

# 神経回路により出力誤差のフィードバックを行う 逆運動学問題の解法

大山 英明\*<sup>1</sup> 舘 暲\*<sup>2</sup>

## Methods for Solving Inverse-kinematics Problems Using Nerual Networks with Output Error Feedback

Eimei Oyama\*<sup>1</sup> and Susumu Tachi\*<sup>2</sup>

Many studies on the learning control of the robot arm have been conducted by using neural newtworks. The method that uses an acquired inverse-kinematics model of the arm by learning are popular. However, acquisition of the inverse-kinematics model has a number of drawbacks. Furthermore, a limited scale neural networks system has only limited precision. Errors still remains in the output of the inverse-kinematics model using the neural networks system. In this paper, a new method for solving inverse-kinematics problem using the learned inverse model of the linearized model as output feedback system is proposed. Two possible configurations of the system are presented. The use of linear adaptive systems including Kalman filter is also proposed for higher accuracy. The performances of the proposed methods are shown by numerical simulations.

**Key Words:** Inverse-kinematics, Neural Networks, Output Feedback Inverse Model, Linear Adaptive System

### 1. はじめに

ロボットアームの逆運動学解を求める問題は、ロボットの制御において重要課題の一つとして研究されている。一般にこの分野では、アームの正確な幾何学モデルの存在を前提に、方程式を解くことによって、あるいは数値計算によって、その解を求める方法が検討されてきた。しかし、このような方法は、例えば、アームが何らかの物体と衝突して、それが変形してしまったような場合、モデル自体が不正確になり、解の精度が保証されなくなる。一方、生体はその構造に何らかの変化が起こり、特性が変化しても、それがある範囲内であれば、学習によって適応することができる。人間の学習機能を解明するために、また、ロボットに人間のような適応性を持たせるために、神経回路を利用した学習制御の研究が行われており、逆運動学解や逆動力学解を求める問題へもその応用が試みられている[1]~[4]。

システムの逆モデルの獲得法として、直接逆モデリング(Direct Inverse Modeling)が提案され、特に逆運動学モデルに関しては、阪口[5]、橋本ら[6]によってロボットアームの

運動制御に応用された。しかし、直接逆モデリングは、多対一対応システムに対して適用できないことが指摘され、その後、多対一対応システムについても適用可能な手法として、Jordanらの順逆モデリング[2]や川人らのフィードバック誤差学習則[1]、Colombanoらの目標指向モデル反転(Goal Directed Model Inversion)[4]などの逆モデルの学習法が提案された。田中らは、フィードバック誤差学習則を拡張して、冗長アームの逆運動学モデルの学習を行っている[7]。

直接逆モデリングや、順逆モデリング、目標指向モデル反転等を用いた手法では、神経回路によって逆モデルを直接学習し、制御に際しては学習で獲得した逆モデルを対象システムに直列に接続して用いている。その結果システム構成が、オープンループとなるため、逆モデルにモデル化誤差がある場合、出力に誤差が残る。逆モデルとフィードバック回路を併用したフィードバック誤差学習則に基づく学習システム[1]では、モデル化誤差を出力誤差のフィードバックにより、補償することが可能であるが、既に獲得したフィードバック回路の存在を前提としている。Funらは順モデルを利用して、出力誤差の最急降下方向を計算し、これを利用してロボットアームの制御を行うことを提案した[8]。この手法を利用すれば、フィードバック回路を構成できるが、バックプロパゲーション学習の信号を外部に取り出し、制御に用いることが可能な多層神経回路にしか適用できない。斉藤らも順モデルを利用した逆運動学解法を

原稿受付 1994年3月25日

\*<sup>1</sup>工業技術院機械技術研究所

\*<sup>2</sup>東京大学工学部

\*<sup>1</sup>Mechanical Engineering Laboratory

\*<sup>2</sup>University of Tokyo

提案しているが、順モデルからヤコビ行列を計算するための機構が必要である[9]。順モデルの数値微分によって、ヤコビ行列の推定が可能であるが、計算に時間がかかる場合が多い。

本研究では、特性が未知のロボットアームに対して、一般的な学習要素を用いて逆運動学問題を解くために、出力誤差のフィードバック回路を学習要素によって獲得する手法を提案する。筆者らは、静的な逆問題の解法システムとして、出力誤差のフィードバック系を学習要素によって構成し、反復改良演算によって逆問題を解く、出力フィードバック型逆モデルを提案している[10]~[12]。この解法システムの手法を利用して、出力誤差のフィードバックを行い、運動学特性が未知のロボットアームの逆運動学問題を解く手法を提案する。さらに、解の高精度化のためにヤコビ行列を逐次的に推定する線形適応系との複合システムを提案する。提案法の有効性を数値シミュレーションによって評価する。

## 2. 逆運動学問題

関節角の位置制御もしくは速度制御を行うロボットアームを考える。アーム先端位置ベクトルを  $\mathbf{x}$  ( $n$ 次元)、関節角ベクトルを  $\boldsymbol{\theta}$  ( $m$ 次元,  $n \leq m$ ) とし、アームの順運動学が次式で与えられるものとする。

$$\dot{\mathbf{x}} = \mathbf{f}(\boldsymbol{\theta}) \quad (1)$$

ここで  $\mathbf{f}(\boldsymbol{\theta})$  は  $n$ 次元ベクトル関数である。このシステムのヤコビ行列を  $\mathbf{J}(\boldsymbol{\theta})$  とする。

$$\mathbf{J}(\boldsymbol{\theta}) = \partial \mathbf{f}(\boldsymbol{\theta}) / \partial \boldsymbol{\theta} \quad (2)$$

離散的に与えられるアーム先端位置の目標値  $\mathbf{x}_d(k)$  に、実際の先端位置  $\mathbf{x}(k)$  を追従させるための関節角位置指令  $\boldsymbol{\theta}_d(k)$  を計算する問題を考える。単純化のため、ダイナミクスを無視し、関節角位置指令  $\boldsymbol{\theta}_d(k)$  は完全に実現できるものと仮定する。

冗長アームに対しては、連続的に変化する目標軌道  $\mathbf{x}_d(k)$  を実現するような関節角ベクトル  $\boldsymbol{\theta}_d(k)$  を逐次的に計算する手法が用いられることが多い。例えば、以下のような関節角ベクトルの更新式がある。

$$\begin{aligned} \boldsymbol{\theta}(k+1) &= \boldsymbol{\theta}_d(k+1) = \boldsymbol{\theta}(k) + \Delta \boldsymbol{\theta}(k) \\ &= \boldsymbol{\theta}(k) + \mathbf{J}^*(\boldsymbol{\theta}(k)) \Delta \mathbf{x}_d'(k) \end{aligned} \quad (3)$$

$$\Delta \mathbf{x}_d'(k) = \Delta \mathbf{x}_d(k) + \mathbf{K}(\mathbf{x}_d(k) - \mathbf{x}(k)) \quad (4)$$

$$\Delta \mathbf{x}_d(k) = \mathbf{x}_d(k+1) - \mathbf{x}_d(k) \quad (5)$$

$\mathbf{J}^*(\boldsymbol{\theta})$  は、冗長でないアームの場合は  $\mathbf{J}(\boldsymbol{\theta})$  の逆行列である。冗長アームの場合は  $\mathbf{J}(\boldsymbol{\theta})$  の一般化逆行列であり、 $\mathbf{J}(\boldsymbol{\theta})$  の疑似逆行列

$$\mathbf{J}^*(\boldsymbol{\theta}) = \mathbf{J}(\boldsymbol{\theta})^T (\mathbf{J}(\boldsymbol{\theta}) \mathbf{J}(\boldsymbol{\theta})^T)^{-1} \quad (6)$$

を用いて、

$$\mathbf{J}^*(\boldsymbol{\theta}) = \mathbf{J}^*(\boldsymbol{\theta}) + (\mathbf{I} - \mathbf{J}^*(\boldsymbol{\theta}) \mathbf{J}(\boldsymbol{\theta})) \mathbf{G} \quad (7)$$

と表すことができる。 $\mathbf{G}$  は適当に選んだ  $m$ 行  $m$ 列の行列である。

式(4)の  $\mathbf{K}$  は出力誤差のフィードバックゲインである。このシステムでは、ゲイン  $\mathbf{K}$  を適切に選ぶことにより、先端位置  $\mathbf{x}(k)$  を目標位置  $\mathbf{x}_d(k)$  に追従させることができる。 $\mathbf{K}$  とし、例えば単位行列を選ぶことができる。冗長アームの場合、

$$\begin{aligned} \boldsymbol{\theta}(k+1) &= \boldsymbol{\theta}(k) + \mathbf{J}^*(\boldsymbol{\theta}(k)) \Delta \mathbf{x}_d'(k) \\ &+ (\mathbf{I} - \mathbf{J}^*(\boldsymbol{\theta}(k)) \mathbf{J}(\boldsymbol{\theta}(k))) [\partial \mathbf{S}(\boldsymbol{\theta}) / \partial \boldsymbol{\theta}]_{\boldsymbol{\theta}=\boldsymbol{\theta}_k} \end{aligned} \quad (8)$$

とすることによって、スカラー関数  $\mathbf{S}(\boldsymbol{\theta})$  を増大させるような軌道を生成できる[13][14]。

式(3)において、対象とするアームの特性が影響するのは、右辺第二項  $\mathbf{J}^*(\boldsymbol{\theta}(k)) \Delta \mathbf{x}_d'(k)$  のみである。本研究では、この項を学習要素を利用して構成することにより、特性が未知のロボットアームの先端位置制御を行うことを提案する。具体的な構成法については、3章で述べる。

