

# ニューラルネットによるロボットインピーダンスの オンライン学習

辻 敏夫\* 赤松 寛優\* 原田 研介\* 金子 真\*

## On-line Learning of Robot Arm Impedance Using Neural Networks

Toshio Tsuji\*, Hiromasa Akamatsu\*, Kensuke Harada\* and Makoto Kaneko\*

Impedance control is one of the most effective control methods for a manipulator in contact with its environments. The characteristics of force and motion control, however, are determined by the impedance parameters of the end-effector of the manipulator which must be designed according to the given task. In this paper, we propose a method to regulate impedance parameters of the manipulator's end-effector while identifying the characteristics of the environments using neural networks through on-line learning. Four kinds of neural networks are used: three for the position, velocity and force control of the end-effector, and one for the identification of environments. First, the neural networks for the position and velocity control are trained during free movements. Then, the neural networks for the force control and the identification of environments are trained during contact movements. Computer simulations show that the method can regulate stiffness, viscosity and inertia parameters of the end-effector and identify the unknown property of the environments through on-line learning.

**Key Words:** Impedance Control, Robot Manipulator, Neural Network, On-line Learning

### 1. ま え が き

環境に接触して作業を行う場合、環境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、環境に対する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作業に有効である[1]。しかしながらこのためには、環境と与えられた作業内容、またそれらの時間変化を考慮して適切なインピーダンスパラメータや目標軌道を設定する必要があり、その設定法を一般的に導出することは容易ではない。従来からこの問題に対する研究が活発に行われており、インピーダンスパラメータ自体を学習的に調節する方法や手先の目標軌道を望ましい方向に修正する方法などが検討されている。

例えば Luo and Ito [2] は、自由運動時における手先の追値制御特性を設計した上で拘束運動時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モデルマッチング法を提案している。また、山本ら [3] は、与えられた作業に対してまず目標のインピーダンスを設定し、環境との接触位置と接触力に基づいてタスクインピーダンスと目標軌道の両方を修正することで目標接触力を実現するというタスクインピーダンス制御を提案している。しかしながらいずれの方法においても、環境

が非線形特性を含んでいる場合や未知の場合には、適切なインピーダンスパラメータや目標軌道の調節は難しい。

この問題に対して本研究では、ニューラルネット(以下 NN と略記)を用いて学習的に目標インピーダンスを調節することを考える。これまでも、NN をマニピュレータの力制御やハイブリッド制御[4]~[11]に適用した研究は数多く行われている。また NN を利用したインピーダンス制御に関しても多くの研究報告がなされているが、それらの多くはマニピュレータや環境のモデルに含まれる不確かさや外乱に対して学習的に望ましいインピーダンスを実現しようとするもので、目標インピーダンス自体はあらかじめ与えられると仮定している[12]~[16]。これに対して目標インピーダンスそのものを作業や環境に応じて学習的に獲得しようとする研究もいくつか報告されている。

例えば, Asada [17] は、カフィードバック制御器に NN を用いることで、非線形な粘性コンプライアンスを学習的に求める方法を提案した。Cohen and Flash [18] は、手先の速度と力の評価に基づいて NN による学習的な粘弾性パラメータ調節法を検討している。この方法では、目標速度の修正を行うことで学習効率の向上を図っているが、慣性インピーダンスの調節は考慮しておらず、また対象とする作業は拘束運動に限定している。また Yang and Asada [19] は作業速度を段階的に変化させながら目標インピーダンスを学習するという Progressive learning 法を提案しているが、作業速度の変化のさせ方が明確でなく、また学習終了までに非常に多くの繰り返し試行(100~300回)

原稿受付 1998年2月4日

\*広島大学工学部

\*Faculty of Engineering, Hiroshima University

が必要である．これに対して，辻ら [20] [21] は慣性，粘性，剛性のすべてのインピーダンスパラメータの調節を NN を利用した繰り返し学習で行っており，同時に目標軌道を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が可能であることを示した．しかしながらこの方法ではオフライン学習を用いているため，周期的な作業にしか用いることができない．

そこで本論文では文献 [20] [21] の方法を拡張し，NN を用いて学習的に慣性，粘性，剛性のインピーダンスパラメータを調節すると同時に目標軌道も積極的に修正可能なオンライン学習法を開発することを目的とする．しかしながら，オンライン学習を実現するためには，適切な学習則の導出が必要であり，そのためには環境の特性を同定する必要がある．本論文で提案する方法は，環境のモデル化誤差を同定する NN を用いることにより，環境の同定とインピーダンスパラメータの学習を同時に実現することができる．以下，2. でインピーダンス制御の定式化を行い，3 章で手先インピーダンスを学習する NN と環境のモデルを同定する NN の学習則を導出する．4. では接触作業の一例として，モデル化誤差を含む環境に対する自由運動から拘束運動への移行作業を取り上げ，シミュレーション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

一般に環境に接触しているマニピュレータの運動方程式は，作業空間の自由度を  $l$ ，関節自由度を  $m$  とすると，

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

となる．ここで， $\theta \in \mathbb{R}^m$  は関節角度ベクトル， $M(\theta) \in \mathbb{R}^{m \times m}$  は慣性行列， $h(\theta, \dot{\theta}) \in \mathbb{R}^m$  はコリオリ力，遠心力等を表す項， $g(\theta) \in \mathbb{R}^m$  は重力トルク， $\tau \in \mathbb{R}^m$  は関節駆動トルク， $J \in \mathbb{R}^{l \times m}$  はヤコビ行列を表す．また， $F_c \in \mathbb{R}^l$  は環境から手先に作用する接触力であり，環境を時変で非線形な要素を含むモデルを用いて表現すると，

$$F_c = g(dX_o, d\dot{X}_o, d\ddot{X}_o, t) \quad (2)$$

となる．ただし， $dX_o = X_o^* - X$  は手先の現在位置  $X$  と環境の平衡位置  $X_o^*$  との偏差を表す． $g(\cdot)$  は非線形関数である．

