

拡張フィードバック系による 逆運動学モデルのオフライン学習

大山 英明*¹ 舘 暲*²

Off-line Inverse-kinematics Model Learning by an Extended Feedback System

Eimei Oyama*¹ and Susumu Tachi*²

Recently the use of neural networks as the inverse-kinematics model of a robot arm has been proposed in learning control of the robot arm. The forward and inverse modeling, the feedback error learning schema and the goal directed model inversion were proposed to extend the acquisition of the inverse model for the systems with many-to-one input-output correspondence. However, these methods can be used only for on-line learning. The learning of neural networks usually requires many iterations of robot arm movement and of its position measurement. In order to reduce the number of movements of the robot arm, the hybrid system which consists of a learning element and an extended feedback controller are proposed. The learning element approximates the inverse kinematics model of the robot arm. By using the extended feedback controller, the high precision solutions of the inverse-kinematics problems are obtained so that these solutions can be used for the teaching signal of the learning element. After the acquisition of these solutions, the off-line learning of the learning element is conducted. The use of forward model of the robot arm is also proposed. The numerical simulations show the good performance of the proposed system.

Key Words: Inverse-kinematics Model, Learning Control, Neural Networks, Extended Feedback 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、ロボットアームの制御において重要課題の一つとして古くから研究されている。その新しい手法として神経回路を利用した学習制御の研究が行われており、特性が未知のロボットアームの逆運動学モデルや逆動力学モデルを獲得する研究が行われている[1]~[3]。逆モデルの獲得法として、直接逆モデリングが提案され、Kuperstein[4]、阪口[5]、橋本[6]等によって、人間の手先位置制御の研究やロボットアームの先端位置制御に応用された。その後、Jordanらによる順逆モデリング[2]や川人らによるフィードバック誤差学習則[1]、Colombanoらによる目標指向モデル反転 (Goal Directed Model Inversion) [3]等の逆モデルの学習法が提案されている。田中らはフィードバック誤差学習則を拡張し、冗長アームの逆運動学モデルの学習を行っている[7]。

順逆モデリングやフィードバック誤差学習則、目標指向モデル反転、田中らの手法では、学習が局所的な情報に基づいて行

われるため、対象システムによっては、逆モデルの獲得が難しい場合があるといった問題点があるが[8]~[10]、ロボットアームについては、関節角のストロークエンド付近を除けば、逆運動学モデルを学習できる。しかしながら、これらの手法は基本的にオンライン学習のための学習法であり、対象とするロボットを実際に動かし、その先端位置を観測する過程を繰り返す必要がある。実際には、ロボットを動かす回数は少ない方が望ましく、オンライン学習は不適当な場合も多い。直接逆モデリングはオフライン学習のための手法であるが、対象が多対一対応システムの場合には、正確な逆モデルを獲得できないことが指摘されている[1][2]。直接逆モデリングでは、冗長アームの逆運動学モデルを獲得することは困難である。新たなオフライン学習法が必要である。

対象システムの順モデルを学習によって獲得し、対象システムの代わりに順モデルを用いることによって、オフライン学習が可能となる。しかしながら、順モデルにはモデル化誤差が存在し、順モデルを利用して学習した逆モデルにも、モデル化誤差が生じる。そのため、逆モデル学習の最終段階では、順モデルを利用するよりも、ロボットアームを動作させて、教師信号を得ることが望ましい。

筆者らは、静的なシステムの逆問題を解く手法として、フィードバック誤差学習則を拡張し、逆モデルと拡張したフィードバック系とを組み合わせた複合逆問題解決システムを提案して

原稿受付 1994年5月26日

*¹工業技術院機械技術研究所

*²東京大学工学部

*¹Agency of Industrial Science and Technology, Mechanical Engineering Laboratory

*²Faculty of Engineering, University of Tokyo

いる[9][10]. 本論では, この手法を進展させて, 拡張フィードバック系を用いた逆運動学モデルのオフライン学習法を提案し, また, 逆運動学モデルと出力誤差のフィードバック系とを組み合わせた冗長アームの先端位置制御系を提案する. さらに, 数値シミュレーションによって, 提案法の有効性を確認する.

2. 拡張フィードバック系による 逆モデルのオフライン学習

2.1 問題の定式化

m 次元関節角ベクトルを θ , n 次元アーム先端位置ベクトルを x として, 入出力関係が

$$x = f(\theta) \quad (1)$$

となるロボットアームがあるものとする. このロボットアームに対して, 関節角ベクトル θ を自由に与えることができ, その時のアーム先端位置 x を観測できるものとする. アーム先端位置 x の目標値 x_d が与えられた時,

$$x_d = f(\theta) \quad (2)$$

を実現する関節角ベクトル θ の値を計算する問題は逆運動学問題と呼ばれている. 対象ロボットアームに対して, 次式のような関係を満たすシステム $g(x)$ を逆運動学モデルと呼ぶ.

$$x = f(g(x)) \quad (3)$$

対象システムの特性が未知の時, 対象システムの入出力を観測して, 逆運動学問題の解を出力する逆運動学モデル $g(x)$ を学習要素 $\Phi(x)$ によって獲得する問題を考える. ここで, 学習要素 $\Phi(x)$ は任意の連続関数を近似できるものとし, 入力 t である時, 教師信号 s , が与えられた場合, 最終的に,

$$\Phi(t) = E(s) \quad (4)$$

となるように, 学習が行われるものとする. $E(s)$ は s の平均値を示す. また補間能力を持つものとする. 現在広く応用されているバックプロパゲーション学習を行う多層神経回路[11]は, この性質を近似的に満たしている.

本論において, オンライン学習とは, 対象システムの出力目標値を逆モデルに入力し, 逆モデルの出力を対象システムに入力して, 対象システムを動作させ, 対象システムの出力を観測して, 教師信号を計算する学習方式を指す. 一方, オフライン学習とは, 出力目標と無関係に対象システムを動作させ, 一括して, 逆モデルの教師信号を作成し, その教師データを基に学習を行う学習方式を指す.

逆モデルの学習法としては, 直接逆モデリング, 順逆モデリング, フィードバック誤差学習則, 目標指向モデル反転といった手法が提案されている. 1章で述べたように, 逆問題の獲得法には問題があり, 新しい逆モデルの学習法が必要である.