## 3. フィードバック回路の構成法

### 3.1 出力フィードバック型逆モデル

筆者らは、静的な逆問題の解法システムとして、対象システムの線形化モデルの逆モデルを学習要素によって構成し、これを利用して、出力誤差をフィードバックし、反復改良演算によって逆問題を解く、出力フィードバック型逆モデルを提案している[10][11](Appendix.1参照)。また、この解法システムの改良案であるゲイン学習方式を提案している[12]。この手法を利用して、冗長アームの先端位置フィードバック回路を学習によって構成できる。その構成法には二種類がある。第一の構成法は、学習要素によってフィードバック回路を直接学習する方式であり、これを直接学習方式(Direct Learning Method)と呼んでいる。第二の構成法は、神経回路によってヤコビ行列  $\mathbf{J}(\boldsymbol{\theta})$  を学習し、その逆行列や疑似逆行列を計算し、先端位置誤差に乗じることによってフィードバック信号を計算する方式であり、これを間接ゲイン学習方式(Indirect Gain Learning Method)と呼んでいる。間接ゲイン学習方式は、逆行列や疑似逆行列の計算が必要であるという問題点があるが、オフライン学習において非常に有利であり、また柔軟なフィードバック回路の構成が可能である。

### 3.2 直接学習方式

本研究では、学習要素  $\boldsymbol{\Phi}(p)$  は任意の連続関数を近似できるものと仮定する。理想的な学習要素では、入力が  $\mathbf{p}$  である時、教師信号  $\mathbf{q}_j$  が与えられた場合、最終的に、

$$\boldsymbol{\Phi}(\mathbf{p}) = \mathbf{E}(\mathbf{q}_j) \quad (9)$$

となるように、学習が行われる。 $\mathbf{E}(\mathbf{s})$  はベクトル  $\mathbf{s}$  の平均値を示す。現在、提案されている神経回路モデルの多くはこの性質を近似的に満たしている。そのような学習要素としては、例えば、AlbusのCMAC[15]、Kohonenのトポロジカルマッピング[16]、バックプロパゲーション学習を行う多層神経回路[17]等がある。

神経回路により構成される非線形学習要素への入力を  $(\boldsymbol{\theta}(k), \Delta \mathbf{x}_d(k))$  とし、その  $m$ 次元出力ベクトルを  $\boldsymbol{\Phi}(\boldsymbol{\theta}(k), \Delta \mathbf{x}_d(k))$  とする。直接学習方式では、学習要素の入出力関係が

$$\boldsymbol{\Phi}(\boldsymbol{\theta}, \Delta \mathbf{x}) = \mathbf{J}^*(\boldsymbol{\theta}) \Delta \mathbf{x} \quad (10)$$

となるように学習を行い、式(4)の  $\Delta \mathbf{x}_d'(k)$  を利用して、関節角ベクトルの更新演算を次式のように行う[11][12]。

$$\begin{aligned} \boldsymbol{\theta}(k+1) &= \boldsymbol{\theta}(k) + \Delta \boldsymbol{\theta}(k) \\ &= \boldsymbol{\theta}(k) + \boldsymbol{\Phi}(\boldsymbol{\theta}(k), \Delta \mathbf{x}_d'(k)) - \boldsymbol{\Phi}(\boldsymbol{\theta}, 0) \\ &= \boldsymbol{\theta}(k) + \mathbf{J}^*(\boldsymbol{\theta}(k)) \Delta \mathbf{x}_d'(k) \end{aligned} \quad (11)$$

これにより、式(4)のアーム先端位置制御系を近似的に実現で

きる。Fig. 1 に、直接学習方式を利用した、ロボットアームの先端位置制御系の構成を示す。図中の  $z^{-1}$  は信号の増分を計算する演算を示す。式(11)の安定性については、Appendix. 2 に示す。

前述のように、式(10)の  $J^*(\theta)$  は  $J(\theta)$  の一般化逆行列である。  $J(\theta)$  の一般化逆行列は無数に存在し、オフライン学習の場合、目的に応じて自由に  $J^*(\theta)$  を設定することができる。一方、オンライン学習の場合、学習される  $J^*(\theta)$  が、どのような一般化逆行列になるかは、学習要素  $\Phi(\theta, \Delta x)$  の初期状態と学習に用いたデータに依存し、自由な設定は困難である。

まず、アームを実際の作業に用いる前に行うオフライン学習における教師信号について述べる。いろいろな関節角  $\theta$  において、関節角ベクトルの各成分を順に微小量変化させた時のアーム先端の位置ベクトルの微小変化を観測すれば、差分近似により、ヤコビ行列  $J(\theta)$  の近似値を得ることが可能であり、目的に応じて、ヤコビ行列の一般化逆行列  $J^*(\theta)$  を設定し、式(10)の右辺を直接計算し、教師信号とする。

実際の作業中に行うオンライン学習では、関節角の微小変化  $\Delta\theta$  とそれに対応する出力の微小変化  $\Delta x$  に着目して、学習要素の学習を行う。両者には、次式のような線形の関係がある。

$$\Delta x = f(\theta + \Delta\theta) - f(\theta) \doteq J(\theta)\Delta\theta \quad (12)$$

十分小さいパラメータ  $r$  について、

$$|\Delta\theta| < r \quad (13)$$

が成立する時、  $\Phi(\theta, \Delta x)$  の教師信号  $\Phi'(\theta, \Delta x)$  を

$$\Phi'(\theta, \Delta x) = \Delta\theta \quad (14)$$

とする。この学習によって、

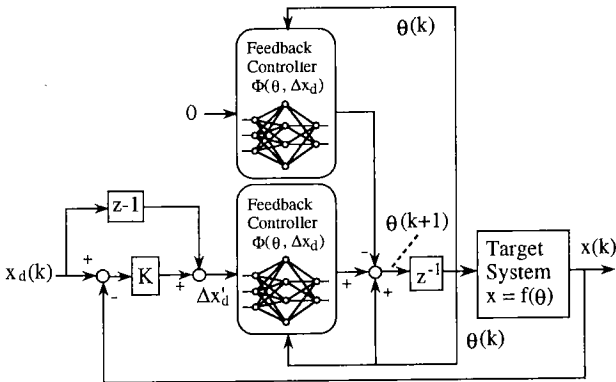


Fig. 1 System configuration of Direct Learning method

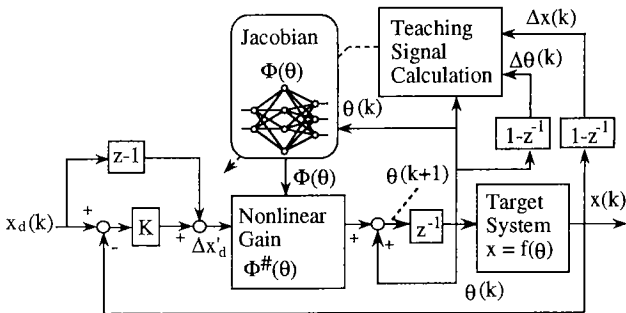


Fig. 2 Indirect Gain Learning method

$$\Phi(\theta, \Delta x) - \Phi(\theta, 0) \doteq J^*(\theta)\Delta x \quad (15)$$

を実現できる (Appendix. 1 参照)。

なお、正規乱数  $\omega$  と適当な係数  $\beta$  を利用して、次式のような学習を行う方が、高速に学習を行うことができる [12]。

$$\begin{aligned} \Phi(\theta, \lambda\Delta x) &= \lambda\Delta\theta \\ \lambda &= \beta\omega/|\Delta\theta| \end{aligned} \quad (16)$$

### 3.3 間接ゲイン学習方式

Fig. 2 に間接ゲイン学習方式の構成を示す。間接ゲイン学習方式では、神経回路により構成される非線形学習要素の出力  $\Phi(\theta(k))$  が  $n$  行  $m$  列のヤコビ行列の近似値  $J_a(k)$  を構成するものとし、

$$J_a(k) = \Phi(\theta(k)) \doteq J(\theta(k)) \quad (17)$$

となるように学習を行い、その疑似逆行列

$$\begin{aligned} \Phi^*(\theta(k)) &= \Phi(\theta(k))^T (\Phi(\theta(k))\Phi(\theta(k))^T)^{-1} \\ &\doteq J^*(\theta(k)) \end{aligned} \quad (18)$$

を計算し、次式のように、関節角ベクトルを更新する。

$$\begin{aligned} \theta(k+1) &= \theta(k) + \Delta\theta(k) \\ &= \theta(k) + \Phi^*(\theta(k))\Delta x'_d(k) \\ &\doteq \theta(k) + J^*(\theta(k))\Delta x'_d(k) \end{aligned} \quad (19)$$

近似ヤコビ行列  $J_a(k) = \Phi(\theta(k))$  を用いて、式(8)のように、冗長自由度を積極的に利用した制御も可能である。式(19)の安定性については、Appendix. 2 に示す。

オフライン学習においては、直接学習方式と同様、いろいろな姿勢において、アームの関節角を微小変化させ、先端位置の変化を観測し、差分近似によって、ヤコビ行列  $J(\theta)$  を計算し、教師信号とする。間接ゲイン学習方式では、学習要素の入力の次元は、直接学習方式に比べて少ないために、必要な教師データ数も少なく、高速に学習を行うことができるという特徴がある [12]。

オンライン学習においては、式(13)が成立する時、  $\Delta\theta$  に対応するアーム先端位置の変化を  $\Delta x$  とすると、学習要素の出力  $\Phi(\theta)$  をヤコビ行列  $J(\theta)$  に近づける教師信号  $\Phi'(\theta)$  を、Broyden の近似ヤコビ行列の更新式 [18] を利用して、次式のように計算する。

$$\Phi'(\theta) = \Phi(\theta) + (\Delta x - \Phi(\theta)\Delta\theta)\Delta\theta^T/|\Delta\theta|^2 \quad (20)$$

さらに、オンライン学習時において、

$$\Delta\theta = \Phi^*(\theta)\Delta x'_d \quad (21)$$

とすれば、式(20)は、次式のように簡略化できる。

$$\Phi'(\theta) = \Phi(\theta) + (\Delta x - \Delta x'_d)\Delta\theta^T/|\Delta\theta|^2 \quad (22)$$

## 4. 線形適応系との組み合わせ

多層神経回路をはじめとする非線形学習要素の場合、通常学習に時間がかかるという弱点がある。本章では、より短期間の学習によって、より高精度の解を得るために、間接ゲイン学習方式と線形適応系とを組み合わせ合わせた複合システムを提案する。