いま，手先の目標インピーダンスを作業空間で

$$M_e d\ddot{X} + B_e d\dot{X} + K_e dX = F_d - F_c \quad (3)$$

により記述する．ただし， $M_e, B_e, K_e \in \mathbb{R}^{l \times l}$  はそれぞれ，手先の目標慣性行列，目標粘性行列，目標剛性行列で， $dX = X_d - X$  は目標手先位置  $X_d$  と現在位置  $X$  との偏差を， $F_d \in \mathbb{R}^l$  は目標接触力を表す．

式 (1) に対して，

$$\tau = \{\hat{M}^{-1}(\theta)J^T M_x(\theta)J\}^T \hat{h}(\theta, \dot{\theta}) + \hat{g}(\theta) - J^T(\theta)F_c + J^T M_x(\theta) \{F_{act} - \hat{J}\dot{\theta}\} \quad (4)$$

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次のように線形化される．

$$\ddot{X} = F_{act} \quad (5)$$

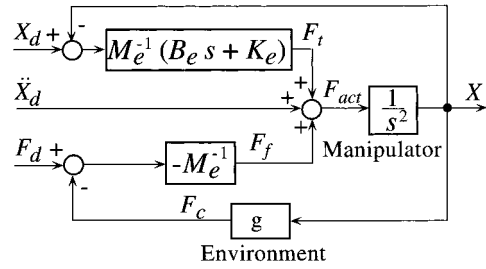


Fig. 1 Impedance control system represented in the task space

ただし， $\hat{h}(\theta, \dot{\theta})$ ， $\hat{g}(\theta)$  はそれぞれ  $h(\theta, \dot{\theta})$ ， $g(\theta)$  の推定値， $M_x(\theta) = (J\hat{M}^{-1}(\theta)J^T)^{-1} \in \mathbb{R}^{l \times l}$  はアームが特異姿勢にない限り正則で， $\hat{M}(\theta)$  は慣性行列の推定値を表す．また， $F_{act} \in \mathbb{R}^l$  は作業空間で表現した制御入力である．

このとき，式 (3) の目標インピーダンスを実現する制御力は，

$$F_{act} = F_t + F_f + \ddot{X}_d \quad (6)$$

$$F_t = M_e^{-1}(B_e d\dot{X} + K_e dX) \quad (7)$$

$$F_f = -M_e^{-1}(F_d - F_c) \quad (8)$$

となる [2] [20] [21] ．

Fig. 1 にインピーダンス制御系のブロック線図を示す．図から明らかなように，自由運動時には  $F_d = F_c = 0$  となり，位置，速度制御のみが機能する．一方，環境と接触している拘束運動時には同時に力制御も行われる．しかしながら，この手先の追従特性はインピーダンスパラメータに大きく依存する．そこで次章では，これらのインピーダンスパラメータを NN により学習的に調節する方法を提案する．

## 3. 手先インピーダンスのオンライン学習

### 3.1 制御系の構造

本手法では，インピーダンスパラメータを調節するための NN と環境を同定するための NN を制御系に導入する．Fig. 2 に本論文で提案する制御系の構成を示す．この制御系には位置制御を行う PCN ( Position Control Network ) と速度制御を行う VCN ( Velocity Control Network )，さらに力制御を行う FCN ( Force Control Network ) の三つの NN が組み込まれている．PCN および VCN は手先の位置，速度，目標軌道との偏差を，FCN はさらに接触力を入力とし，手先インピーダンスパラメータ  $M_e^{-1}B_e$ ， $M_e^{-1}K_e$ ， $M_e^{-1}$  をそれぞれ出力する多層型 NN である．ただし，入力ユニットには線形の出力関数を，中間層，出力層にはシグモイド関数  $\sigma_i(x)$

$$\sigma_i(x) = a_i \tanh(x) \quad (9)$$

を用いる． $a_i$  はユニットからの出力の最大値を調節するための正の定数である．また，各 NN の出力は

$$O_p = (o_{p1}^T, o_{p2}^T, \dots, o_{pl}^T)^T \quad (10)$$

$$O_v = (o_{v1}^T, o_{v2}^T, \dots, o_{vl}^T)^T \quad (11)$$

$$O_f = (o_{f1}^T, o_{f2}^T, \dots, o_{fl}^T)^T \quad (12)$$

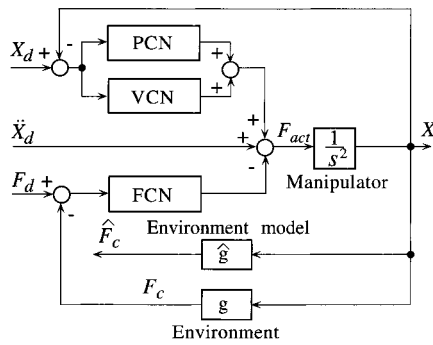


Fig. 2 Impedance control system including three neural networks

のように表現する．ただし， $o_{pi}, o_{vi}, o_{fi} \in \mathbb{R}^l$  はそれぞれ  $M_e^{-1}B_e, M_e^{-1}K_e, M_e^{-1}$  の第  $i$  行に対応したベクトルで， $O_p, O_v, O_f \in \mathbb{R}^{l^2}$  はそれぞれ PCN, VCN, FCN の出力を縦に並べたベクトルである [20]．

3 種類の NN の学習は次のような手順で行う．まず，マニピュレータを自由空間で運動させることにより，目標軌道に対する手先の追従誤差（位置誤差  $dX$  と速度誤差  $d\dot{X}$ ）を減少させるように PCN と VCN を学習する．そして，自由運動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，PCN と VCN は固定し，手先の力制御誤差を減少させるように FCN および目標軌道  $X_d$  を修正する．FCN だけでなく目標軌道の修正を行うことにより，より効果的な学習を目指している．

### 3.2 自由運動の学習

Fig. 3 に自由運動における PCN, VCN の構成を示す．このとき，マニピュレータの制御入力  $F_{act}$  は，

$$F_{act} = F_t + \ddot{X}_d = F_p + F_v + \ddot{X}_d$$

$$= \begin{bmatrix} o_{p1}^T \\ o_{p2}^T \\ \dots \\ o_{pl}^T \end{bmatrix} dX + \begin{bmatrix} o_{v1}^T \\ o_{v2}^T \\ \dots \\ o_{vl}^T \end{bmatrix} d\dot{X} + \ddot{X}_d \quad (13)$$

