誤差、力制御誤差を減少させるようにニューラルネットを学習させるもので、慣性、粘性、剛性のすべてのインピーダンスパラメータを調節することができる。また、誤差関数に窓関数を導入することにより、接触に備えてあらかじめ仮想軌道を変化させることができ、自由運動から拘束運動への移行を滑らかに行う

Table 2 Impedance parameters before and after learning for free contact movement ($N=100$)

	$E[M_d]$	$E[B_d]$	$E[K_d]$
before learning	$\begin{bmatrix} 3.64677 & -1.81733 \\ -4.53264 & 4.88745 \end{bmatrix}$	$\begin{bmatrix} 1816.51098 & -899.88635 \\ -2256.31941 & 2422.31156 \end{bmatrix}$	$\begin{bmatrix} 1802.87264 & -858.30643 \\ -2239.63566 & 2310.56670 \end{bmatrix}$
after learning (30 trials)	$\begin{bmatrix} 2.40574 & -0.10942 \\ -0.30697 & 0.27765 \end{bmatrix}$	$\begin{bmatrix} 1198.88872 & -53.43823 \\ -152.88977 & 137.68601 \end{bmatrix}$	$\begin{bmatrix} 1190.01721 & -50.52533 \\ -151.66848 & 131.14677 \end{bmatrix}$

ことが可能である。

本手法は、学習アルゴリズムとして誤差逆伝搬法を採用しているので、必ずしも慣性、粘性、剛性および仮想軌道を全て学習的に獲得する必要はない。たとえば、作業空間の剛性を直接、設定したい場合には、剛性をあらかじめ望ましい値に固定し、剛性以外のインピーダンスパラメータをニューラルネットで学習させればよい。このとき、本手法は剛性を設定値に固定したという拘束条件のもとで、制御誤差を最小(極小)にするという意味で最適なインピーダンスパラメータを求めることができる。もちろん、設計者は、任意のインピーダンスパラメータを指定することが可能である。

今後は、マニピュレータの非線形補償が完全に行えない場合やマニピュレータの手先に働く非線形な摩擦力などを考慮して、betterment process を広義ニュートン法などに拡張するとともに、学習によって実現される手先インピーダンスや仮想軌道の力学的な解釈、学習速度の改善、ニューラルネットの一般化能力などについて考察する必要がある。なお、本研究の一部は(財)日産科学振興財団の研究助成によるもので、ここに改めて謝意を表します。

参 考 文 献

- 1) 吉川恒夫：ロボット制御基礎論，コロナ社（1988）
- 2) 羅，伊藤：コンプライアントマニピュレーションのためのロボットの制御設計，計測自動制御学会論文集，26-4，427/434（1990）
- 3) 松尾，岩城：組立作業におけるコンプライアンス設定に関する検討，日本機械学会ロボティクス・メカトロニクス講演会講演論文集，910-31，（Vol. A），51/54（1991）
- 4) H. Gomi and M. Kawato: Neural-Network Control for a Closed-Loop System Using Feedback-Error-

Learning, ATR Technical Report, TR-A-0114, 1/35 (1991)

- 5) H. Asada: Teaching and Learning of Compliance Using Neural Nets: Representation and Generation of Nonlinear Compliance, Proc. of IEEE Int. Conf. on Robotics and Automation, 1237/1244 (1990)
- 6) M. Cohen and T. Flash: Learning Impedance Parameters for Robot Control Using an Associative Search Network, IEEE Trans., RA-7-3, 382/390, June (1991)
- 7) 川村，宮崎，有本：動的システムの学習的制御法(Betterment Process)の提案，計測自動制御学会論文集，22-1，56/62（1986）
- 8) 川人，宮本，磯部，鈴木：作業座標における手の繰り返し学習制御—広義ニュートン法のアルゴリズム—，信学技報，MBE-85-91，83/92（1985）
- 9) 前田，狩野：インピーダンス制御されたマニピュレータの繰り返し学習制御，第14回 Dynamical System Theory シンポジウム，261/266（1991）
- 10) 伊藤正美監修：生体信号処理の基礎，オーム社（1985）

[著 者 紹 介]

辻 敏 夫（正会員）

（Vol. 28, No. 11 参照）

西 田 正 孝（正会員）



1992 年，広島大学大学院工学研究科博士課程前期情報工学専攻修了。同年富士ゼロックス(株)入社，在学中，ニューラルネットによるマニピュレータ制御に関する研究に従事。

伊 藤 宏 司（正会員）

（Vol. 28, No. 11 参照）