### 4.1 Broyden の更新式に基づいた線形適応系

Broyden の準ニュートン法では、対象システムのヤコビ行列の近似値  $J_a(k)$  を利用して、非線形方程式を解く [18]。  $J_a(k)$  は、入力変化  $\Delta\theta(k)$  とそれに対応する出力の変化  $\Delta x(k)$  によって、式(20)と同様、次式のように更新される。

$$\begin{aligned} \mathbf{J}_a(k+1) &= \mathbf{J}_a(k) \\ &+ (\Delta \mathbf{x}(k) - \mathbf{J}_a(k) \Delta \boldsymbol{\theta}(k)) \Delta \boldsymbol{\theta}(k)^T / |\Delta \boldsymbol{\theta}(k)|^2 \end{aligned} \quad (23)$$

このような線形適応系を、間接ゲイン学習方式に利用することを考える。対象システムのヤコビ行列の近似行列  $\mathbf{J}_a(k)$  を、非線形学習要素と  $nm$  個の適応パラメータによって構成される  $n$  行  $m$  列の行列  $\Delta \mathbf{J}_a(k)$  の和によって、次式のように構成する。

$$\mathbf{J}_a(k) = \boldsymbol{\Phi}(\boldsymbol{\theta}(k)) + \Delta \mathbf{J}_a(k) \quad (24)$$

関節角ベクトル  $\boldsymbol{\theta}(k)$  の更新は

$$\begin{aligned} \boldsymbol{\theta}(k+1) &= \boldsymbol{\theta}(k) + \Delta \boldsymbol{\theta}(k) \\ &= \boldsymbol{\theta}(k) + \mathbf{J}_a^*(\boldsymbol{\theta}(k)) \Delta \mathbf{x}(k) \end{aligned} \quad (25)$$

のように行う。さらに、 $\Delta \boldsymbol{\theta}(k)$  に対応するアーム先端の変化  $\Delta \mathbf{x}(k)$  を利用して、適応パラメータ行列  $\Delta \mathbf{J}_a(k)$  を

$$\begin{aligned} \Delta \mathbf{J}_a(k+1) &= \Delta \mathbf{J}_a(k) \\ &+ \alpha (\Delta \mathbf{x}(k) - \mathbf{J}_a(k) \Delta \boldsymbol{\theta}(k)) \Delta \boldsymbol{\theta}(k)^T / |\Delta \boldsymbol{\theta}(k)|^2 \end{aligned} \quad (26)$$

のように更新する。 $\alpha$  は  $0 < \alpha < 2$  を満たす係数である。非線形学習要素の学習は、3章で述べたように、式(20)に従って行い、学習要素の学習によって、 $\boldsymbol{\Phi}(\boldsymbol{\theta}(k))$  が、 $\boldsymbol{\Phi}^+(\boldsymbol{\theta}(k))$  に変化した場合には、

$$\Delta \mathbf{J}_a^*(k) = \Delta \mathbf{J}_a(k) - (\boldsymbol{\Phi}^+(\boldsymbol{\theta}(k)) - \boldsymbol{\Phi}(\boldsymbol{\theta}(k))) \quad (27)$$

とし、 $\Delta \mathbf{J}_a^*(k)$  を新たな  $\Delta \mathbf{J}_a(k)$  とし、 $\mathbf{J}_a(k)$  を一定に保つ。

#### 4.2 カルマンフィルタに基づく線形適応系

対象システムが線形系の場合、式(26)の適応則によって、十分な回数の更新を行えば真のヤコビ行列の推定が可能であるが、非線形系の場合、ヤコビ行列の変化が急であれば、真のヤコビ行列への追従は難しい。より高速にヤコビ行列を推定するために、カルマンフィルタの利用を提案する。ヤコビ行列や非線形学習要素の変化は通常連続であり、

$$\begin{aligned} \mathbf{J}(\boldsymbol{\theta} + \Delta \boldsymbol{\theta}) - \boldsymbol{\Phi}(\boldsymbol{\theta} + \Delta \boldsymbol{\theta}) &= \mathbf{J}(\boldsymbol{\theta}) - \boldsymbol{\Phi}(\boldsymbol{\theta}) \\ &+ [\partial \mathbf{J}(\boldsymbol{\theta}) / \partial \boldsymbol{\theta} - \partial \boldsymbol{\Phi}(\boldsymbol{\theta}) / \partial \boldsymbol{\theta}] \Delta \boldsymbol{\theta} + \dots \\ &= \mathbf{J}(\boldsymbol{\theta}) - \boldsymbol{\Phi}(\boldsymbol{\theta}) + O(\Delta \boldsymbol{\theta}) \end{aligned} \quad (28)$$

と展開できる。 $O(s)$  は、各方程式において、ベクトル  $\mathbf{s}$  に関して1次以上の項の総和を表すものとする。ここで、式(28)の  $O(\Delta \boldsymbol{\theta})$  を正規ノイズで近似することを提案する。 $O(\Delta \boldsymbol{\theta})$  の第  $i$  行  $j$  列成分を  $O_{ij}(\Delta \boldsymbol{\theta})$  とすると、その2乗の時間平均に着目し、

$$E(O_{ij}(\Delta \boldsymbol{\theta})^2) \leq q^2 |\Delta \boldsymbol{\theta}|^2 \quad (29)$$

を満たすパラメータ  $q$  を考える。十分小さな  $\Delta \boldsymbol{\theta}$  について、例えば、

$$\begin{aligned} q &= \gamma \sqrt{\max_{i,j} \left| \frac{\partial J_{ij}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} - \frac{\partial \Phi_{ij}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right|^2} \\ &\quad \gamma > 1 \end{aligned} \quad (30)$$

は式(29)を満たす。 $\gamma$  は完全率である。

パラメータ  $q$  と  $nm$  個の正規ノイズによって構成される  $n$  行  $m$  列の行列  $\boldsymbol{\omega}_{nm}$  によって、 $\Delta \mathbf{J}_a(k)$  の変化を

$$\Delta \mathbf{J}_a(k+1) = \Delta \mathbf{J}_a(k) + |\Delta \boldsymbol{\theta}(k)| q \boldsymbol{\omega}_{nm} \quad (31)$$

と近似的にモデル化する。明らかに、式(28)の右辺第三項は、正規ノイズではないが、式(31)は  $\Delta \mathbf{J}_a(k)$  のシステムダイナミクスの特徴を表現していると考えられる。ヤコビ行列の  $\boldsymbol{\theta}$  による微分  $[\partial \mathbf{J}(\boldsymbol{\theta}) / \partial \boldsymbol{\theta}]$  や  $[\partial \boldsymbol{\Phi}(\boldsymbol{\theta}) / \partial \boldsymbol{\theta}]$  についてのより詳しい知見があれば、より精密なモデル化も可能である。 $q$  の値は非線

形学習要素の学習が進むにつれ、小さくなるべきであるが、本稿のモデル化では定数とする。観測方程式は、観測ノイズの標準偏差  $r$  と  $n$  個の正規ノイズによって構成される  $m$  次元ベクトル  $\boldsymbol{\omega}_m$  によって、

$$\Delta \mathbf{x}(k) = \mathbf{J}_a(k) \Delta \boldsymbol{\theta}(k) + r \boldsymbol{\omega}_m \quad (32)$$

のようにモデル化する。

$\mathbf{J}_a(k)$  の  $j$  行ベクトルを  $\mathbf{J}_a(k)_{(j)}$  と表すことにする。各行ベクトルについて、システムダイナミクスは

$$\mathbf{J}_a(k+1)_{(j)}^T = \mathbf{J}_a(k)_{(j)}^T + |\Delta \boldsymbol{\theta}(k)| q \boldsymbol{\omega}_m \quad (33)$$

となり、観測方程式は

$$\Delta \mathbf{x}_j(k) = \Delta \boldsymbol{\theta}(k)^T \mathbf{J}_a(k)_{(j)}^T + r \omega \quad (34)$$

となる。 $\omega$  は正規ノイズである。推定すべき変数の個数は  $nm$  個であり、通常のカルマンフィルタの定式化では、 $nm$  行  $nm$  列の共分散行列が必要になる。しかしながら、各行ベクトルのための、システムダイナミクスと観測方程式の右辺は同一であり、各行ベクトルの成分に関する共分散行列の更新式は同一になる。 $nm$  行  $nm$  列の共分散行列の初期値が対角行列であるとすると、 $m$  行  $m$  列の対称行列  $\mathbf{P}(k)$  を用意することによって、以下のようなカルマンフィルタによる  $\Delta \mathbf{J}_a(k)$  の推定アルゴリズムを構成できる。

時刻  $k+1$  の時点において、観測による更新前の  $\Delta \mathbf{J}_a(k+1)$  の値を  $\Delta \mathbf{J}_a(k+1|-)$  とする。 $\mathbf{J}_a(k+1|-)$  や  $\mathbf{P}(k+1|-)$  も同様である。時間更新式は

$$\Delta \mathbf{J}_a(k+1|-) = \Delta \mathbf{J}_a(k) \quad (35)$$

$$\mathbf{J}_a(k+1|-) = \boldsymbol{\Phi}(\boldsymbol{\theta}(k) + \Delta \boldsymbol{\theta}(k)) + \Delta \mathbf{J}_a(k+1|-) \quad (36)$$

$$\mathbf{P}(k+1|-) = \mathbf{P}(k) + |\Delta \boldsymbol{\theta}(k)|^2 q^2 \mathbf{I} \quad (37)$$

となり、 $\Delta \mathbf{J}_a(k+1|-)$  の各行ベクトルを更新するためのカルマンゲインは次式のように計算される。

$$\begin{aligned} K(k+1) &= (\Delta \boldsymbol{\theta}(k)^T \mathbf{P}(k+1|-) \Delta \boldsymbol{\theta}(k) + r^2)^{-1} \\ &\quad \mathbf{P}(k+1|-) \Delta \boldsymbol{\theta}(k) \end{aligned} \quad (38)$$