2.2 複合逆問題解法システム

フィードバック誤差学習[1]においては, フィードバック系の出力した逆モデルの修正方向に, 解が存在する保証が無いため, 対象システムの種類によっては, 逆モデルを獲得することが困難な場合がある[8]~[10]. 例えば, ロボットアームの逆運動学モデルの学習において, 単純な順逆モデリングやフィードバック誤差学習則では, 関節角のストロークエンド付近において, 教師信号が, 関節角の可動範囲外の値をとる場合があり, そのような場合には, 正確な逆モデルを得られない. この

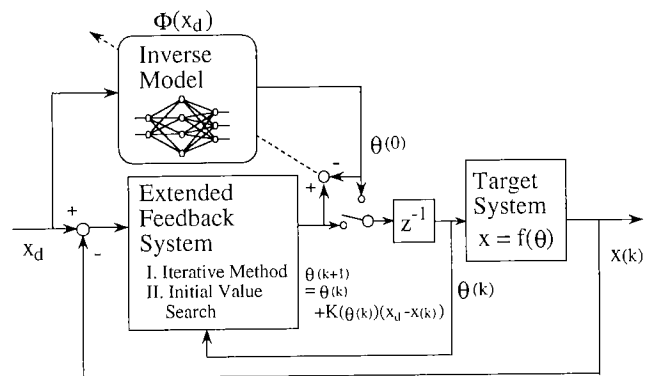


Fig.1 Hybrid system that consists of Inverse model and extended feedback system

問題を解決するために, 筆者らが提案した逆モデルと拡張されたフィードバック系から構成される複合逆問題解法システム[9][10]を利用できる.

Fig. 1 に逆モデルと拡張フィードバック系より構成される複合逆問題解法システムの概念図を示す. 図中のスイッチは, 逆モデルまたは拡張フィードバック系を対象システムに接続する. 逆モデルは逆問題の解の初期値を生成し, 拡張フィードバック系は, ニュートン法に基づく反復改良演算によって, 逆問題の解を求める. 従来のシステムと大きく異なる点は, 解が局所最適解に陥ってしまったり, 或いは定義域の外に外れてしまった場合に, 初期値を変更して, 反復改良演算をやり直す機構を, 拡張フィードバック系が備えているという点である. これにより, 連続な対象システムに対して, 正確な解を得ることが可能である.

複合逆問題解法システムの基本動作は以下の通りである.

- (I) 出力目標値 x_d が与えられた時, 先ず, 逆運動学モデルの出力 $\Phi(x_d)$ を対象システムの入力の初期値 $\theta(0)$ とする. すなわち,

$$\theta(0) = \Phi(x_d) \quad (5)$$

とする.

- (II) 続いて, 拡張フィードバック系を対象システムに接続し, 次のような反復改良を行う.

$$\theta(k+1) = \theta(k) + K(\theta(k))(x_d - f(\theta(k))) \quad (6)$$

ここで, $K(\theta)$ は適切な非線形ゲインであり, 本稿では, ロボットアームのヤコビ行列 $J(\theta(k))$ の近似行列 $J_a(\theta(k))$ を利用して,

$$K(\theta(k)) = \alpha(k) J_a^+(\theta(k)) \\ = \alpha(k) J_a^T(\theta(k)) (J_a(\theta(k)) J_a^T(\theta(k)))^{-1} \quad (7)$$

とする. $J_a^+(\theta)$ は $J_a(\theta)$ の疑似逆行列 (Moore-Penrose の一般化逆行列) であり, $\alpha(k)$ は直線探索によって決定される縮小因子である. この反復改良演算は, 近似的に広義のニュートン法[12]を実現している. ヤコビ行列の計算法については, 2.4 節で述べる.

- (III) 反復演算によって, 解を得られない場合には, 固定した初期値の系列から新たな初期値 $\theta(0)$ を定め, 過程 (b) に戻って, 再び式 (6) の反復演算を行う. この過程を解が得られるか, または一定の時間を超過するまで繰り返す.

i 番目の初期値変更によって生成される初期値を $\theta(0)^{(i)}$ とする。過程(III)で、固定した初期値の系列を用いることによって、 $\theta(0)^{(i)} (i=1, 2, 3, \dots)$ が、逆モデルへの全ての入力に対して、共通となるようにし、逆モデルへの入力に対する教師信号の一意性を保証する[9][10]。これによって、過程(II)によって、逆問題の解が得られた場合には、その解を教師信号として、逆モデルの学習を行うことが可能である。

2.3 逆モデルのオフライン学習法

2.3.1 逆モデルのオフライン学習法の提案

本論では、拡張フィードバック系による逆モデルのオフライン学習法として、次のような手順を提案する。

- (I) 目標値空間から十分な数の標本点を選び、それぞれの標本点について、逆モデルの初期値を反復の初期値とし、拡張フィードバック系を利用して、目標値を実現する関節角ベクトルを求め、教師データとして記録する。
- (II) 得られた教師データについて、学習要素の学習をオフラインで行う。
- (III) (I)(II)の過程を繰り返す。

逆モデルが未学習の状態では、逆モデルの初期値を、拡張フィードバック系の反復の初期値として利用しない方が良い場合がある。その場合、過程(I)は次のように変更される。

- (I)' 目標値空間から十分な数の標本点を選び、それぞれの標本点について、適当な初期値から、拡張フィードバック系を利用して、目標値を実現する関節角ベクトルを求め、教師データとして記録する。

逆モデルの学習が進行すると、過程(I)'の代わりに、(I)によって、教師信号の計算を行う。また、直接逆モデリングによってオフライン学習を行い、逆モデルの初期値として、その学習結果を利用することも有効である。

学習要素が十分な近似能力を持っている場合には、過程(I)または(I)'によって得られた教師信号に従って、過程(II)の学習要素の学習を行えば、逆モデルの学習が可能である。しかしながら、過程(I)や(I)'で得られた教師信号は、学習要素にとって必ずしも近似しやすいものであるとは限らない。初期値の変更が行われた場合には、教師信号に不連続領域が生じる。現在用いられている学習要素、例えば、多層神経回路の場合、不連続領域の近似は難しく、その近傍において、近似精度が低下する。シミュレーションにおける経験的な事例であるが、過程(II)のオフライン学習が進行した時点で、再び、過程(I)により、教師データの更新を行った方が、近似しやすい教師データを得ることができる場合が多い。未学習の逆モデルの学習を行う場合、逆モデルが不正確な学習の初期段階では、教師データ更新の効果は大きい。現時点では、数値シミュレーションの経験から、教師データの更新スケジュールをアドホックに決定している。更新スケジュールの最適化は、今後の課題である。

入力次元が小さい場合、入力空間の分割によって生成された格子点を代表点とすれば良い。入力次元が増加すれば、ランダムサンプリングによって代表点を選ぶことが有効となるであろう。最初に少ない代表点において、オフライン学習を行い、その後、代表点の数を増やしていく手法によって、より高速な最適化を行うことができる。

2.3.2 アームの動作回数

アーム先端の目標点の個数を N_{xd} 、解を求めるために必要な式(6)の反復演算の回数の平均値を N_I 、式(6)の更新演算を行うために必要な対象システムの動作回数の平均値を N_U とすると、過程(I)の教師信号の計算を行うために必要な対象システムの動作回数の総和 N_T は $N_{xd} N_I N_U$ となる。過程(III)の繰り返し回数を N_R とすると、対象システムの動作回数の総計 N_T は $N_{xd} N_I N_U N_R$ となる。