となる．ここで， $F_p, F_v \in \mathbb{R}^l$  はそれぞれ，PCN, VCN を用いて計算された制御入力である．

学習の評価関数には

$$E_t(t) = E_p(t) + E_v(t) \quad (14)$$

$$E_p(t) = \frac{1}{2} dX(t)^T dX(t) \quad (15)$$

$$E_v(t) = \frac{1}{2} d\dot{X}(t)^T d\dot{X}(t) \quad (16)$$

を用い，PCN および VCN の重み係数  $w_{ij}^{(p)}, w_{ij}^{(v)}$  を誤差の最急降下方向へ修正する．

$$\Delta w_{ij}^{(p)}(t) = -\eta_p \frac{\partial E_t(t)}{\partial w_{ij}^{(p)}(t)} \quad (17)$$

$$\Delta w_{ij}^{(v)}(t) = -\eta_v \frac{\partial E_t(t)}{\partial w_{ij}^{(v)}(t)} \quad (18)$$

$$\frac{\partial E_t(t)}{\partial w_{ij}^{(p)}(t)} = \frac{\partial E_t(t)}{\partial X(t)} \frac{\partial X(t)}{\partial F_p(t)} \frac{\partial F_p(t)}{\partial O_p(t)} \frac{\partial O_p(t)}{\partial w_{ij}^{(p)}(t)} \quad (19)$$

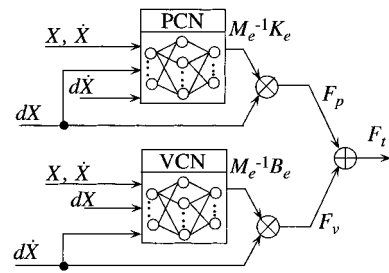


Fig. 3 Neural networks for trajectory control

$$\frac{\partial E_t(t)}{\partial w_{ij}^{(v)}(t)} = \frac{\partial E_t(t)}{\partial \dot{X}(t)} \frac{\partial \dot{X}(t)}{\partial F_v(t)} \frac{\partial F_v(t)}{\partial O_v(t)} \frac{\partial O_v(t)}{\partial w_{ij}^{(v)}(t)} \quad (20)$$

$\eta_p, \eta_v$  は各 NN の学習率を表す．ここで， $\frac{\partial E_t(t)}{\partial X(t)}, \frac{\partial E_t(t)}{\partial \dot{X}(t)}, \frac{\partial F_p(t)}{\partial O_p(t)}, \frac{\partial F_v(t)}{\partial O_v(t)}$  は式 (13)~(16) より計算可能であり， $\frac{\partial O_p(t)}{\partial w_{ij}^{(p)}(t)}$  と  $\frac{\partial O_v(t)}{\partial w_{ij}^{(v)}(t)}$  は誤差逆伝播法 [22] により計算できる．しかしながら  $\frac{\partial X(t)}{\partial F_p(t)}, \frac{\partial \dot{X}(t)}{\partial F_v(t)}$  はマニピュレータのダイナミクスを考慮して計算せねばならない．

オンライン学習を行うためには式 (19)(20) に含まれる偏微分計算を毎サンプリングごとに行うことになる．そこで，微小な変分を用いて  $\frac{\partial X(t)}{\partial F_p(t)} \approx \frac{\Delta X(t)}{\Delta F_p(t)}, \frac{\partial \dot{X}(t)}{\partial F_v(t)} \approx \frac{\Delta \dot{X}(t)}{\Delta F_v(t)}$  と近似する． $F_p, F_v$  を微量  $\Delta F_p, \Delta F_v$  だけ変化させたとき，

$$\Delta X(t) \approx \Delta F_p(t) \Delta t_s^2 \quad (21)$$

$$\Delta \dot{X}(t) \approx \Delta F_v(t) \Delta t_s \quad (22)$$

と近似することができるので [23]， $\frac{\partial X(t)}{\partial F_p(t)} \approx \Delta t_s^2 I, \frac{\partial \dot{X}(t)}{\partial F_v(t)} \approx \Delta t_s I$  を得る．ただし， $\Delta t_s$  は十分に小さなサンプリングタイムで， $I$  は  $l$  次の単位行列である．

以上より，この学習則を用いて追従誤差が最小となるまで学習を行うことができれば，PCN, VCN はそれぞれ最適なインピーダンスパラメータ  $M_e^{-1}K_e, M_e^{-1}B_e$  をネットワークの出力  $O_p(t), O_v(t)$  として表現することになる．

### 3.3 NN による環境の同定

拘束運動における手先インピーダンスの学習をオンラインで行うためには，力制御誤差を手先の運動誤差に変換して学習を行うことになるため，環境モデル

$$\hat{F}_c = \hat{g}(dX_o, d\dot{X}_o, d\ddot{X}_o, t) \quad (23)$$

の同定が必要となる．

ここで問題によっては環境モデルがあらかじめ与えられる場合も考えられる．本論文では線形で時不変な環境モデル

$$F_{cm} = g_m(dX_o, d\dot{X}_o, d\ddot{X}_o) = K_c dX_o + B_c d\dot{X}_o + M_c d\ddot{X}_o \quad (24)$$

が与えられるものとする．ただし， $K_c, B_c, M_c \in \mathbb{R}^{l \times l}$  は環境モデルの剛性，粘性，慣性である．しかしながら，一般には実際の環境式 (2) と与えられた環境モデル  $g_m$  の間には誤差が生じる可能性があるため，ここではニューラルネット EIN

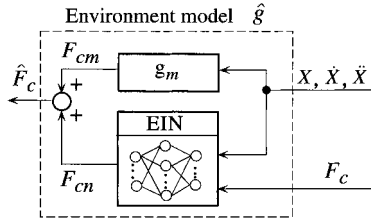


Fig. 4 Identification of the environment using EIN

(Environment Identification Network) を用いて環境モデルの同定を行う。EIN を導入した環境モデルの構成を Fig. 4 に示す。EIN はあらかじめ与えられた環境モデル  $g_m$  と並列に配置し、接触力の推定を行う。EIN の入力は測定した接触力および、手先位置、速度、加速度で、出力は接触力の修正量  $F_{cn}$  である。このとき接触力の推定値は

$$\hat{F}_c = F_{cm} + F_{cn} \quad (25)$$

となる。

EIN の学習は評価関数

$$E_e(t) = \frac{1}{2} \{ \hat{F}_c(t) - F_c(t) \}^T \{ \hat{F}_c(t) - F_c(t) \} \quad (26)$$