$(\Delta \boldsymbol{\theta}(k)^T \mathbf{P}(k+1|-) \Delta \boldsymbol{\theta}(k) + r^2)$  はスカラーであり、カルマンゲイン  $K(k+1)$  の計算は容易である。 $\Delta \mathbf{x}(k)$  の観測による  $\Delta \mathbf{J}_a(k+1)$  の修正は次式のように行われる。

$$\begin{aligned} \Delta \mathbf{J}_a(k+1)^T &= \Delta \mathbf{J}_a(k+1|-)^T \\ &+ K(k+1) (\Delta \mathbf{x}(k) - \mathbf{J}_a(k+1|-) \Delta \boldsymbol{\theta}(k))^T \end{aligned} \quad (39)$$

$$\mathbf{P}(k+1) = \mathbf{P}(k+1|-) - K(k) \Delta \boldsymbol{\theta}(k)^T \mathbf{P}(k+1|-) \quad (40)$$

学習要素の学習後は、式(27)の更新演算を行う。

実際には、特性が未知の対象システムに関して、式(30)の右辺の計算は難しく、 $q$  の値の設定は難しい。適切でない  $q$  の値を用いると、推定精度は低下することになる。しかしながら、ある時点で  $q$  の値が小さすぎる場合でも、学習要素による非線形補償によって、最終的には、式(30)の右辺は設定した  $q$  よりも小さくなり、その前後の期間では  $q$  の値は適切である。 $q$  の値が無限大の極限では、式(39)は  $\alpha$  を1とした式(26)に一致する。 $q$  の値が大きすぎる場合でも、式(26)による推定よりは高精度の推定が期待できる。 $q$  のオンラインでの推定は今後の課題である。

対象システムの非線形性が小さい場合、線形適応系単独でも逆運動学解の計算が可能である。その場合の問題点は、対象システムの非線形性によるヤコビ行列の変化を、ノイズで近似しているため、高精度のヤコビ行列の推定は難しいことと、フィ

ードバックが断たれた場合に、適応パラメータの更新が不可能となり、逆運動学解の計算が不可能になることである。

### 5. シミュレーション

本システムの有効性を示すために、2次元平面上を動く3自由度アームにおいて、逆運動学問題の解法システムを学習により獲得するシミュレーションを行った。シミュレーションを行う逆運動学問題の解法システムは5種類である。直接学習方式(Direct Learning)、間接ゲイン学習方式(Indirect Gain Learning)、式(26)に基づく線形適応系を組み込んだ間接ゲイン学習方式(Indirect Gain Learning with Linear Adaptive System)、カルマンフィルタに基づく線形適応系を組み込んだ間接ゲイン学習方式(Indirect Gain Learning with Kalman Filter)に加えて、性能の比較のために、数値解法方式(Numerical Method)を用意した。数値解法方式は学習要素を使用せず、ヤコビ行列の疑似逆行列を計算によって求め、先端位置誤差のフィードバックを行う。以下、表や図中では、直接学習方式はDL、間接ゲイン学習方式はIGL、線形適応系を利用した間接ゲイン学習方式をIGLL、カルマンフィルタを利用した間接ゲイン学習方式をIGLK、数値解法はNMと略す。

アーム先端位置を高精度で検知する機能を有するハンド・アイシステムを想定する。2次元平面上で運動する、3自由度アームの関節角 $\theta=(\theta_1, \theta_2, \theta_3)$ とそのアーム先端位置 $x=(x, y)$ の関係を

$$\begin{aligned} x &= x_0 + L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2) \\ &\quad + L_3 \cos(\theta_1 + \theta_2 + \theta_3) \\ y &= y_0 + L_1 \sin(\theta_1) + L_2 \sin(\theta_1 + \theta_2) \\ &\quad + L_3 \sin(\theta_1 + \theta_2 + \theta_3) \end{aligned} \quad (41)$$

とする。Fig. 3に、アームの構成を示す。

$\theta_1$ の可動範囲を $(-30^\circ, 120^\circ)$ 、 $\theta_2$ の可動範囲を $(0^\circ, 150^\circ)$ 、 $\theta_3$ の可動範囲を $(-60^\circ, 60^\circ)$ とする。 $(x_0, y_0)=(0.2[\text{m}], 0[\text{m}])$ とし、アームの寸法を $L_1=0.3[\text{m}]$ 、 $L_2=0.25[\text{m}]$ 、 $L_3=0.15[\text{m}]$ とする。このアームに対し、アーム先端位置の目標軌道 $x_d(k)$ を実現する関節角ベクトルの軌道 $\theta(k)$ を計算する逆運動学問題について、解法システムの評価を行う。

制御システムとしての性能評価においては、以下に述べるような逆運動学問題を解く試行を行い、軌道の追従誤差を測定する。まず一様乱数によって関節角ベクトル $\theta_s$ を発生させる。その時のアーム先端位置を初期値 $x_s$ として定め、続いて、正

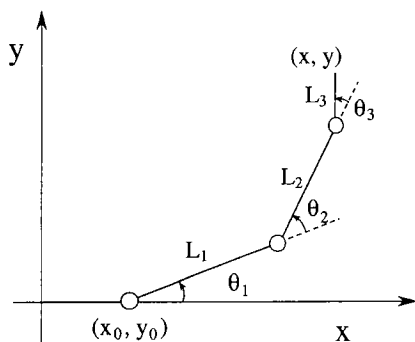


Fig. 3 Configuration of arm

規乱数 $\omega$ によってアーム先端位置 $x_s$ から円の中心 $x_c$ へ向かうベクトル $\Delta x_c$ の第 $i$ 成分を

$$\Delta x_{ci} = 0.05\omega \quad [\text{m}] \quad (42)$$

のように定め、次のように円の中心 $x_c$ を定める。

$$x_c = x_s - \Delta x_c \quad (43)$$

動作時間を $T$ として、反復 $k$ 回時の、円の中心回りの回転角 $\theta_c(k)$ を次のように定め、

$$\theta_c(k) = 2\pi k/T \quad (44)$$

$k=0, 1, \dots, T$ において、目標軌道を次式で生成する。

$$\begin{aligned} x_d(k) &= x_c + R(k)\Delta x_c \\ R(k) &= \begin{bmatrix} \cos \theta_c(k) & -\sin \theta_c(k) \\ \sin \theta_c(k) & \cos \theta_c(k) \end{bmatrix} \end{aligned} \quad (45)$$

乱数で発生させた関節角ベクトル $\theta_s$ をアームの初期状態の関節角ベクトル $\theta(0)$ として、提案した逆運動学解法システムによって、アーム先端を目標軌道へ追従させる。

軌道追従誤差 $e(k)$ を、

$$e(k) = x_d(k) - f(\theta(k)) \quad (46)$$

として、いろいろな目標軌道に対して先端を追従させる試行を行い、 $|e(k)|$ のRMS (Root Mean Square) 誤差と最大値によって、解法システムの性能を評価する。ただし、計算中に、関節角の可動範囲を越えた軌道や、実現不可能な軌道は除くものとする。

今回のシミュレーションにおいては、式(4)のゲイン $K$ を単位行列として、 $\Delta x'_d(k)$ を次式のように計算する。

$$\Delta x'_d(k) = x_d(k+1) - x(k) \quad (47)$$

また、 $T$ を100とする。

線形適応系を利用する間接ゲイン学習方式においては、各試行の開始時に

$$\Delta J_d(0) = 0 \quad (48)$$

とする。式(24)のパラメータ $\alpha$ の値は0.5とする。これは、シミュレーションの経験から、良好と考えられる値を選んだ。カルマンフィルタの共分散行列は、

$$P(0) = I \quad (49)$$

と初期化するものとする。今回のシミュレーションでは、観測ノイズを無視しているが、数値的な安定化のため、式(38)の $r$ を $1.0 \times 10^{-4}$ とする。式(37)の $q$ を1.0とする。

学習要素として、4層からなる神経回路を用いる。中間層を構成する第二層と第三層に、シグモイド関数( $y = \tan h(x)$ )で記述される入出力特性を持った非線形のニューロンを配置し、入力層と出力層は線形ニューロンとする。直接学習方式の神経回路は入力層より順に、5, 15, 15, 3個のニューロンによって構成され、間接ゲイン学習方式の神経回路は、3, 15, 15, 6個のニューロンによって構成される。

まず、オフライン学習の評価を行い、続いて、オンライン学習の評価を行う。

#### 5.1 オフライン学習

オフライン学習においては、直接ゲイン学習方式の入力空間 $(\theta, \Delta x)$ のうち、出力誤差 $\Delta x = (\Delta x, \Delta y)^T$ の学習領域を、それぞれの成分について、 $(-0.05, 0.05)$ とする。関節角ベクトル $\theta$ の学習領域は、関節の可動範囲全域とする。 $\theta$ および、 $\Delta x$ の学習領域を、各座標軸について7分割し、分割によって

生成される計  $8^5=32,768$  個の格子点について、式(10)の右辺を計算し、教師信号とする。式(10)の一般化逆行列  $\mathbf{J}^*(\theta)$  としては、疑似逆行列  $\mathbf{J}^+(\theta)$  を用いる。ヤコビ行列の計算には4回アームを動かす必要があり、関節角空間  $(\theta_1, \theta_2, \theta_3)$  の  $8^3=512$  点において、計2,048回のアームの動作を行い、ヤコビ行列を計算し、それぞれのヤコビ行列について、出力誤差空間  $(\Delta x, \Delta y)$  の  $8^2=64$  点の値から、教師信号を計算する。

間接ゲイン学習方式については、入力である関節角空間  $\theta = (\theta_1, \theta_2, \theta_3)^T$  の各座標軸をそれぞれ7分割し、分割によって生成される計  $8^3=512$  個の格子点について、ヤコビ行列を差分近似によって求め、これを教師信号として、学習要素の学習を行う。直接学習方式と同様、計2,048回のアームの動作によって、教師信号を計算した。このような教師信号をもとに、共役勾配法を用いて、オフライン学習によって、ヤコビ行列を神経回路によって学習した。

**Table 1** に、提案法と、数値解法の性能を示す。この数値は、前述の逆運動学問題を解く試行を2,500回行うことによって求めた。表中の  $P$  は生成した軌道の中で、追従動作中に、関節角の可動範囲を越えたりすることなく、実現可能であった軌道の割合である。RMSEは、追従誤差  $e$  のRMS誤差であり、MAXEは、 $|e|$  の最大値である。Tはワークステーション(シリコングラフィクス社製、IRIS-Crimson, R4000, 60[MHz])において、関節角の更新演算一回にかかる時間である。

直接学習方式と間接学習方式は、精度的には数値解法には及ばないものの、軌道の追従を実現できている。間接ゲイン学習方式は、直接学習方式よりも、必要な学習時間や追従精度の点で有利である。さらに線形適応系による補償の効果は大きく、精度が大きく改善されている。カルマンフィルタを利用した複合システムの精度は最も高い。ただし、計算にかかる時間も、最も大きい。 $\mathbf{P}(0)$  や  $q^2$  や  $r^2$  の値を1, 2桁変化させても、性能はあまり変わらない。

**Table 1** Results of offline learning

Methods	P[%]	RMSE[m]	MAXE[m]	T[ $\mu$ s]
DL	41.0	$5.00 \times 10^{-3}$	$3.16 \times 10^{-2}$	259
IGL	44.2	$1.26 \times 10^{-4}$	$4.07 \times 10^{-3}$	247
IGLL	44.2	$2.50 \times 10^{-5}$	$7.95 \times 10^{-4}$	275
IGLK	43.8	$1.30 \times 10^{-5}$	$5.90 \times 10^{-4}$	306
NM	43.9	$2.90 \times 10^{-5}$	$2.02 \times 10^{-4}$	104

Note:

P is the Percentage of successful trials.