逆モデルが正確なほど、 N_I は小さい。 N_U の値は拡張フィードバック系の構成によって異なる。 N_R の値は、逆モデルの初期状態や対象システムによって異なる。正確な逆運動学モデルが学習されている場合には、1か2で十分であるが、未学習の場合には、3以上が望ましい。

順逆モデリングやフィードバック誤差学習の場合、学習回数を N_L とすると、学習に必要な対象システムの動作回数 N_T は、 $N_{xd} N_L$ となる。共役勾配法[13]等の非線形最適化の手法を用いた場合、非線形最適化の反復回数を N_{Io} とし、各反復における直線探索のための動作回数の平均値を N_{Ls} とすれば、動作回数 N_T は $N_{xd} N_{Io} N_{Ls}$ となる。通常、 N_L や $N_{Io} N_{Ls}$ は $N_I N_U N_R$ に比べて、十分大きく、提案する手法は、従来法に比べて、実際に対象システムを動作させる回数は少ない。

2.4 順モデルの利用

関節角空間から適切なサンプル点を選び、アーム先端を観測して、アーム先端位置を記録し、関節角ベクトルとそれに対応するアーム先端位置とを教師データとし、神経回路等の学習要素の学習を行うことによって、アームの順モデルを獲得できる。

対象システムを動作させる代わりに、順モデルを利用することにより、より高速に、逆運動学モデルの獲得が可能である。順モデルを利用して、比較的正確な逆運動学モデルを獲得した後、実際にアームを動かして、教師信号を得て、逆運動学モデルの高精度化を図ることによって、高速な学習が可能となる。

順モデルを利用する場合、2.3節で述べた対象システムの動作回数は、そのまま、順モデルの評価回数となる。神経回路等の非線形学習要素によって順モデルを作成した場合、順モデルの出力の計算は、他の四則演算に比べて、時間がかかる場合が多い。提案法は順モデルの評価回数が少ないため、高速に計算が行えるものと期待できる。ただし、計算時間の優位性は、実際にアームを動かす場合に比べて小さい。従来法による学習で、精度の高い逆モデルが得られる可能性もある。そのような場合には、順モデルを利用して、従来法を用いる方が良い。また、順モデルがあれば、GA等の大域的な最適化手法を用いることもできる。順モデルを用いて学習を行った場合、最終段階で、提案法を利用することにより、順モデルのモデル化誤差に起因する逆モデルの誤差を減少させることが可能である。

2.5 ヤコビ行列の計算法

拡張フィードバック系において、式(6)の反復演算を行うためには、ヤコビ行列の近似行列 $J_a(\theta)$ を計算する必要がある。本稿では、二種類の手法を用意する。修正ニュートン法のように、対象システムを動作させ、数値微分によりヤコビ行列を推定する手法と、ヤコビ行列を学習する手法の二つである。

対象システムを動作させ、数値微分によってヤコビ行列を推定する場合、ヤコビ行列の推定値 $J_a(\theta)$ の i 行 j 列成分 $J_{aj}(\theta)$ は、 d を十分小さな数として、次のように計算できる。

$$J_{aj}(\theta) = \frac{f_i(\theta + de^{(j)}) - f_i(\theta)}{d} \quad (8)$$

ここで、 $e^{(j)}$ は、その成分 $e_i^{(j)}$ がクロネッカーのデルタ δ_{ij} を利用して、

$$e_i^{(j)} = \delta_{ij} \quad (9)$$

のように定義される m 次元基底ベクトルである。この手法でヤコビ行列の推定を行う場合、2.3.2 項で述べた $J_a^*(\theta(k))$ の計算に必要な対象システムの動作回数は m である。

ヤコビ行列 $J(\theta)$ を学習する手法としては、筆者らが提案している手法を用いる [14]~[17]。適当なサンプリング点を選び、式(8)によって、ヤコビ行列を推定し、その値を、関節角ベクトルを入力とし、 n 行 m 列に対応する出力を持つ学習要素の教師信号とすれば良い。ヤコビ行列のオンライン学習も可能である。ヤコビ行列を学習した場合、疑似逆行列の計算に必要な対象システムの動作回数は 0 になる。

本稿のシステムでは、関節角ベクトルの単位はラジアンである。 d は 0.05 とした。

3. 逆モデル・フィードバック系複合系によるアーム制御

本来の複合解法システムは、対象システムの出力目標値が不連続に与えられる場合を想定している。そのままでも、出力目標値が連続的に与えられる場合にも対応可能であるが、効率は良くない。本章では、連続的にアーム先端の目標位置が与えられる場合のアーム制御系について述べる。

拡張フィードバック系の基本動作を示す式(6)を利用して、逆モデルの出力を利用することなく、次式のようなアーム先端位置のフィードバック制御系を構成できる [17]。

$$\begin{aligned} \theta(t+1) &= \theta(t) + \Delta\theta(t) \\ \Delta\theta(t) &= J_a^*(\theta(t))(x_d(t+1) - f(\theta(t))) \end{aligned} \quad (10)$$

しかしながら、上式は関節角の可動範囲を考慮していないため、適切な関節角軌道の設計を行えば実現可能な目標軌道であっても、 $\theta(t)$ が関節角のストロークエンドを越えてしまい、追従できない場合がある。

冗長自由度のあるアームに関しては、冗長自由度を有効に利用することによって、追従可能な軌道を増加させることが、可能である。微分可能なスカラーの評価関数 $V(\theta)$ が与えられた時、適当な正の係数 β を用いて、

$$\begin{aligned} \Delta\theta(t) &= J_a^*(\theta(t))(x_d(t+1) - f(\theta(t))) \\ &+ (I - J_a^*(\theta(t))J_a(\theta(t)))\beta \frac{\partial V(\theta)}{\partial \theta} \end{aligned} \quad (11)$$

とすることによって、局所的に、スカラー関数 $V(\theta)$ が増大する方向に $\Delta\theta(t)$ を定めることが可能である [18] [19]。特に、

$$\begin{aligned} \Delta\theta(t) &= J^*(\theta(t))(x_d(t+1) - f(\theta(t))) \\ &+ (I - J^*(\theta(t))J(\theta(t)))(\theta_0 - \theta(t)) \end{aligned} \quad (12)$$