を減少させるように行う。EIN の重み係数  $w_{ij}^{(e)}$  の修正量は、

$$\Delta w_{ij}^{(e)}(t) = -\eta_e \frac{\partial E_e(t)}{\partial w_{ij}^{(e)}(t)} \quad (27)$$

$$\frac{\partial E_e(t)}{\partial w_{ij}^{(e)}(t)} = \frac{\partial E_e(t)}{\partial \hat{F}_c(t)} \frac{\partial \hat{F}_c(t)}{\partial F_{cn}(t)} \frac{\partial F_{cn}(t)}{\partial w_{ij}^{(e)}(t)} \quad (28)$$

となる。 $\eta_e$  は NN の学習率を表す。ここで  $\frac{\partial E_e(t)}{\partial \hat{F}_c(t)}$ 、 $\frac{\partial \hat{F}_c(t)}{\partial F_{cn}(t)}$  は式 (25)(26) より計算可能で、 $\frac{\partial F_{cn}(t)}{\partial w_{ij}^{(e)}}$  は誤差逆伝播法を用いて計算できる。

この学習則を用いて評価関数  $E_e(t)$  が最小となるまで学習を行ったとき、 $\hat{F}_c = F_c$  の関係が成り立ち、 $\hat{F}_c$  を拘束運動の学習則に利用することが可能となる。

### 3.4 拘束運動の学習

次に Fig. 5 に拘束運動における FCN の構成を示す。拘束運動時は目標手先力を実現する方向に FCN の学習を行う。ただし、PCN および VCN の重み係数は固定しておく。このとき、マニピュレータへの制御入力  $F_{act}$  は式 (13) を変更して、

$$F_{act} = F_t + F_f + \ddot{X}_d \\ = F_t - \begin{bmatrix} o_{f1}^T \\ o_{f2}^T \\ \dots \\ o_{fl}^T \end{bmatrix} (F_d - F_c) + \ddot{X}_d \quad (29)$$

となる。

FCN の学習は評価関数

$$E_f(t) = \frac{1}{2} \{ F_d(t) - F_c(t) \}^T \{ F_d(t) - F_c(t) \} \quad (30)$$

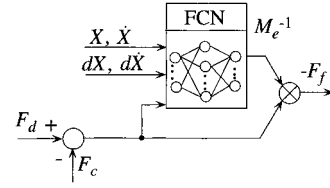


Fig. 5 Neural network for force control

を減少させるように行う。FCN の重み係数  $w_{ij}^{(f)}(t)$  は誤差の最急降下方向へ修正する。

$$\Delta w_{ij}^{(f)}(t) = -\eta_f \frac{\partial E_f(t)}{\partial w_{ij}^{(f)}(t)} \quad (31)$$

$$\frac{\partial E_f(t)}{\partial w_{ij}^{(f)}(t)} = \frac{\partial E_f(t)}{\partial F_c(t)} \left\{ \frac{\partial F_c(t)}{\partial X(t)} \frac{\partial X(t)}{\partial F_f(t)} + \frac{\partial F_c(t)}{\partial \dot{X}(t)} \frac{\partial \dot{X}(t)}{\partial F_f(t)} \right. \\ \left. + \frac{\partial F_c(t)}{\partial \ddot{X}(t)} \frac{\partial \ddot{X}(t)}{\partial F_f(t)} \right\} \frac{\partial F_f(t)}{\partial O_f(t)} \frac{\partial O_f(t)}{\partial w_{ij}^{(f)}(t)} \quad (32)$$

$\eta_f$  は NN の学習率を表す。 $\frac{\partial E_f(t)}{\partial F_c(t)}$ 、 $\frac{\partial F_f(t)}{\partial O_f(t)}$  は式 (29)(30) より計算可能であり、 $\frac{\partial O_f(t)}{\partial w_{ij}^{(f)}(t)}$  は誤差逆伝播法により計算できる。

$\frac{\partial X(t)}{\partial F_f(t)}$ 、 $\frac{\partial \dot{X}(t)}{\partial F_f(t)}$ 、 $\frac{\partial \ddot{X}(t)}{\partial F_f(t)}$  は自由運動時と同様に  $\frac{\partial X(t)}{\partial F_f(t)} \approx \Delta t_s^2 I$ 、 $\frac{\partial \dot{X}(t)}{\partial F_f(t)} \approx \Delta t_s I$ 、 $\frac{\partial \ddot{X}(t)}{\partial F_f(t)} = I$  と近似することができる。一方、 $\frac{\partial F_c(t)}{\partial X(t)}$ 、 $\frac{\partial F_c(t)}{\partial \dot{X}(t)}$ 、 $\frac{\partial F_c(t)}{\partial \ddot{X}(t)}$  の計算は環境のダイナミクスを考慮しなければならないため、環境モデルにより推定した  $\hat{F}_c$  を用いる。 $\hat{F}_c = F_c$  が成り立つまで EIN の学習が進んでいれば、式 (24)(25) を用いることでこれらの偏微分計算が可能となる。

さらに拘束運動では、FCN の学習の負担を軽減するために、インピーダンスパラメータを調節するだけでなく、同時に目標軌道の修正も行う。目標軌道の修正量  $\Delta X_d(t)$  は、

$$\Delta X_d(t) = -\eta_d \frac{\partial E_f(t)}{\partial X_d(t)} \quad (33)$$

$$\frac{\partial E_f(t)}{\partial X_d(t)} = \frac{\partial E_f(t)}{\partial \hat{F}_c(t)} \frac{\partial \hat{F}_c(t)}{\partial X(t)} \frac{\partial X(t)}{\partial F_f(t)} \frac{\partial F_f(t)}{\partial X_d(t)} \quad (34)$$