RMSE is the Root Mean Square Error of the output.

MAXE is the MAXimum Error of output.

T is the CPU Time of one iteration.

**Table 2** Performance of linear adaptive systems

Methods	P[%]	RMSE[m]	MAXE[m]	T[ $\mu$ s]
LA	23.0	$2.00 \times 10^{-3}$	$8.62 \times 10^{-2}$	96
KF	20.8	$1.15 \times 10^{-3}$	$6.21 \times 10^{-2}$	123
LA'	28.0	$3.17 \times 10^{-3}$	$1.68 \times 10^{-1}$	96
KF'	41.0	$1.08 \times 10^{-3}$	$1.92 \times 10^{-1}$	123

参考のため、式(24)の  $\Phi(\theta(k))$  を  $\mathbf{0}$  とし、線形適応系のみで、逆運動学問題を解いた結果を **Table 2** に示す。図中のLAは式(26)に基づく線形適応系を示し、KFはカルマンフィルタに基づく線形適応系を示している。試行の開始時に、 $\Delta \mathbf{J}_a(0)$  と  $\mathbf{P}(0)$  の初期化を行っている。追従可能な軌道は少なく、間接ゲイン学習方式に比べて、追従精度も低い。さらに、LAとKFについて、一回の試行の終点から次の試行の始点へと移動する際にも、アーム先端位置を観測して、 $\Delta \mathbf{J}_a(k)$  と  $\mathbf{P}(k)$  の更新を継続した場合の結果を、それぞれ、LA' と KF' とする。この場合、カルマンフィルタ単独でも、RMS誤差1[mm]程度の精度を達成できるが、間接ゲイン学習方式や複合システムに比べて精度は低い。

**Fig. 4** に、半径0.11[m]の円軌道を目標軌道として、アーム先端を追従させる制御を行った時の先端の軌道とアームの動きを示す。これは、時刻  $k$  について、25おきに、アームを表示したものである。Fig. 4(a)において、アーム先端の近くにある数字が時刻を示している。破線の円は目標軌道を示し、実線の円は、アーム先端の軌道を示している。Fig. 4の(b)、(c)において、破線が見えないのは、実線と重なっているためである。線形適応系を利用した間接ゲイン学習方式やカルマンフィルタを利用した間接ゲイン学習方式のアームの動きは、肉眼では、数値解法方式のFig. 4(c)と区別がつかない。

参考のため、出力誤差のフィードバックを行わない場合のシミュレーション結果を **Table 3** に示す。これは、式(4)の  $K$  を0と置いたものである。直接学習方式では、先端位置制御を行うことは困難であるが、間接ゲイン学習方式ならば、RMS誤差6[mm]程度の制御は可能である。

## 5.2 オンライン学習

学習要素が未学習の状態から、逆運動学問題を解く試行を行いつつ、オンライン学習を行い、その性能の向上を調べるシミュレーションを行う。

オンライン学習においては、バックプロパゲーション法により学習を行った。  $i$  回目の学習時における。神経回路の出力を  $\Phi(\mathbf{q}_i)$ 、教師信号を  $\mathbf{p}_i$  とする。誤差評価関数を以下のように定義した。

$$S(i) = |\mathbf{p}_i - \Phi(\mathbf{q}_i)|^2 \quad (50)$$

神経回路の一つの結線の結合加重を  $w$ 、  $i$  回目の学習時における  $w$  の修正量を  $\Delta w(i)$  とおけば、

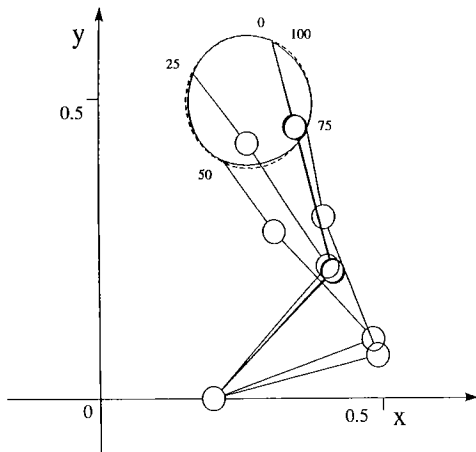
$$\Delta w(i) = -\eta \partial S(i) / \partial w + \xi \Delta w(i-1) \quad (51)$$

のように修正量を決定した。慣性率  $\xi$  は0.5とする。学習率  $\eta$  は、1.0, 0.1, 0.01, 0.001と変えてシミュレーションを行った中で、最も高速に性能が向上している値を選んだ。学習率  $\eta$  は、直接学習方式においては0.1とし、間接ゲイン学習方式に基づく手法においては0.01とした。

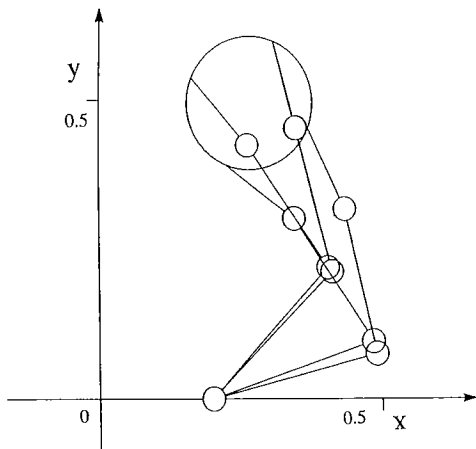
直接学習方式の学習式(14)の性能は良くなかったため、改良型の学習式(16)を利用して学習を行う。式(16)のパラメータ  $\beta$  の値は、0.05とする。間接ゲイン学習方式の学習には、式(20)を用いた。通常の場合、円軌道に追従する試行1回当たり、100回の学習が行われる。

学習に用いる  $|\Delta \theta|$  の上限を

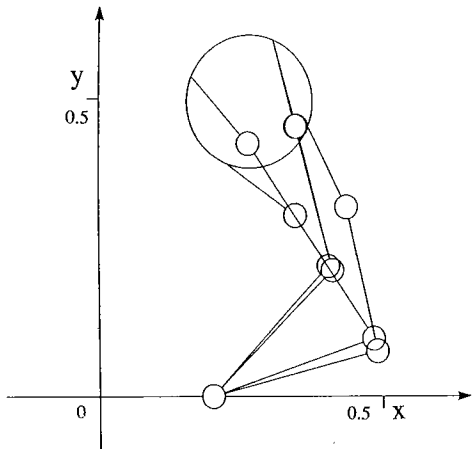
$$|\Delta \theta| < 0.5 [\text{rad}] = 28.6^\circ$$



(a) Direct Learning method



(b) Indirect Gain Learning method



(c) Numerical method

Fig. 4 Path of hand

とした。

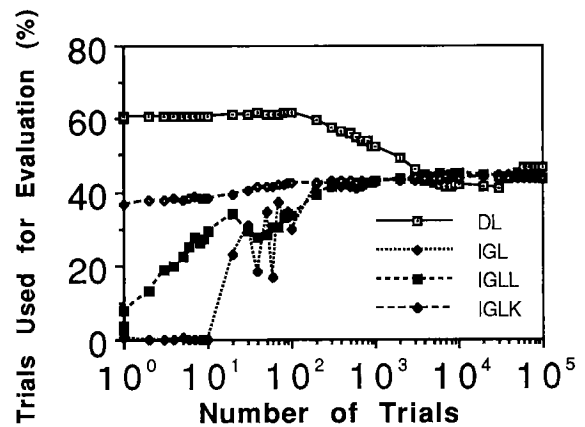
性能の評価においては、前節と同様に、学習を止めて、2,500回の試行を行い、追従誤差のRMS誤差を測定した。Fig. 5に評価に利用した軌道の割合とRMS誤差の変化を示す。Fig. 5に示すように、学習とともに性能が向上していくこ

Table 3 Results of offline learning (without output error feedback)

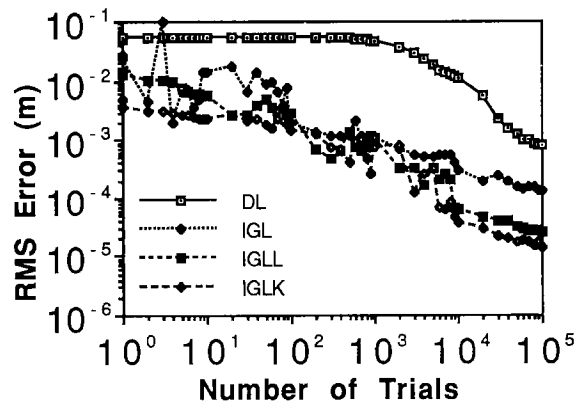
Methods	P[%]	RMSE[m]	MAXE[m]
DL	66.4	$5.20 \times 10^{-2}$	$2.95 \times 10^{-1}$
IGL	44.6	$5.73 \times 10^{-3}$	$1.32 \times 10^{-1}$
NM	44.6	$1.73 \times 10^{-3}$	$4.47 \times 10^{-2}$