とすれば、この $\Delta\theta(t)$ は、対象システム

$$x_d(t+1) = f(\theta(t) + \Delta\theta(t)) \quad (13)$$

の線形化モデル

$$x_d(t+1) = f(\theta(t)) + J(\theta(t))\Delta\theta(t) \quad (14)$$

を満たし、かつ基準となる関節角ベクトル θ_0 からのノルムの二乗和

$$U = 1/2 \|\theta(t+1) - \theta_0\|^2 \quad (15)$$

を最小化する。

関節角 θ_i の可動範囲を $[\theta_{\min i}, \theta_{\max i}]$ とする。 θ_0 として、各関節角の可動範囲の中心点

$$\theta_{mi} = \frac{(\theta_{\min i} + \theta_{\max i})}{2} \quad (16)$$

によって構成されるベクトル θ_m を利用し、適当な正の係数 β を利用して、

$$\begin{aligned} \Delta\theta(t) &= J_a^*(\theta(t))(x_d(t+1) - f(\theta(t))) \\ &+ (I - J_a^*(\theta(t))J_a(\theta(t)))\beta(\theta_m - \theta(t)) \end{aligned} \quad (17)$$

という制御則を得る。 θ_m に近づく方向に、関節角ベクトル $\theta(t)$ の更新が行われ、結果として、 $\theta(t+1)$ は、ストロークエンドから遠ざかる。しかしながら、式(17)は局所的な情報しか用いていないため、過去の経過によっては、ストロークエンドを越えてしまう場合がある。

逆運動学モデルが正確に学習されていれば、

$$\theta(t) = \Phi(x_d(t)) \quad (18)$$

によって生成される $\theta(t)$ はストロークエンドを越えることは無く、目標軌道が実現可能な軌道であれば実現できる。ただし、学習によって獲得された逆運動学モデルには、通常、モデル化誤差が存在し、高精度の逆運動学解は得られない場合が多い。 θ_0 として、逆運動学モデルの出力 $\Phi(x_d)$ を利用することによって、フィードバック系によってアーム先端目標位置の実現を計りつつ、逆運動学モデルを補助的に用いて、関節角を逆運動学モデルの出力に近づける、次のようなアーム先端位置制御系を得ることができる。

$$\begin{aligned} \Delta\theta(t) &= J_a^*(\theta(t))(x_d(t+1) - f(\theta(t))) \\ &+ (I - J_a^*(\theta(t))J_a(\theta(t)))(\Phi(x_d(t+1)) - \theta(t)) \end{aligned} \quad (19)$$

一方、主として逆運動学モデルによってアーム先端目標位置の実現を計り、フィードバック系によって逆運動学モデルの出力を修正する、次式のような制御系も構成できる。

$$\begin{aligned} \theta(t+1) &= \Phi(x_d(t+1)) + \Delta\theta(t+1) \\ \Delta\theta(t+1) &= \Delta\theta(t) + J_a^*(\theta(t))(x_d(t) - f(\theta(t))) \\ &+ (I - J_a^*(\theta(t))J_a(\theta(t)))(\Phi(x_d(t+1)) - \theta(t)) \end{aligned} \quad (20)$$

一般の逆運動学モデルは、しばしば不連続関数になる。逆運動学モデルの不連続点では、本章で提案した線形近似に基づく手法は必ずしも有効で無くなり、効率は落ちるが、拡張フィードバック系の本来の動作に基づく制御を行うことが望ましい。

4. シミュレーション

提案する手法の有効性を示すために、Fig. 2 に示すような 2 次元平面上を動く 3 自由度アームについて、逆運動学モデルを獲得し、アーム先端位置制御を行うシミュレーションを行う。関節角ベクトル $\theta = (\theta_1, \theta_2, \theta_3)^T$ とそれに対応するアーム先端位置ベクトル $x = (x, y)$ の関係は

$$\begin{aligned} x &= x_0 + L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2) + L_3 \cos(\theta_1 + \theta_2 + \theta_3) \\ y &= y_0 + L_1 \sin(\theta_1) + L_2 \sin(\theta_1 + \theta_2) + L_3 \sin(\theta_1 + \theta_2 + \theta_3) \end{aligned} \quad (21)$$

である。 θ_1 の可動範囲は $[-30^\circ, 120^\circ]$ 、 θ_2 の可動範囲は $[0^\circ, 120^\circ]$ 、 θ_3 の可動範囲は $[-60^\circ, 60^\circ]$ とする。アームの寸法は $L_1=0.30$ [m]、 $L_2=0.25$ [m]、 $L_3=0.15$ [m]とする。

シミュレーションで用いる拡張フィードバック系の具体的な構成は Appendix. 1 に示す。以下、提案する手法において、拡張フィードバック系の近似ヤコビ行列に修正ニュートン法の数値微分を用いたものを ND (Numerical Differentiation method)、学習要素によってヤコビ行列を学習したものを GL (Gain Learning method) と略す。また、従来の手法の代表として順逆モデリングを選び、順逆モデリングを利用した逆モデル獲得法を FIM (Forward and Inverse Modeling) と略す。FIM については、Appendix. 2 に従って、学習を行う。

今回のシミュレーションでは、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 ($y=\tanh(x)$) 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。

4.1 逆運動学モデルの学習

本節では、2章で提案した逆運動学モデルの学習法によって、逆運動学モデルを獲得するシミュレーションを行う。

4.1.1 ヤコビ行列のオフライン学習

逆運動学モデルの学習を行う前に、ヤコビ行列の学習を行う。入力層より順に、3個、15個、15個、6個のニューロンによって構成される神経回路を用いる。入力である関節角空間 θ の各座標軸をそれぞれ7等分し、分割によって生成される計 $8 \times 8 \times 8 = 512$ 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、これを教師信号とする。計2,048回のアームの動作によって、教師信号を計算できる。このような教師信号をもとに、共役勾配法を用いて学習を行った。共役勾配法の反復回数は500回であり、反復20回ごとに、最急降下方向を修正ベクトルの方向とした[20][21] (Appendix. 3 参照)。

学習の結果、ヤコビ行列を学習する学習要素の出力を $\Phi_f(\theta)$ とすると、ヤコビ行列の近似誤差を、 $\mathbf{J}(\theta)$ のフロベニウスノルム $|\mathbf{J}(\theta)|_F$ で正規化して表すと、1,000個の代表点における平均値は、

$$E\left(\frac{|\Phi_f(\theta) - \mathbf{J}(\theta)|_F}{|\mathbf{J}(\theta)|_F}\right) = 0.007$$

となった。 $\Phi_f(\theta)$ から計算された疑似逆行列 $\Phi_f^*(\theta)$ の精度は

$$E\left(\frac{|\Phi_f^*(\theta) - \mathbf{J}^*(\theta)|_F}{|\mathbf{J}^*(\theta)|_F}\right) = 0.012$$