となる。 $\eta_d$  は目標軌道の修正率を表す。目標速度の修正も同様に行うことができる。

以上より、この学習則を用いて手先の力誤差が最小となるまで学習を行ったとき、FCN は最適なインピーダンスパラメータ  $M_c^{-1}$  をネットワークの出力  $O_f(t)$  として表現することになる。

本節で導出した学習則は環境モデルの同定が十分進んでいる場合 ( $\hat{F}_c \approx F_c$ ) のみ有効である。しかしながら、予期せぬ環境の特性の変化や EIN の学習誤差などにより、 $\hat{F}_c$  の推定誤差が増大する恐れがある。そこで FCN の学習率  $\eta_f$  および目標軌道の修正率  $\eta_d$  を同定誤差の関数とする。

$$\eta_f(t) = \frac{\eta_f^{\text{MAX}}}{1 + p E_e(t)} \quad (35)$$

$$\eta_d(t) = \frac{\eta_d^{\text{MAX}}}{1 + p E_e(t)} \quad (36)$$

ここで、 $\eta_f^{MAX}$ 、 $\eta_d^{MAX}$  はそれぞれ  $\eta_f(t)$ 、 $\eta_d(t)$  の最大値で、 $p$  は正の定数である。この関数を用いることで EIN の学習が十分進んでおらず学習誤差  $E_e(t)$  が大きい場合には、自動的に学習率、修正率が小さくなり、誤った学習が行われるのを回避することができる。

#### 4. 接触作業への応用

本手法の有効性を検討するため、4 関節マニピュレータによる平面運動のコンピュータシミュレーションを行った。一例として、Fig. 6 に示す自由運動と拘束運動の両方を含む作業を取り上げる。なお、インピーダンス制御則には冗長マニピュレータを対象とした階層型インピーダンス制御則 [24] [25] を用いた。使用したマニピュレータのリンクパラメータは Table 1 に示す。

##### 4.1 自由運動の学習

まず自由運動における PCN, VCN の学習を行う。用いた NN は、PCN, VCN とともに 4 層構造で、ユニット数は入力層 8、中間層  $20 \times 2$  層、出力層 4 である。重み行列  $w_{ij}^{(p)}$ 、 $w_{ij}^{(v)}$  の初期値はともに  $|w_{ij}| < 0.01$  の一様乱数で、学習率は試行錯誤的に決定し、 $\eta_p = 13,000$ 、 $\eta_v = 15$  とした。学習率がかなり大きな値になっているのは、学習則に式 (21) (22) の  $\Delta t_s^2$ 、 $\Delta t_s$  が含まれるためである。動力学計算のサンプリングタイムを  $\Delta t_s = 0.001$  [s] としたので、これを補うために学習率をかなり大きく設定する必要があった。また、NN の出力値は最大値を 100、最小値を  $-100$  とした。

Fig. 7 に学習に伴うマニピュレータの手先軌道の変化を示す。手先の目標軌道は 8 [s] で反時計回りに一回転する円軌道で、時間軌道は滑らかな 5 次多項式 [26] で与えた。Fig. 7 は学習を行わずにこの目標軌道を一回転 (8 秒間) マニピュレータに与えた場合の結果と、NN の学習を行いながらこの目標軌道連続して 2 回転 (16 秒間) 与えた場合の結果を重ね書き

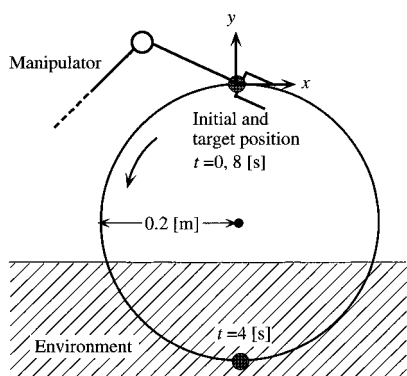


Fig. 6 An example of a contact task

Table 1 Link parameters of the four-link manipulator

	link 1	link 2	link 3	link 4
length [m]	0.2	0.2	0.2	0.2
mass [kg]	1.57	1.57	1.57	1.57
moment of inertia [kgm <sup>2</sup> ]	0.8	0.8	0.8	0.8

ている。図中の数字はこの回転の回数を表す。

図より NN の学習を行わない場合は、目標軌道に対する追従能力は非常に低いが、オンラインで学習を行うことにより、1 回目からほぼ目標軌道を追従しており、2 回目以降ではほとんど一致していることが分かる。

Fig. 8 に学習前と学習後の PCN と VCN の出力であるインピーダンスパラメータを示す。(a) は自由運動を行う前の値で、(b) は学習終了時の行列の各要素をグラフ表示したものである。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、非対角要素は非常に小さい値に収束していることが分かる。また、各出力値は時間に依らずほぼ一定の値を示した。

##### 4.2 拘束運動の学習

次に学習後の PCN, VCN を用いて、拘束運動の学習を行う。マニピュレータにとって環境の位置は未知で、モデル化誤差を含んでいる。そこで環境に接触した時点からモデル化誤差を EIN で同定しながら、接触力の推定値  $\hat{F}_c$  を用いて FCN の

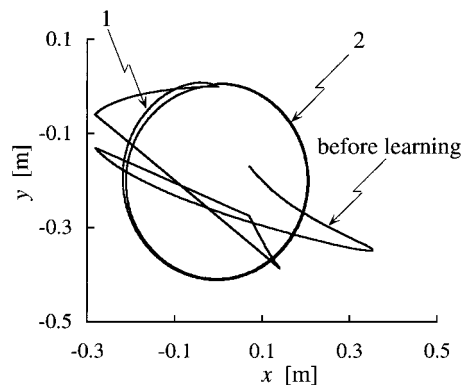


Fig. 7 End-point trajectories of the manipulator during learning of a free movement