Table 4 Results of online learning

Methods	P[%]	RMSE[m]	MAXE[m]
DL	46.7	$8.09 \times 10^{-4}$	$2.19 \times 10^{-2}$
IGL	44.1	$1.12 \times 10^{-4}$	$7.13 \times 10^{-4}$
IGLL	43.6	$2.50 \times 10^{-5}$	$9.07 \times 10^{-4}$
IGLK	43.4	$1.40 \times 10^{-5}$	$1.10 \times 10^{-3}$



(a) Percentage of trials used for evaluation



(b) RMS error

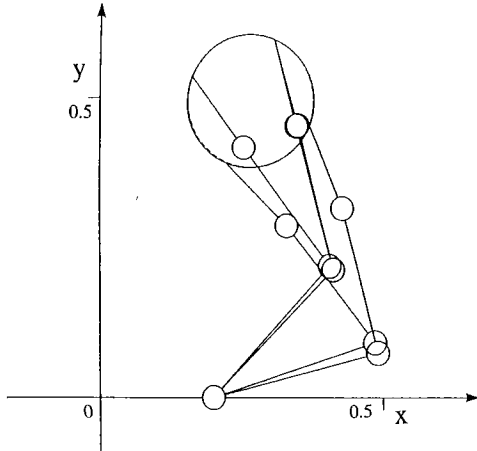
Fig. 5 Online learning

とが判明した。Table 4に、10万回のオンライン学習を行った時点での、解法システムの性能を示す。Table 5に、フィードバックを行わない場合のシステムの性能を示す。

オフライン学習時に比べて、直接学習方式の最終的な性能は、著しく向上しているが、RMS誤差が1[mm]以下になるためには、7万回の試行を必要とし、実用的ではない。間接ゲイン学習方式は、千回の試行で、RMS誤差が、1[mm]以下となっている。線形適応系との複合システムでは、200から300

**Table 5** Results of online learning  
(without output error feedback)

Methods	P[%]	RMSE[m]	MAXE[m]
DL	46.7	$2.00 \times 10^{-2}$	$1.62 \times 10^{-1}$
IGL	46.1	$2.84 \times 10^{-3}$	$1.51 \times 10^{-2}$



**Fig. 6** Path of hand (Direct Learning method)

回の試行で、RMS 誤差が、1[mm]以下になっている。

直接学習方式や単純な間接ゲイン学習方式では、完全に未学習の状態から、オンライン学習単独で学習を行うことは困難である。線形適応系を利用した間接ゲイン学習方式では、オンライン学習単独でも、場合によっては、有効であろう。オフライン学習で学習を行い、オンライン学習によって、モデル化誤差やシステムの特長変化に対応するという方策が、実用的である。

Fig. 6 に、直接学習方式について、Fig. 4 と同様の目標軌道と実際のアーム先端の軌道を示す。間接ゲイン学習方式並びに、線形適応システムやカルマンフィルタを利用した間接ゲイン学習方式のアームの動きは、肉眼では、Fig. 4 の(c) と区別がつかない。

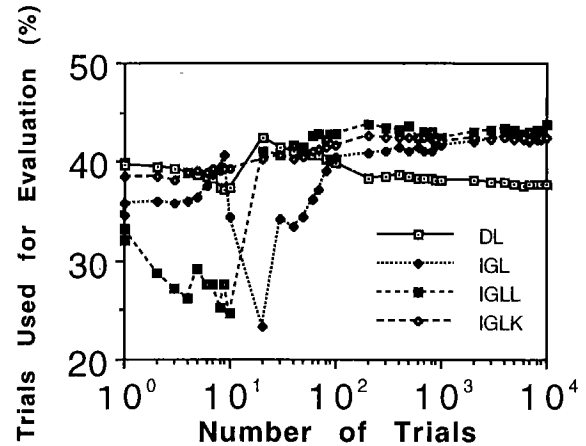
### 5.3 システムの特長変化への対応

オンライン学習によるシステムの特長変化への対応の効果を示すために、アームの特長を変化させ、円軌道に追従する試行を行いながら、非線形ゲインの学習を行うシミュレーションを行う。対象アームの特長は、以下のように変化させた。

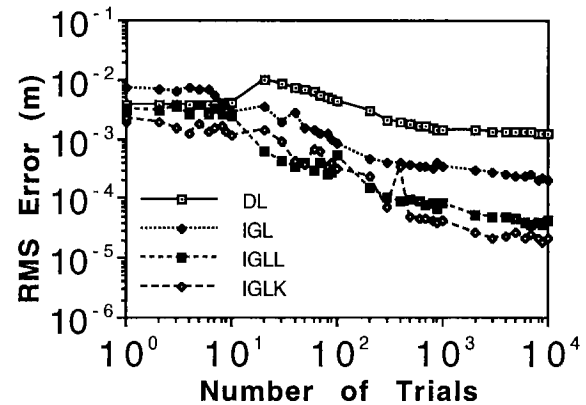
$$\begin{aligned}
 x &= x_0 + (L_1 + \Delta L_1) \cos(\theta_1 + \Delta\theta_1) \\
 &\quad + (L_2 + \Delta L_2) \cos(\theta_1 + \Delta\theta_1 + \theta_2 + \Delta\theta_2) \\
 &\quad + (L_3 + \Delta L_3) \cos(\theta_1 + \Delta\theta_1 + \theta_2 + \Delta\theta_2 + \theta_3 + \Delta\theta_3) \\
 y &= y_0 + (L_1 + \Delta L_1) \sin(\theta_1 + \Delta\theta_1) \\
 &\quad + (L_2 + \Delta L_2) \sin(\theta_1 + \Delta\theta_1 + \theta_2 + \Delta\theta_2) \\
 &\quad + (L_3 + \Delta L_3) \sin(\theta_1 + \Delta\theta_1 + \theta_2 + \Delta\theta_2 + \theta_3 + \Delta\theta_3) \\
 \Delta L_1 &= -0.15 \text{ [m]} \\
 \Delta L_2 &= 0.0 \text{ [m]} \\
 \Delta L_3 &= 0.15 \text{ [m]} \\
 \Delta\theta_1 = \Delta\theta_2 = \Delta\theta_3 &= 0.3 \text{ [rad]} = 17.2^\circ
 \end{aligned}$$

**Table 6** Simulation results after change of arm

Methods	P[%]	RMSE[m]	MAXE[m]
DL	39.9	$3.65 \times 10^{-3}$	$3.03 \times 10^{-2}$
IGL	34.6	$7.23 \times 10^{-3}$	$3.50 \times 10^{-1}$
IGLL	35.6	$6.57 \times 10^{-3}$	$2.95 \times 10^{-2}$
IGLK	37.8	$2.30 \times 10^{-3}$	$2.90 \times 10^{-2}$
NM	34.0	$7.40 \times 10^{-3}$	$3.25 \times 10^{-1}$



(a) Percentage of trials used for evaluation



(b) RMS error

**Fig. 7** Results of online learning after change of arm

これによって、座標変換回路の精度は悪化する。シミュレーションの結果を **Table 6** に示す。学習要素の初期状態は、4.2 節のオンライン学習の結果を利用した。各システムは、出力誤差のフィードバックを行っているため、平均的な精度の低下は極端ではないが、RMS 誤差が、1, 2 桁のオーダーで悪化している。

この状態から、前節に述べた手法で、アーム先端目標値と関節角ベクトルの初期値を乱数により生成し、逆運動学問題の求解問題を解く試行を繰り返す。逆運動学問題の計算過程において、学習要素のオンライン学習を行う。これによって、システムの性能がどのように向上するかを調べる。

**Fig. 7** に、試行の回数とシステムの性能との関係を示す。直



接学習方式では、一万回の試行後も、RMS 誤差は 1[mm]以上である。間接ゲイン学習方式はおよそ 100 回の試行によって、RMS 誤差が 1[mm]以下になっている。間接ゲイン学習方式において、追従可能な軌道の割合が、20 回付近で大きく低下している。これは、学習の初期段階において、学習係数  $\eta$  が大き過ぎて、性能の変化が振動的になったためと考えられる。前節でも、Fig. 5(a)に示すように、性能変化が振動的な部分があった。線形適応系を利用した間接ゲイン学習方式やカルマンフィルタを利用した間接ゲイン学習方式では、20 から 30 回の試行によって、1[mm]以下になっている。また、カルマンフィルタによる補償は、性能変化の安定化にも有効である。100 回程程度の試行で、対象の変化に対応可能であれば、十分実用的と考えられる。間接ゲイン学習方式に基づく手法のオンライン学習は、対象システムの変化への対応策として有効である。

以上のようなシミュレーションの結果から、直接学習方式については、精度及び学習性能において問題があるが、間接ゲイン学習方式は、有用であると結論できる。間接学習方式においては、線形適応系の利用によって精度は格段に向上しており、特にカルマンフィルタを用いた複合システムが、有効であると結論できる。

## 6. おわりに

出力誤差のフィードバック回路を学習によって獲得することによって、逆運動学問題を解くシステムとその学習法を提案し、シミュレーションによってその評価を行った。直接学習方式と間接ゲイン学習方式の二つの手法を提案した。さらに、間接ゲイン学習方式においては、線形適応系との組み合わせによる複合システムを提案した。シミュレーションの結果、間接ゲイン学習方式が有効であり、線形適応系と組み合わせることによって、一層高精度の解を得ることが可能になることが判明した。提案した手法は、学習機能により、対象システムの特性変化に対してロバストであるという特徴がある。

オンライン学習における学習時間は、学習係数によって、大きく変化する。今回のシミュレーションにおいては、学習時のオンラインでの係数最適化を行わなかった。学習時の学習係数の最適化手法の利用は今後の課題である。