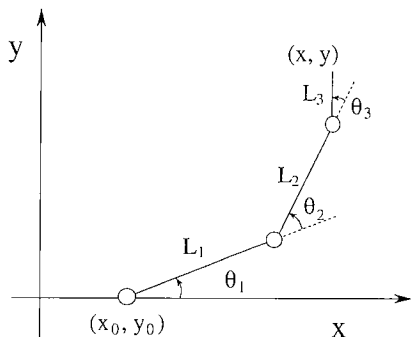


Fig. 2 Configuration of arm

となった。ヤコビ行列の近似精度は十分高い。

4.1.2 逆運動学モデルの学習

未学習の状態から、逆運動学モデルの学習を行うシミュレーションを行う。ヤコビ行列の学習と同様、関節角空間の各軸を7等分し、分割によって生じる512個の格子点を代表点とし、代表点に対応するアーム先端位置を目標位置として定める。また、1,000個の点を、一様乱数によって生成し、それに対応するアームの先端位置を、チェック用目標点とする。チェック用目標点を逆運動学モデルに入力した時のアーム先端位置の RMS (Root Mean Square) 誤差によって逆運動学モデルの精度の評価を行う。

学習要素は、入力層より順に、2個、15個、15個、3個のニューロンによって構成されており、決定すべきパラメータの数は300個である。神経回路の結合荷重の初期値は $(-0.5, 0.5)$ に分布する一様乱数で与える。

教師データの更新を行わずに、共役勾配法による学習を行った場合、チェック用目標点についてのアーム先端位置の RMS 誤差は5.5[mm]程度であり、精度は不十分であった。そこで、反復100回目と反復1,000回目の時点において、教師データの更新を行うことにする。反復100回目と反復1,000回目以外に教師データを更新しない場合、反復100回目と1,000回目の更新に加えて、反復1万回ごとに教師データの更新を行う場合、反復100回ごとに教師信号の更新を行う場合の三つの場合について、シミュレーションを行う。それぞれの場合について、NDでは、順にND0、ND1、ND2と名前を付ける。GLについても、GL0、GL1、GL2と名前を付ける。

共役勾配法のような局所的な情報に基づく最適化手法では、局所最適解に収束してしまう可能性がある。そのため、今回のシミュレーションでは、10種類の初期値から最適化を行った。共役勾配法の反復1万回の時点における、逆運動学モデルの精度を Table 1 に示す。表中の T_i は教師データの更新間隔を示し、 N_T はアームの動作回数の総和の平均値を示す。ただし、RMS 誤差の評価のための動作回数は含めていない。MINE は、10種類のシミュレーション結果のうち、アーム先端位置の RMS 誤差が最小になったものの値であり、MEANE は、10種類のシミュレーション結果のアーム先端の RMS 誤差の平均値であり、 σ はその標準偏差である。

続いて、1万回の反復の結果、最も誤差が少なかった逆運動学モデルについて、各手法により、さらに最大4万回の反復改良を行う。1,000回置きに逆運動学モデルの精度を調べ、チェック用目標点の RMS 誤差 (図中の RMSE) が最小になった時点での RMS 誤差を Table 2 に示す。表中の、 N_{io} は、その時点での共役勾配法の反復回数である。

シミュレーションの結果、提案法は FIM に比べて、対象システムの動作回数が少なく、また獲得される逆モデルの精度も高かった。GL の対象システムの動作回数 N_T は ND の数分の1以下になっており、GLの方がNDよりも有利である。提案法においても、ND2やGL2は、アーム動作の回数がかなり大きいため、実用的では無い。今回のシミュレーションでは、GL1が、学習に必要なアームの動作回数が少なく、また、学習された逆運動学モデルの精度も比較的高く、性能が高かつ

Table 1 Results of offline learning after 10,000 iterations of conjugate gradients method

Method	T_U	N_T	MINE [m]	MEANE[m]	σ
ND 0, ND 1	—	1.41×10^5	4.12×10^{-3}	5.14×10^{-3}	7.24×10^{-4}
ND 2	100	2.66×10^6	4.11×10^{-3}	5.13×10^{-3}	5.40×10^{-4}
GL 0, ND 1	—	3.77×10^4	4.90×10^{-3}	5.98×10^{-3}	8.60×10^{-4}
GL 2	100	7.67×10^5	4.57×10^{-3}	5.41×10^{-3}	7.93×10^{-4}
FIM	—	9.41×10^7	8.23×10^{-3}	8.99×10^{-3}	5.99×10^{-4}

Table 2 Examples of results of offline learning

Method	T_U	N_{Io}	N_T	RMSE [m]
ND 0	—	33,000	1.41×10^5	4.08×10^{-3}
ND 1	10,000	20,000	2.43×10^5	3.76×10^{-3}
ND 2	100	33,000	1.24×10^7	3.58×10^{-3}
GL 0	—	32,000	3.77×10^4	4.02×10^{-3}
GL 1	10,000	45,000	8.26×10^4	3.33×10^{-3}
GL 2	100	50,000	8.74×10^5	3.22×10^{-3}
FIM	—	18,000	1.70×10^8	7.97×10^{-3}

た. 適切な教師信号の更新間隔を採用した提案法の有効性が確認された.

4.2 順モデルの利用

ヤコビ行列の場合と同様, 512 個の格子点について, ロボットアームの先端位置を教師信号とし, 順モデルの学習を行った. 順モデルは, 入力層より順に, 3 個, 15 個, 15 個, 2 個のニューロンによって構成されている. 共役勾配法による学習によって, 1,000 個の代表点における順モデルの RMS 誤差は 6.6×10^{-4} [m] となった. これを利用して, 逆モデルのオフライン学習を行う.

SPARC station 10 (45 [MHz]) において, 1,000 回の反復学習を行った場合, FIM が 4,481 秒かかったのに対して, ND 0, ND 1 では 2,198 秒, ND 2 では 2,364 秒, GL 0, GL 1 では 2,182 秒, GL 2 では 2,294 秒であった. 同じ反復回数で比較した場合, 提案法は, 従来法に比べて, 計算時間が短くて済む. ただし, 実際にアームを動作させる場合に比べれば, 学習にかかる時間の優位性は小さい.

順モデルを利用した場合, 共役勾配法の反復 1 万回の時点において, 順モデルを用いない Table 1 の場合に比べて, MINE, MEANE が増大し, 0.001 [m] 程度, 逆モデルの精度が悪化した. これは順モデルのモデル化誤差のためと考えられる.

順モデルを利用して, 共役勾配法の 3 万回の反復を行った後, 実際にアームを動作させて, ND 0, GL 0 によって, 学習を行った結果を Table 3 に示す. 順モデルを用いない場合に比べて, 非常に少ない動作回数で, 同程度の精度の逆運動学モデルを学習できることが判明した.

4.3 逆運動学モデルとフィードバック系を利用した

アーム先端位置制御

本節では, 学習した逆運動学モデルを利用して, 3 章で述べたアーム先端位置制御系を構成し, その性能評価を行う.