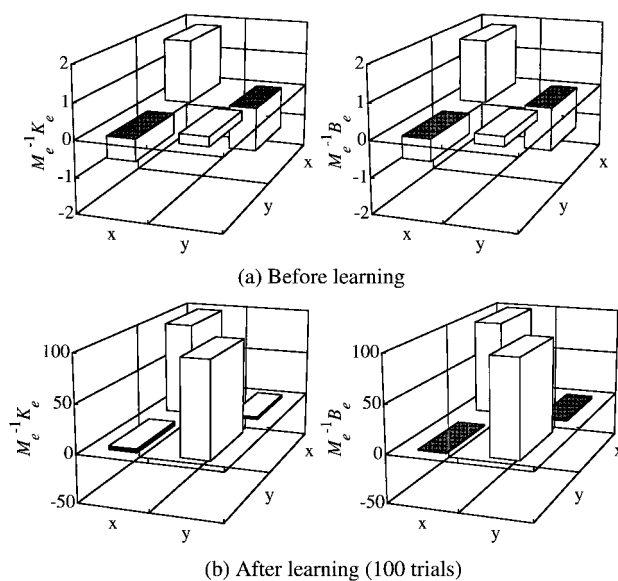


Fig. 8 Impedance parameters before and after learning of a free movement

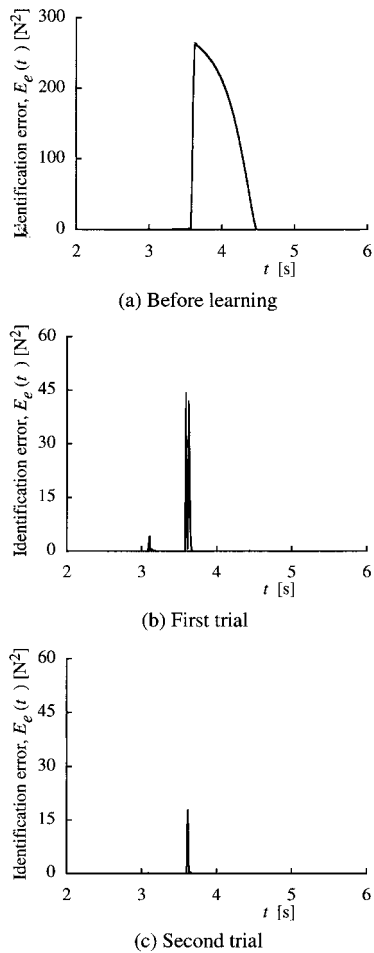


Fig. 9 Identification error of the environment during learning of a constrained movement

学習と目標軌道の修正を行う。

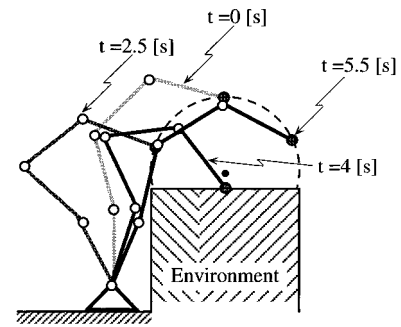
FCN および EIN は 4 層構造で、ユニット数は入力層が FCN は 10, EIN は 8, 中間層  $20 \times 2$  層, 16, 出力層 4 である。学習率は  $\eta_f^{MAX} = 0.0001$ ,  $\eta_d^{MAX} = 0.01$ ,  $p = 10$ ,  $\eta_e = 0.001$  で、手先の目標接触力は  $F_d = (0, 5)^T$  [N] とした。FCN の学習では、学習則式 (31) (32) の中に環境の特性が含まれることになる。後述するように、シミュレーションでは非常に剛性の高い環境を用いているため、この部分が等価的に学習率として働き、その結果 FCN の学習率は自由運動時の PCN, VCN の学習率に比較して小さな値となっている。

環境の特性は  $x < -0.1$  [m] では  $K_e = \text{diag.} [0, 10000]$  [N/m],  $B_e = \text{diag.} [0, 20]$  [Ns/m],  $M_e = \text{diag.} [0, 0.1]$  [kg] で環境モデル  $g_m$  (式 (24)) と等しいが、 $x \geq -0.1$  [m] ではモデル化誤差が存在し、非線形特性を有するものとする。

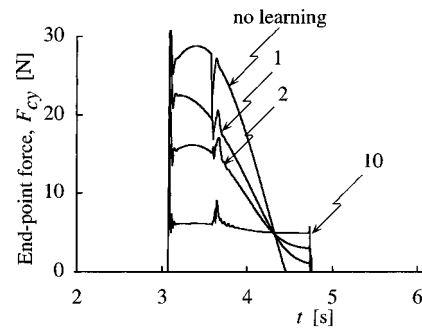
$$F_{cy} = K_{cy} dy_o^2 + B_{cy} dy_o^2 + M_{cy} d\ddot{y}_o \quad (37)$$

ただし、 $F_{cy}$  は環境から手先に働く法線方向の力で、接線方向は  $F_{cx} = 0$ , また各パラメータは  $K_{cy} = 1,000,000$  [N/m<sup>2</sup>],  $B_{cy} = 2,000$  [Ns/m<sup>2</sup>],  $M_{cy} = 0.1$  [kg] とした。

Fig. 9 に EIN の学習によって得た  $\hat{F}_c$  と実際の接触力  $F_c$  と



(a) Stick pictures



(b) End-point force

Fig. 10 End-point force of the manipulator during learning of a contact movement

の誤差の 2 乗和  $E_e(t)$  (式 (26)) の時間変化を示す。(a) は学習前, (b) は 1 回目, (c) は 2 回目の作業時のものである。学習前では非常に大きな誤差が生じているのに対し、学習を行うと、手先が環境に接触する瞬間および、環境の特性が不連続に変化する瞬間にはある程度の誤差が生じているが、学習を重ねるにつれて誤差が減少している様子が分かる。

Fig. 10 にマニピュレータの姿勢と手先接触力  $F_c$  の変化を示す。学習が十分に進むまでは自由運動で学習した大きなインピーダンスを有しているため、接触の際に大きな接触力を生じる。ところが、FCN の学習と目標軌道の修正が進むにつれ、しだいに目標接触力 (5 [N]) に近づき、学習後は接触した瞬間から目標接触力が実現されていることが分かる。しかしながら、環境特性が不連続に変化する瞬間には、EIN の学習に誤差が含まれるため手先接触力に制御誤差が残っている。