提案した手法によって、フィードバック誤差学習を行うことにより、逆運動学モデルを獲得することが可能になる。逆運動学モデルとの複合システムについても発表する予定である。また、線形適応系の利用は、逆動力学問題の場合にも、有効であろうと考えられ、研究を進める予定である。

## 参考文献

[1] M. Kawato, K. Furukawa and R. Suzuki: "A Hierarchical neural-network model for control and learning of voluntary movement," Biol. Cybern., 57, pp. 169-185, 1987.  
 [2] M. I. Jordan: "Supervised learning and systems with excess degrees of freedom," COINS Technical Report, 88-27, pp. 1-41, 1988.  
 [3] M. Kuperstein: "Neural Model of Adaptive Hand-Eye Coordination for Single Postures," Science, 239, pp. 1308-1311, 1988.  
 [4] S. I. Colombano, Compton Michael and M. Bualat: "Goal Direct-

ed Model Inversion," Proc. of International Joint Conference on Neural Networks (IJCNN '91 Singapore), pp. 2422-2427, 1991.  
 [5] 阪口豊, 前田太郎, 座間雅夫, 大森隆司, 中野肇: "感覚系と運動系の協調システム(1)", 第 26 回計測自動制御学会大会予稿集, pp. 73-74, 1987.  
 [6] 橋本秀紀ほか: "視覚情報によるマニピュレータの位置・姿勢制御", 日本ロボット学会誌, vol. 8, no. 4, pp. 390-396, 1990.  
 [7] 田中健一, 清水優子, 土屋和雄: "神経回路網による冗長マニピュレータの逆キネマティクス問題の解法", 計測制御学会論文誌, vol. 27, no. 7, pp. 815-821, 1991.  
 [8] W. Fun and M. I. Jordan: "The moving basin; Effective action-search in adaptive control," Technical Report, #9204, Dept. of Brain & Cognitive Sciences, MIT, 1992.  
 [9] 斉藤, 北村: "ニューラルネットワークによる順キネマティクスの学習と逆キネマティクスへの応用", 計測自動制御学会第 13 回知能システムシンポジウム, pp. 29-34, 1991.  
 [10] E. Oyama and S. Tachi: "A Study of Human Hand Position Control Learning—Output Feedback Inverse Model—," Proc. of International Joint Conference on Neural Networks, (IJCNN '91 Singapore), pp. 1434-1443, 1991.  
 [11] E. Oyama and S. Tachi: "A Learning Method for Solving Inverse Problems of Static Systems," Proc. of International Joint Conference on Neural Networks (IJCNN '93 Nagoya), pp. 2843-2849, 1993.  
 [12] 大山英明, 館暉: "非線形ゲインを学習する逆問題解法システム", 電子情報通信学会論文誌, vol. J 77-D-II, no. 7, 1994.  
 [13] A. Liegeois: "Automatic Supervisory Control of the Configuration and Behavior of Multibody Mechanisms," IEEE Trans. SMC-7, 12, pp. 868-871, 1977.  
 [14] Y. Nakamura, H. Hanafusa and T. Yoshikawa: "Task-Priority Based Redundancy Control of Robot Manipulators," The International Journal of Robotics Research, 6, 2, pp. 3-15, 1987.  
 [15] J. S. Albus: "A New Approach to Manipulator control: the Cerebellar Model Articulation Controller (CMAC)," ASME, J. Dynamic Syst. Meas. Contr., 97, pp. 220-227, Sept. 1975.  
 [16] T. Kohonen: "Self-organized formation of topographically correct feature maps," Biol. Cybern., 43, 1982.  
 [17] D. E. Rumelhart, G. E. Hinton and R. J. Williams: "Learning International Representation by Error Propagation," Parallel Distributed Processing, D. E. Rumelhart and J. L. MacLelland and the PDP Research Group ed., pp. 318-326, MIT Press, 1986.  
 [18] C. G. Broyden: "A Class of Methods for Solving Nonlinear Simultaneous Equations," Math. Comp., vol. 19, pp. 577-593, 1965.  
 [19] C. A. Desoer and M. Vidyasagar: "Feedback Systems: Input Output Properties," Academic Press, 1975.

## Appendix. 1 出力フィードバック型逆モデル

出力フィードバック型逆モデルは、広義のニュートン法を学習要素によって、近似的に実現する。

連続なシステムの入力の微小変化  $\Delta\theta$  とそれに対応する微小変化  $\Delta x$  には、本文中の式(12)のような線形の関係が成立する。式(12)の逆モデルを学習要素によって獲得することを考える。学習要素  $\Phi(\theta, \Delta x)$  の教師信号  $\Phi'(\theta, \Delta x)$  として本文中の式(14)を利用する。式(12)の線形化モデル

$$\Delta x = J(\theta)\Delta\theta \tag{A 1}$$

を満たす  $\Delta\theta$  は  $\theta$  と  $\Delta x$  を利用して、

$$\Delta\theta^* = J^*(\theta)\Delta x + (I - J^*(\theta)J(\theta))u \tag{A 2}$$

と書くことができる。  $J^*(\theta)$  は  $J(\theta)$  の一般化逆行列であり、  $J(\theta)$  の疑似逆行列 (Moore-Penrose の一般化逆行列)  $J^{\#}(\theta)$  と、  $m$  行  $m$  列の行列  $G(\theta)$  を利用して、次式のように表される。

$$\mathbf{J}^*(\boldsymbol{\theta}) = \mathbf{J}^r(\boldsymbol{\theta}) + (\mathbf{I} - \mathbf{J}^r(\boldsymbol{\theta})\mathbf{J}(\boldsymbol{\theta}))\mathbf{G}(\boldsymbol{\theta}) \quad (\text{A } 3)$$

$\mathbf{G}(\boldsymbol{\theta})$  の値は、学習に用いた  $\Delta\boldsymbol{\theta}$  の分布によって決定される。 $\mathbf{u}$  は  $\Delta\mathbf{x}$  と無相関の  $m$  次元ベクトルである。ここでは、 $\Delta\mathbf{x}$  が微小であることを考慮し、 $\Delta\mathbf{x}$  の 2 次以上の項を無視している。

学習の結果として、

$$\begin{aligned} \boldsymbol{\Phi}(\boldsymbol{\theta}, \Delta\mathbf{x}) &\doteq E(\Delta\boldsymbol{\theta}^*) = \mathbf{J}^*(\boldsymbol{\theta})\Delta\mathbf{x} \\ &+ (\mathbf{I} - \mathbf{J}^r(\boldsymbol{\theta})\mathbf{J}(\boldsymbol{\theta}))E(\mathbf{u}) \end{aligned} \quad (\text{A } 4)$$

を得る。 $\boldsymbol{\Phi}(\boldsymbol{\theta}, \Delta\mathbf{x})$  は式 (A 1) の逆モデルを近似している。

オンライン学習においては、学習要素  $\boldsymbol{\Phi}(\boldsymbol{\theta}, \Delta\mathbf{x})$  に出力誤差

$$\Delta\mathbf{x}_d = \mathbf{x}_d - \mathbf{f}(\boldsymbol{\theta}) \quad (\text{A } 5)$$

を入力する。 $\Delta\mathbf{x}_d$  が十分小さい場合、 $\Delta\boldsymbol{\theta}$  とそれに対応する  $\Delta\mathbf{x}$  は、

$$\begin{aligned} \Delta\boldsymbol{\theta} &= \boldsymbol{\Phi}(\boldsymbol{\theta}, \Delta\mathbf{x}_d) - \boldsymbol{\Phi}(\boldsymbol{\theta}, \mathbf{0}) = \mathbf{H}(\boldsymbol{\theta})\Delta\mathbf{x}_d \\ \mathbf{H}(\boldsymbol{\theta}) &= \partial\boldsymbol{\Phi}(\boldsymbol{\theta}, \mathbf{x})/\partial\mathbf{x}|_{\mathbf{x}=\mathbf{0}} \end{aligned} \quad (\text{A } 6)$$

$$\Delta\mathbf{x} = \mathbf{J}(\boldsymbol{\theta})\mathbf{H}(\boldsymbol{\theta})\Delta\mathbf{x}_d \quad (\text{A } 7)$$

であり、 $\mathbf{J}(\boldsymbol{\theta})\mathbf{H}(\boldsymbol{\theta})$  が正則であるとする、

$$\begin{aligned} \boldsymbol{\Phi}(\boldsymbol{\theta}, \Delta\mathbf{x}) &\doteq E(\boldsymbol{\Phi}^r(\boldsymbol{\theta}, \Delta\mathbf{x})) \\ &= \mathbf{H}(\boldsymbol{\theta})(\mathbf{J}(\boldsymbol{\theta})\mathbf{H}(\boldsymbol{\theta}))^{-1}\Delta\mathbf{x} \end{aligned} \quad (\text{A } 8)$$

と学習される。 $\mathbf{H}(\boldsymbol{\theta})(\mathbf{J}(\boldsymbol{\theta})\mathbf{H}(\boldsymbol{\theta}))^{-1}$  は  $\mathbf{J}(\boldsymbol{\theta})$  の一般化逆行列の一つである。式 (A 4) と式 (A 8) の比較により、

$$\boldsymbol{\Phi}(\boldsymbol{\theta}, \Delta\mathbf{x}) \doteq \mathbf{J}^*(\boldsymbol{\theta})\Delta\mathbf{x} \quad (\text{A } 9)$$

を得る。

出力フィードバック型逆モデルは、学習要素の出力を利用して、システムへの入力の現在値  $\boldsymbol{\theta}(k)$  を次のように更新する。

$$\begin{aligned} \boldsymbol{\theta}(k+1) &= \boldsymbol{\theta}(k) + \boldsymbol{\Phi}(\boldsymbol{\theta}(k), \Delta\mathbf{x}_d(k)) - \boldsymbol{\Phi}(\boldsymbol{\theta}(k), \mathbf{0}) \\ &\doteq \boldsymbol{\theta}(k) + \mathbf{J}^*(\boldsymbol{\theta}(k))(\mathbf{x}_d - \mathbf{f}(\boldsymbol{\theta}(k))) \end{aligned} \quad (\text{A } 10)$$