アーム先端位置の目標軌道の生成は以下のように行う. まず一様乱数によって関節角ベクトル θ_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, 正規乱数 ω

Table 3 Examples of results of offline learning after using forward model

Method	N_{Io}	N_T	RMSE [m]
ND 0	19,000	6.57×10^4	3.48×10^{-3}
GL 0	20,000	1.89×10^4	3.42×10^{-3}

Table 4 Simulation results of hand position control

Method	P (%)	RMSE [m]	MAXE [m]
WOIM 1	71.7	1.36×10^{-4}	8.00×10^{-3}
WOIM 2	89.4	1.74×10^{-4}	2.42×10^{-3}
IM	99.2	3.29×10^{-3}	1.51×10^{-2}
WIM 1	98.6	1.52×10^{-4}	1.78×10^{-3}
WIM 2	98.6	2.03×10^{-4}	1.96×10^{-3}

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 Δx_c の第 i 成分を

$$\Delta x_{ci} = 0.05 \omega \quad [\text{m}] \quad (22)$$

のように定め, 次のように円の中心 x_c を定める.

$$x_c = x_s - \Delta x_c \quad (23)$$

動作時間を T として, 反復 k 回目における, 円の中心回りの回転角 $\theta_c(k)$ を次のように定め,

$$\theta_c(k) = \frac{2\pi k}{T} \quad (24)$$

$k=0, 1, \dots, 10T$ において, 円軌道を次式で生成する.

$$x_d(k) = x_c + R(k) \Delta x_c \quad (25)$$

$$R(k) = \begin{bmatrix} \cos \theta_c(k) & -\sin \theta_c(k) \\ \sin \theta_c(k) & \cos \theta_c(k) \end{bmatrix}$$

式(25)のような円軌道のうち, 逆運動学解が存在するものを目標軌道とする. 2,500 回の円軌道のうち, 1,539 個の実現可能な軌道が生成された. T を 100 とする.

軌道追従誤差 $e(k)$ を,

$$e(k) = x_d(k) - f(\theta(k)) \quad (26)$$

とし, $|e(k)|$ の RMS (Root Mean Square) エラーと最大値によって, 逆運動学解法システムの性能を評価する.

逆運動学モデルを用いず, 式(10)によって関節角軌道を計算したものを, WOIM1 (WithOut Inverse Model No. 1) と略す. 同様に, 式(17)によって関節角軌道を計算したものを, WOIM 2 (WithOut Inverse Model No.2) と略す. β の値は 0.1 とする. 式(18)によって逆運動学モデルのみを用いて, 関節角軌道を計算したものを IM (Inverse Model) と略す. フィードバック系を主として用い, 逆運動学モデルを補助的に用いた式(19)によって関節角軌道を計算したものを WIM 1 (With Inverse Model No. 1), 逆運動学モデルを主として用い, フィードバック系を補助的に用いた式(20)によって関節角軌道を計算したものを WIM 2 (With Inverse Model No. 2) と略す. なお, WOIM 1, WOIM 2 において, $\theta(t)$ の初期値 $\theta(0)$ は

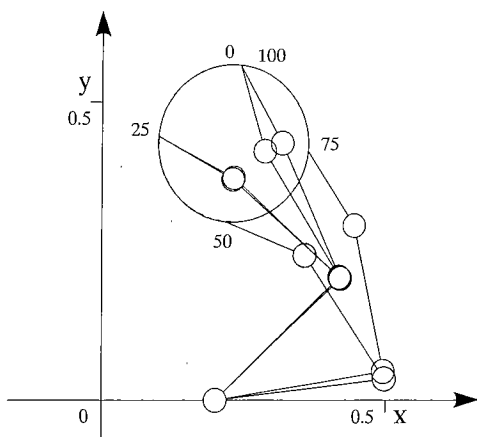
$$\theta(0) = \theta_s \quad (27)$$

とし, WIM 1 の $\theta(t)$ の初期値 $\theta(0)$ は,

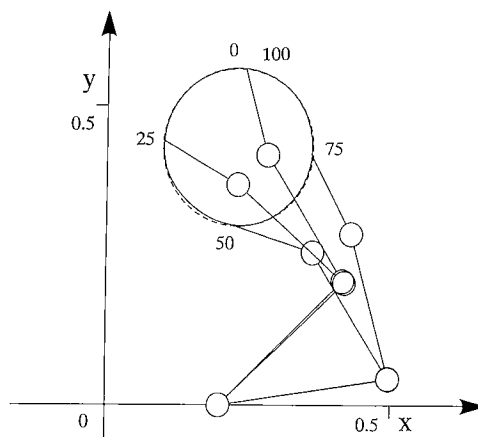
$$\theta(0) = \Phi(x_d(0)) + J^*(\Phi(x_d(0)))(x_d(0) - f(\Phi(x_d(0)))) \quad (28)$$

WIM 2 の $\Delta \theta(t)$ の初期値 $\Delta \theta(0)$ は,

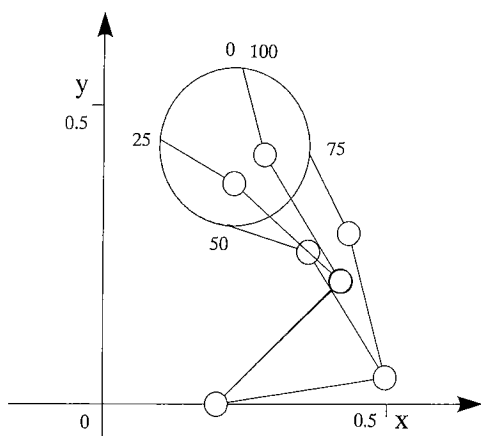
$$\Delta \theta(0) = J^*(x_d(0))(x_d(0) - f(x_d(0))) \quad (29)$$



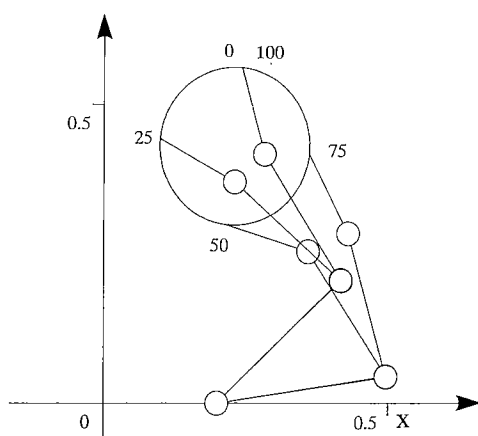
(a) WOIM 2 (hand position controller WithOut Inverse Model no. 2)



(b) IM (hand position controller by Inverse Model)



(c) WIM 1 (hand position controller With Inverse Model no. 1)



(d) WIM 2 (hand position controller With Inverse Model no. 2)

Fig. 3 Path of Hand

と置く。逆運動学モデルとしては、4.1.2項において、GL1によって学習した神経回路を用いる。

シミュレーションの結果を、Table 4に示す。表中のPは、実現可能な軌道のうち、関節角のストロークエンドを越えることなく、目標軌道に追従した割合を示す。RMSEは追従誤差の平均値であり、MAXEは追従誤差の最大値である。