また、Fig. 11 に 1 回目と 10 回目の作業時の FCN の出力であるインピーダンスパラメータ  $M_e^{-1}$  の時間変化を示す。図中の  $(i, j)$  は行列  $M_e^{-1}$  の  $(i, j)$  要素を表している。y 方向のモビリティである  $(2, 2)$  要素以外は非常に小さな値に収束している。一方、学習後の  $(2, 2)$  要素の時間変化から、慣性パラメータは接触面の法線方向 (y 軸方向) の値が環境と接触している間のみ減少していることが分かる。拘束運動の学習中は PCN, VCN の出力であるインピーダンスパラメータ  $M_e^{-1} K_e$ ,  $M_e^{-1} B_e$  の学習は行わないため、これらの出力はほぼ一定に保たれる。したがって環境と接触している間は法線方向の粘弾性  $K_e$ ,  $B_e$  も小さくなり手先をやわらかくしていることになる。

最後に、Fig. 12 に目標軌道の変化を示す。接触力が目標接

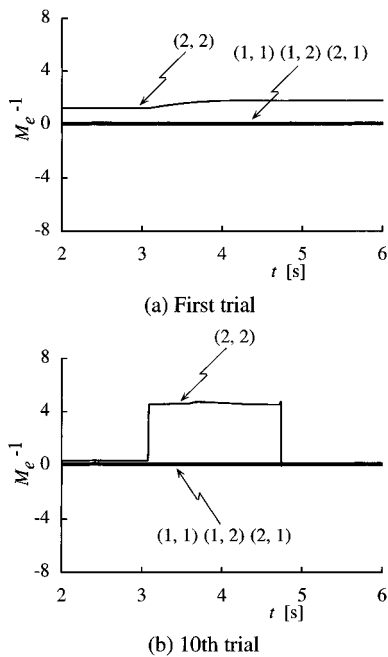


Fig. 11 Impedance parameters before and after learning of a contact movement

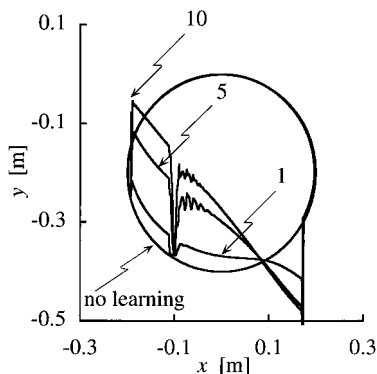


Fig. 12 Virtual trajectories of the manipulator during learning

触力より大きい場合は、目標軌道は環境の表面に近づく方向に修正され、接触力が目標接触力より小さい場合は、環境の表面から遠ざかる方向に修正されている。これにより、手先が環境に接触した時点から離れる瞬間まで目標接触力を実現するように目標軌道が修正されていることが分かる。

以上、非線形な特性を含む未知の環境に接触した場合でも、EINで環境を同定し、オンラインで手先インピーダンスの調節が実現できることを示した。

## 5. ま と め

本論文では、NNを利用してマニピュレータのインピーダンスパラメータをオンラインで学習的に調節する方法を提案した。オンラインでの学習を可能とするためには適切な学習則の導出が必要であるが、本手法ではインピーダンスパラメータを調節するNNとは別に、環境の非線形特性を同定するNNを導入することでオンラインによるインピーダンスパラメータの学習を

実現した。インピーダンスパラメータの学習は、追値制御誤差と力制御誤差を減少させるように行い、剛性、粘性、慣性の全パラメータを調節することができる。さらに同時に目標軌道の修正を積極的に導入することで、手先の目標接触力の実現可能性を高めている。

本論文で提案したオンライン学習法と文献[20][21]のオフライン学習法との最も大きな違いは、繰り返し試行が本質的に必要かどうかという点にある。文献[20][21]のオフライン学習法では誤差の勾配方向を前回の試行結果から計算しているため、必ず繰り返し試行を行う必要がある。一方、本手法では誤差の勾配方向をオンラインで計算することができるため、NNの学習が効率よく進めば必ずしも繰り返し試行を必要としない。例えば自由運動の学習(Fig.7)では1回目の試行から目標とする円軌道がほぼ実現されている。しかしながら、拘束運動の学習(Fig.9~12)においては学習が収束するのに10回程度の繰り返し試行が必要であった。ただこの場合でも各試行における初期状態に関する条件が大きく異なることに注意されたい。オフライン学習法ではマニピュレータの初期状態を試行ごとに完全に一致させる必要があるが、オンライン学習法の場合は異なっても構わない。現実の場面を考えると試行ごとの初期状態を常に一定に保つというのはかなり厳しい条件であり、この点は本論文で提案するオンライン学習法の大きな利点であろう。

この学習収束に必要な繰り返し回数を削減し、より効率よくNNの学習を進めるためには、NN自体の工夫が必要であると考えている。本論文では制御系の構成法や学習則を工夫することにより、一般的な誤差逆伝搬型NNでもかなり効率的なロボットインピーダンスの学習が可能であることを示した。しかしながらより効果的な学習や実機による学習実験を実現するためには、各NNの構造自体もロボットインピーダンスに適したものにしていく必要がある。今後は、インピーダンス制御に適したNNの構造、効果的な学習率決定法などについて考察するとともに、より複雑な作業を対象にして本手法を改良していく予定である。

## 参 考 文 献

- [1] N. Hogan: "Impedance Control: An Approach to Manipulation, Parts I, II, III," ASME Journal of Dynamic Systems, Measurement and Control, vol.107, no.1, pp.1-24, 1985.
- [2] Z.W. Luo and M. Ito: "Control design of robot for compliant manipulation on dynamic environments," IEEE Trans. on Robotics and Automation, vol.9, no.3, pp.286-296, 1993.
- [3] 山本, 羅, 伊藤: "タスクインピーダンス構造の実現および調節に関する考察", 第14回日本ロボット学会学術講演会予稿集, no.2, pp.621-622, 1997.
- [4] 鶴田, 光岡, 福田, 栗原: "神経回路モデルによるロボットの力制御(1自由度マニピュレータの制御)", 日本ロボット学会誌, vol.7, no.1, pp.47-51, 1989.
- [5] 福田, 栗原, 鶴田, 光岡: "神経回路網によるロボット・マニピュレータの位置・力制御の研究(第1報)", 機械学会論文誌, 56-527, C, pp.210-216, 1990.
- [6] 福田, 柴田, 鶴田, 光岡: "神経回路網によるロボット・マニピュレータの位置・力制御の研究(第2報)", 機械学会論文誌, 57-535, C, pp.866-873, 1991.
- [7] 柴田, 福田, 小菅, 新井, 鶴田, 光岡: "神経回路網によるロボット・マニピュレータの位置・力制御の研究(第6報)", 機械学会論文誌, 58-549, C, pp.108-115, 1992.