この更新式は広義のニュートン法を近似的に実現している。フィードバック項  $\boldsymbol{\Phi}(\boldsymbol{\theta}, \Delta\mathbf{x}_d)$  を  $\Delta\mathbf{x}_d$  についてテイラー展開すると、

$$\boldsymbol{\Phi}(\boldsymbol{\theta}, \Delta\mathbf{x}_d) = \boldsymbol{\Phi}(\boldsymbol{\theta}, \mathbf{0}) + \mathbf{H}(\boldsymbol{\theta})\Delta\mathbf{x}_d + \mathbf{O}(\Delta\mathbf{x}_d^2) \quad (\text{A } 11)$$

を得る。 $\mathbf{O}(t^2)$  は、それぞれの方程式において  $t$  の二次以上の高次項の総和を表すものとする。出力誤差の変化は

$$\begin{aligned} \Delta\mathbf{x}_d(k+1) &= (\mathbf{I} - \mathbf{J}(\boldsymbol{\theta}(k))\mathbf{H}(\boldsymbol{\theta}(k)))\Delta\mathbf{x}_d(k) \\ &+ \mathbf{O}(\Delta\mathbf{x}_d(k)^2) \end{aligned} \quad (\text{A } 12)$$

である。このような反復改良システムは、 $\Delta\mathbf{x}_d(0)$  が十分小さい時、

$$0 < |\mathbf{I} - \mathbf{J}(\boldsymbol{\theta})\mathbf{H}(\boldsymbol{\theta})| < 1 \quad (\text{A } 13)$$

が成立すれば、

$$\Delta\mathbf{x}_d(k) \rightarrow \mathbf{0} \quad (k \rightarrow \infty) \quad (\text{A } 14)$$

となり、反復改良によって、高精度の解を得ることが可能である。式 (A 9) より、通常、式 (A 13) は成立する。

反復改良の初期値  $\boldsymbol{\theta}(0)$  が解から離れている場合、式 (A 10) の反復改良によって、 $\boldsymbol{\theta}(k)$  は、解の定義域の外に外れたり、局所最適解に収束してしまったり、振動してしまう場合がある。そのような場合は、ランダム探索により、初期値  $\boldsymbol{\theta}(0)$  を変更し、再び反復改良演算を行う過程を繰り返すことによって、十分な計算時間があれば、解に到達することが可能である。

## Appendix. 2 連続な目標値に対する入出力安定性

出力誤差を

$$\mathbf{e}(k) = \mathbf{x}_d(k) - \mathbf{x}(k) \quad (\text{A } 15)$$

とする。ここでは、

$$\Delta\mathbf{x}_d(k) = \mathbf{x}_d(k+1) - \mathbf{x}_d(k) \quad (\text{A } 16)$$

を入力、 $\mathbf{e}(k)$  を出力と見なした場合の入出力安定性を考える [19]。 $|\Delta\mathbf{x}_d(k)|$  の上限を考え、

$$|\Delta\mathbf{x}_d| = \sup_{k>0} (|\Delta\mathbf{x}_d(k)|) \quad (\text{A } 17)$$

とする。連続な目標値については、 $|\Delta\mathbf{x}_d|$  は十分小さいと仮定できる。

関節角の更新演算は、直接学習方式においては、

$$\begin{aligned} \Delta\boldsymbol{\theta}(k) &= \boldsymbol{\Phi}(\boldsymbol{\theta}(k), \Delta\mathbf{x}_d(k)) - \boldsymbol{\Phi}(\boldsymbol{\theta}(k), \mathbf{0}) \\ &= \mathbf{H}(\boldsymbol{\theta})\Delta\mathbf{x}_d(k) + \mathbf{O}(\Delta\mathbf{x}_d(k)^2) \end{aligned}$$

$$\Delta\mathbf{x}'_d(k) = \Delta\mathbf{x}_d(k) + \mathbf{K}\mathbf{e}(k) \quad (\text{A } 18)$$

であり、間接ゲイン学習方式においては、

$$\Delta\boldsymbol{\theta}(k) = \boldsymbol{\Phi}^r(\boldsymbol{\theta}(k))\Delta\mathbf{x}'_d(k) \quad (\text{A } 19)$$

である。ヤコビ行列の一般化逆行列  $\mathbf{J}^*(\boldsymbol{\theta})$  の近似行列  $\mathbf{J}_d^*(\boldsymbol{\theta}(k))$  を用いて、直接学習方式の更新演算は

$$\Delta\boldsymbol{\theta}(k) = \mathbf{J}_d^*(\boldsymbol{\theta}(k))\Delta\mathbf{x}'_d(k) + \mathbf{O}(\Delta\mathbf{x}'_d(k)^2) \quad (\text{A } 20)$$

間接ゲイン学習方式の更新演算は

$$\Delta\boldsymbol{\theta}(k) = \mathbf{J}_d^*(\boldsymbol{\theta}(k))\Delta\mathbf{x}'_d(k) \quad (\text{A } 21)$$

と表すことが可能である。いずれの場合も、先端位置は次式のように更新され、

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{x}(k) \\ &+ \mathbf{J}(\boldsymbol{\theta}(k))\mathbf{J}_d^*(\boldsymbol{\theta}(k))\Delta\mathbf{x}'_d(k) \\ &+ \mathbf{O}(\Delta\mathbf{x}'_d(k)^2) \end{aligned} \quad (\text{A } 22)$$

出力誤差は次式のように更新される。

$$\begin{aligned} \mathbf{e}(k+1) &= (\mathbf{I} - \mathbf{J}(\boldsymbol{\theta}(k))\mathbf{J}_d^*(\boldsymbol{\theta}(k))\mathbf{K})\mathbf{e}(k) \\ &+ (\mathbf{I} - \mathbf{J}(\boldsymbol{\theta}(k))\mathbf{J}_d^*(\boldsymbol{\theta}(k)))\Delta\mathbf{x}_d(k) \\ &+ \mathbf{O}(\Delta\mathbf{x}'_d(k)^2) \end{aligned} \quad (\text{A } 23)$$

式 (A 21) の  $\mathbf{O}(\Delta\mathbf{x}'_d(k)^2)$  を  $\mathbf{O}^*(\Delta\mathbf{x}'_d(k)^2)$  と置く。十分小さい正の数  $d$  について、

$$|\Delta\mathbf{x}'_d(k)| < d \quad (\text{A } 24)$$

と仮定すると、

$$c = \sup_{k>0} (|\mathbf{O}^*(\Delta\mathbf{x}'_d(k)^2)|/|\Delta\mathbf{x}'_d(k)|) \quad (\text{A } 25)$$

は十分小さくなる。 $d$  を小さくすることによって、 $c$  はいくらでも小さくなる。

$$a = \sup_{k>0} (|\mathbf{I} - \mathbf{J}(\boldsymbol{\theta}(k))\mathbf{J}_d^*(\boldsymbol{\theta}(k))|) + c$$

$$b = \sup_{k>0} (|\mathbf{I} - \mathbf{J}(\boldsymbol{\theta}(k))\mathbf{J}_d^*(\boldsymbol{\theta}(k))\mathbf{K}|) + c|\mathbf{K}|$$

$$(\text{A } 26)$$

と置く、出力誤差のノルムについて

$$\begin{aligned} |\mathbf{e}(k+1)| &\leq b|\mathbf{e}(k)| + a|\Delta\mathbf{x}_d(k)| \\ &\leq b^k|\mathbf{e}(0)| + a(|\Delta\mathbf{x}_d(k)| + b|\Delta\mathbf{x}_d(k-1)| \\ &\quad + \dots + b^k|\Delta\mathbf{x}_d(0)|) \\ &\leq b^k|\mathbf{e}(0)| \\ &\quad + a(1 + b + b^2 + \dots + b^k)|\Delta\mathbf{x}_d| \end{aligned} \quad (\text{A } 27)$$

という不等式が成立し、

$$b < 1 \quad (\text{A } 28)$$

であれば、1 以上の任意の  $k$  について、

$$|\mathbf{e}(k)| \leq b|\mathbf{e}(0)| + a/(1-b)|\Delta\mathbf{x}_d| \quad (\text{A } 29)$$

が成立する。式(A 28)は、 $J_a^*(\theta)$ が $J^*(\theta)$ に十分近く、 $K$ が適切であり(例えば、 $K=I$ )、 $c$ が十分小さければ、成立する。 $|\Delta x_d|$ の上界について考えると、

$$|\Delta x_d| \leq (1+a|K|/(1-b))|\Delta x_d| + b|K||e(0)| \quad (A 30)$$

であり、

$$(1+a|K|/(1-b))|\Delta x_d| + b|K||e(0)| < d \quad (A 31)$$

であれば、式(A 24)の成立が保証される。式(A 31)は、 $|\Delta x_d|$ と初期誤差 $|e(0)|$ が十分小さければ成立する。

以上のように、 $|e(0)|$ 、 $|\Delta x_d|$ が十分小さく、 $J_a^*(\theta)$ が $J^*(\theta)$ に十分近ければ、式(A 24)、式(A 28)、式(A 31)を成立させる $d$ が存在し、式(A 29)の成立より、明らかに提案法の入出力安定性が成立する。



大山英明 (Eimei Oyama)

1962年10月23日生。1987年3月東京大学大学院工学系研究科修士過程修了。同年4月通産省工業技術院機械技術研究所入所。ロボット工学部バイオロボティクス研究室所属。テレグジスタンス、人間の運動制御系のモデル化、神経回路、非線形最適化等の研究に従事。電子情報通信学会、計測自動制御学会等の会員。(日本ロボット学会正会員)



館 暲 (Susumu Tachi)

1946年1月1日生。1973年東京大学大学院工学系研究科博士課程修了。同年同大学助手。1975年機械技術研究所入所。遠隔制御課長、バイオロボティクス課長を歴任。1979~1980年MIT客員研究員。1989年東京大学助教授。1992年同教授。国勢制御連合学会(IMEKO)TC17(ロボティクス)議長。SICEフェロー。バースペクトルを用いる信号処理、盲導犬ロボット、テレグジスタンスなどの研究に従事。工学博士。計測自動制御学会などの会員。(日本ロボット学会正会員)