Fig. 3に各手法による、軌道追従の一例を示す。破線は目標軌道であり、実線はアームの実際の軌道である。図には $k=0, 1, \dots, 10T$ における軌道を示している。また、 $k=qT+p$ ($q=0, 1, \dots, 9, p=0, 25, 50, 75, 100$) の時刻に対応するアームの姿勢を示している。アーム先端に p の値を示した。WOIM 1は、関節角がストロークエンドを越えてしまい、この軌道に追従することはできなかった。

WOIM 1やWOIM 2のように逆運動学モデルを用いないシステムでは、関節角のストロークエンドを越えてしまうことがあり、適切な関節角軌道の設計を行えば実現可能な軌道であっても、追従できない場合が多い。IMのように、逆運動学モデルのみを用いた手法は、ストロークエンドを越えることはほとんど無いが、精度が低い。逆運動学モデルとフィードバック系の両方を利用したWIM 1とWIM 2では、関節角が逆運動学モデルの出力の付近にくるように生成されるため、関節角指令

がストロークエンドを越えることはほとんど無く、追従可能な軌道が多い。逆運動学モデルとフィードバック系の併用の有効性が確認された。

5. おわりに

ロボットアームの逆運動学問題を解くために、拡張フィードバック系を利用した逆運動学モデルのオフライン学習方式を提案した。提案法によって、従来法よりも少ない対象システムの動作回数で、逆モデルの学習が可能であることをシミュレーションによって確認した。さらに、逆運動学モデルと誤差のフィードバック系とを組み合わせた逆運動学解法システムによって、ストロークエンドを越えることなく、アーム先端位置の制御を行えることを確認した。

現時点において、教師データ更新のスケジュール管理が不完全であり、また次元が増加すると、計算時間が非常に大きくなるという問題点はあるが、逆運動学モデルを学習によって獲得する有効な方法となる可能性がある。

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 Directed 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(I)", 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, 久保田孝, 佐藤基夫, 原島文雄: "感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] 大山英明, 館暉: "多対一対応システムの逆モデル学習の一手法", 第3回インテリジェントシステム・シンポジウム, pp. 305-310, 1993.
- [10] 大山英明, 館暉: "拡張フィードバック系による逆モデル学習", 電子情報通信学会論文誌 D-II, vol. J 78-D-II, no. 2, pp. 349-357, 1995.
- [11] D. E. Rumelhart, G. E. Hinton and R. J. Williams: "Learning Internal Representation by Error Propagation," Parallel Distributed Processing, D. E. Rumelhart and J. L. McClelland and the PDP Research Group ed., pp. 318-326, MIT Press, 1986.
- [12] D. G. Luenberger: "Optimization by vector space methods," John-Wiley & Sons, New York, 1969. 日本語訳 増淵正美, 加納秀明訳: "関数解析による最適理論", コロナ社, p. 304, 1973.
- [13] R. Fletcher and C. M. Reeves: "Function Minimization by Conjugate Gradients," Computer Journal, 10, pp. 149-154, 1964.
- [14] 大山英明, 館暉: "神経回路によるロボット・アームの逆キネマティクス問題の解法", SICE '92 予稿集, pp. 695-696, 1992.
- [15] E. Oyama and S. Tachi: "A Learning Method for Solving Inverse Problems of Static Systems," Proc. of 1993 International Joint Conference on Neural Networks, pp. 2843-2849, 1993.
- [16] 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 D-II, vol. J 77-D-II, no. 7, pp. 1345-1354, 1994.
- [17] 大山英明, 館暉: "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", 日本ロボット学会誌, vol. 13, no. 1, pp. 89-99, 1995.
- [18] A. Liegeois: "Automatic Supervisory Control of the Configuration and Behavior of Multibody Mechanisms," IEEE Trans. SMC-7, 12, pp. 868-871, 1977.
- [19] 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.
- [20] 嘉納秀明: "システムの最適理論と最適化", コロナ社, pp. 84-85, 1987.
- [21] 嘉納秀明: "システムの最適理論と最適化", コロナ社, pp. 234-270, 1987.

Appendix. 1 拡張フィードバック系のインプリメンテーション

本論で構成した逆運動学モデルを学習するための拡張フィードバック系の具体的な構成について述べる。

A1.1 反復改良のアルゴリズム

今回の構成では, 次式のような広義のニュートン法を実現するように, 反復改良演算を行う。

$$\boldsymbol{\theta}(k+1) = \boldsymbol{\theta}(k) + a(k) \mathbf{J}_a^{-1}(\boldsymbol{\theta}(k)) (\mathbf{x}_d - \mathbf{f}(\boldsymbol{\theta}(k))) \quad (\text{A 1.1})$$

このほか, Fun らが提案しているように, 順モデルを利用して, 最急降下方向を計算し, 共役勾配法[12]を利用することも有効である[8]。

A1.2 係数の決定法

式(A 1.1)の反復改良の係数 $a(k)$ の決定法として, 本稿では, 非常に簡単な手法を用いる。 p を非負の整数として,

$$a(k) = \left(\frac{1}{2}\right)^p \quad (\text{A 1.2})$$

とおき, p を非線形性の度合いに応じて調節し, 反復計算において, 出力誤差の二乗和関数

$$S(\mathbf{x}_d, \boldsymbol{\theta}) = \|\mathbf{x}_d - \mathbf{f}(\boldsymbol{\theta})\|^2 \quad (\text{A 1.3})$$

が減少することを保証する。すなわち,

$$S(\mathbf{x}_d, \boldsymbol{\theta}(k+1)) \leq S(\mathbf{x}_d, \boldsymbol{\theta}(k)) \quad (\text{A 1.4})$$

が成立するまで, p を増大させ, $a(k)$ を $1/2$ にしていく。反復の次のステップでは, $a(k)$ が $1/2$ 以下であれば, p を一つだけ減らし,

$$a(k+1) = 2.0a(k) \quad (\text{A 1.5})$$

とし, $a(k)$ が 1 であれば, $a(k+1)$ も 1 とする。無限ループに陥らないように, p には上限を設ける。なお, p の初期値は 2 とする。

このような簡単な直線探索を行うため, 式(A 1.1)の更新演算に必要なシステムの動作回数は一定では無い。

A1.3 関節角の制限

通常のロボットには, 関節角の可動範囲に制限があり, 式(A 1.1)に従って, 教師信号を計算する場合, $\boldsymbol{\theta}(k)$ が可動範囲を越えてしまう場合がある。

反復計算を打ち切り, 新たな初期値 $\boldsymbol{\theta}(0)$ から出発して, 再び計算を行うことも一つの方法であるが, ロボットの運動学システムでは, 反復回数が増大する場合が多い。そこで, 反復中に, 関節角の可動範囲を越えてしまうような計算値が計算された場合は, その関節角のストロークエンドの値をとって, 計算値を修正することにする。