- [8] H.-L. Pei, Q.-J. Zhou and T.P. Leung: "A Neural Network Robot Force Controller," Proc. of the 1992 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp.1974-1979, 1992.
- [9] T.H. Connolly and F. Pfeiffer: "Neural Network Hybrid Position/Force Control," Proc. of the 1993 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp.240-244, 1993.
- [10] J.M. Tao and J.Y. S. Luh: "Application of Neural Network with Real-Time Training to Robust Position/Force Control of Multiple Robots," Proc. of IEEE Int. Conf. on Robotics and Automation, pp.142-148, 1993.
- [11] 木口, D.S. Neculescu, 福田: "ニューラルネットワークを利用した未知環境でのハイブリッド制御", 日本ロボット学会誌, vol.13, no.2, pp.291-296, 1995.
- [12] G. Gomi and M. Kawato: "Neural Network Control for a Closed Loop System Using Feedback-Error-Learning," Neural Networks, vol.6, no.7, pp.933-946, 1993.
- [13] D. Katić and M. Vukobratović: "Learning Impedance Control of Manipulation Robots by Feedforward Connectionist Structures," Proc. of IEEE Int. Conf. on Robotics and Automation, pp.45-50, 1994.
- [14] C.C. Cheah and D. Wang: "Learning Impedance Control for Robotic Manipulators," Proc. of IEEE Int. Conf. on Robotics and Automation, pp.2150-2155, 1995.
- [15] S. Jung and T.C. Hsia: "On Neural Network Application to Robust Impedance Control of Robot Manipulators," Proc. of IEEE Int. Conf. on Robotics and Automation, pp.869-874, 1995.
- [16] S.-T. Lin and H.-C. Tsai: "Impedance Control with On-Line Neural Network Compensator for Dual-Arm Robots," J. of Intelligent and Robotic Systems, vol.18, pp.87-104, 1997.
- [17] 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, pp.1237-1244, 1990.
- [18] M. Cohen and T. Flash: "Learning Impedance Parameters for Robot Control Using an Associative Search Networks," IEEE Trans. on Robotics and Automation, vol.7, no.3, pp.382-390, 1991.
- [19] B.-H. Yang and H. Asada: "Progressive Learning and Its Application to Robot Impedance Learning," IEEE Trans. on Neural Networks, vol.7, no.4, pp.941-952, 1996.
- [20] 辻, 西田, 伊藤: "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", 計測自動制御学会論文集, vol.28, no.12, pp.1461-1468, 1992.
- [21] T. Tsuji, K. Ito and P.G. Morasso: "Neural Network Learning of Robot Arm Impedance in Operational Space," IEEE Trans. on Systems, Man and Cybernetics, Part B: Cybernetics, vol.26, no.2, pp.290-298, 1996.
- [22] D.E. Rumelhart, G.E. Hinton and R.J. Williams: "Learning Representations by Error Propagation," in Parallel Distributed Processing, MIT Press, vol.1, pp.318-362, 1986.
- [23] 徐, 辻, 金子: "ニューラルネットに基づく適応制御を用いたフレキシブルビームのトルク制御", 計測自動制御学会論文集, vol.33, no.2, pp.109-117, 1997.
- [24] 辻, ヤジディ: "冗長マニピュレータのインピーダンス制御: 冗長性を利用した関節インピーダンスの調節法", 日本ロボット学会誌, vol.12, no.4, pp.609-615, 1994.
- [25] T. Tsuji, A. Jazidie and M. Kaneko: "Multi-Point Impedance Control for Redundant Manipulators," IEEE Trans. on Systems, Man and Cybernetics, Part B: Cybernetics, vol.26, no.5, pp.707-718, 1996.
- [26] 吉川: ロボット制御基礎論. コロナ社, 1988.



辻 敏夫 (Toshio Tsuji)

1959年12月25日生. 1985年広島大学大学院工学研究科博士課程前期修了. 同年同大学工学部助手, 1994年同助教授, 現在に至る. 工学博士. 人間とロボットの運動制御, ニューラルネット, マン・マシンシステムなどの研究に従事. 計測自動制御学会, 日本機械学会, 電気学会, 電子情報通信学会などの会員.

(日本ロボット学会正会員)



原田 研介 (Kensuke Harada)

1968年9月28日生. 1997年3月京都大学大学院工学研究科機械工学専攻博士後期課程修了. 博士(工学). 同年4月広島大学助手, 現在に至る. ロボットハンド, ロボットマニピュレータの力学と制御に関する研究に興味を持つ. IEEE, 計測自動制御学会, 日本機械学会などの会員.

(日本ロボット学会正会員)



赤松寛優 (Hiromasa Akamatsu)

1974年1月1日生. 1996年3月, 広島大学工学部第二類(電気系)卒業. 1998年3月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4月京セラ株式会社入社. 広島大学在学中, マニピュレータのインピーダンス制御の研究に従事.



金子 真 (Makoto Kaneko)

1954年1月18日生. 1981年3月東京大学工学系研究科博士課程修了. 工学博士. 同年4月通産省工業技術院機械技術研究所入所. 1990年4月, 九州工業大学情報工学部助教授. 1993年10月広島大学教授, 現在に至る. ロボットハンド, 力覚センサ, 触覚ベースアクティブセンシングなどの研究に興味を持つ. IEEE, 計測自動制御学会, 日本機械学会などの会員.

(日本ロボット学会正会員)