A1.4 局所最小解の判別条件

反復の停止の条件は, 出力誤差が目標値以下になった場合と目標値が与えられてからの初期値変更の回数が一定回数を越えた場合とする。本稿のシステムでは, 出力誤差の目標値は 1.0×10^{-4} [m] とする。初期値変更の回数の最大値は 200 回とし, 一つの初期値から出発する反復回数の上限も 200 回とする。これらの数値は, 経験的に定めたものである。

次式が成立した場合には, 局所最小解に収束したものと見なし, 反復演算の初期値の変更を行う。

$$\|\mathbf{x}(k+1) - \mathbf{x}(k)\| < \gamma \|\mathbf{x}_d - \mathbf{x}(k)\| \quad (\text{A 1.6})$$

γ は十分小さな正の実数で, 本稿では, 0.001 と置いた。

Appendix. 2 従来法による逆運動学モデル獲得

通常の順逆モデリングやフィードバック誤差学習では, 学習要素に出力目標値を入力し,

$$\boldsymbol{\theta} = \boldsymbol{\Phi}(\mathbf{x}_d) \quad (\text{A 2.1})$$

学習要素の出力 $\boldsymbol{\Phi}(\mathbf{y}_d)$ を得て, 対象システムの入力とする。出力目標値 \mathbf{x}_d と実際の出力 $\mathbf{f}(\boldsymbol{\theta})$ との誤差をバックプロパゲーション法やフィードバック系によりフィードバックし, 学習

要素 $\Phi(x_d)$ の教師信号 $\Phi'(x_d)$ を次式のように計算し、学習を行う。

$$\begin{aligned}\Phi'(x_d) &= \Phi(x_d) + \Delta\Phi(x_d) \\ \Delta\Phi(x_d) &= K(\Phi(x_d))(x_d - f(\Phi(x_d)))\end{aligned}\tag{A 2.2}$$

順逆モデリングの場合、対象システム $f(\theta)$ のヤコビ行列 $J(\theta) = \partial f(\theta) / \partial \theta$ の転置行列 $J(\theta)^T$ に適当な正の係数 β をかけた行列を $K(\theta)$ は近似している。すなわち

$$K(\theta) \approx \beta J(\theta)^T \tag{A 2.4}$$

である。フィードバック誤差学習では、特に $K(\theta)$ は決まっていなくても、ヤコビ行列 $J(\theta)$ の転置行列 $J(\theta)^T$ を利用して、

$$K(\theta) = \beta J(\theta)^T \tag{A 2.5}$$

としたり、ヤコビ行列の一般化逆行列 $J(\theta)^*$ を用いて、

$$K(\theta) = \beta J(\theta)^* \tag{A 2.6}$$

とする場合が多い。

関節角の可動領域に制限がある通常のロボットアームの逆運動学モデルを獲得するためには、出力誤差の評価関数

$$\begin{aligned}V &= \sum_i S(x_{di}) \\ S(x_d) &= |x_d - f(\Phi(x_d))|^2\end{aligned}\tag{A 2.7}$$

を最小化するだけでは不十分であり、

$$\theta_{\min i} \leq \theta_i \leq \theta_{\max i} \quad (i=1, \dots, m) \tag{A 2.8}$$

という不等式拘束条件を満たさなければならない。

今回は、ペナルティ法によって、式(A 3.8)の不等式拘束条件を扱う。すなわち、次式のようなペナルティ関数 $P(\theta)$ を用意する。

$$\begin{aligned}P(\theta) &= \sum_j P_j(\theta_j) \\ P_i(\theta_i) &= \begin{cases} 0 & (\theta_{\min i} \leq \theta_i \leq \theta_{\max i}) \\ \mu(\theta_i - \theta_{\min i})^2 & (\theta_i < \theta_{\min i}) \\ \mu(\theta_i - \theta_{\max i})^2 & (\theta_i > \theta_{\max i}) \end{cases}\end{aligned}\tag{A 2.9}$$

μ は十分大きな正の実数である。今回のシミュレーションでは、100とした。次のような拡張された誤差の二乗和関数 V' を共役勾配法のような数値最適化の手法を用いて最小化することによって、逆モデル $\Phi(x_d)$ の学習が可能となる。

$$\begin{aligned}V' &= \sum_i S'(x_{di}) \\ S'(x_d) &= |x_d - f(\Phi(x_d))|^2 + P(\Phi(x_d))\end{aligned}\tag{A 2.10}$$

Appendix. 3 共役勾配法

推定すべき変数の多い最小二乗最適化問題においては、共役勾配法は有力な手法である。推定すべき N_w 次元ベクトルを w とし、最小化すべき評価関数を $V(w)$ と置く。非線形の共役勾配法では、探索方向を定期的に最急降下方向に取り直すのが、その間隔を n_c とする。 $V(w)$ が二次関数の場合、 $n_c = N_w$ とするが、非線形系では、 n_c は N_w よりも小さい数にした方がよい。共役勾配法のアルゴリズムは、以下の通りである。アルゴリズムの記述は、文献[20]を参考にした。また、本稿で用いたプログラムは、同文献の付録[21]を基に作成した。

- (I) 適当な手法で w の初期値 $w(0)$ を生成し、 $k=0$ と置き、反復回数 $N_r=0$ と置く。
- (II) 評価関数 $V(w)$ の最急降下方向 $g(w)$ を探索方向 $\Delta w(0)$ として定める。

$$g(w) = -\frac{\partial V(w)}{\partial w}$$

$$\Delta w(0) = g(w)$$

- (III) $V(w(k) + \alpha \Delta w(k))$ を最小化する α を直線探索によって求め、その α を $\alpha(k)$ とし、次のように $w(k)$ を更新する。

$$w(k+1) = w(k) + \alpha(k) \Delta w(k)$$

- (IV) 収束条件のチェックを行い、収束と判断した場合には終了する。 $N_r = N_r + k$ とする。

- (V) k が $n_c - 1$ であれば、(VIII)へ行く。

- (VI) 次式のように $\Delta w(k)$ を更新する。

$$\Delta w(k+1) = g(w(k+1)) + \beta(k) \Delta w(k)$$

$$\beta(k) = \frac{g(w(k+1))^T g(w(k+1))}{g(w(k))^T g(w(k))}$$

- (VII) $k = k + 1$ とし、(III)へ行く。

- (VIII) $w(0) = w(n_c)$, $k = 0$ と置き、 $N_r = N_r + n_c$ とし、(II)へ行く。



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課程修了。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。トレイグジスタンス、人間の運動制御系のモデル化、神経回路、非線形最適化などの研究に従事。電子情報通信学会、計測自動制御学会などの会員。(日本ロボット学会正会員)



館 暉 (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) TC 17 (ロボティクス) 議長。SICEフェロー。パイスベクトルを用いる信号処理、盲導ロボット、トレイ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)