論　文

逆ダイナミックスモデルを用いた反復制御による運動適応

大谷　将司[†]　　田地　宏一[†a)]　　宇野　洋二[†b)]

Motion Adaptation with Iterative Control Using Inverse-Dynamics Model

Masashi OTANI[†], Kouichi TAJI[†a)], and Yoji UNO[†b)]

あらまし　人の脳内には内部モデルと呼ばれるフィードフォワード制御を担う機構があると考えられている．しかし，様々な運動環境に対し固有の内部モデルを保有しているとは考えにくく，主となる内部モデルを利用し，その出力を修正し目的の動作に合った運動指令の生成を行っていると考えられる．この考えに基づき，運動環境の変化に対する運動適応手法を提案する．提案する手法では，まずニューラルネットワークに制御対象のダイナミックスを学習し，逆モデル（内部モデル）を構成する．そして，運動環境の変化に対し逆モデルを用いて運動指令の誤差を推定し修正する．計算機シミュレーションと実機実験による検証を行い，この手法の有効性を確認した．

キーワード　逆ダイナミックスモデル，反復制御，運動適応，ニューラルネットワーク

1. まえがき

人の複雑で滑らかな運動は，フィードバック制御だけでは難しいことから，脳内には「内部モデル」[1] と呼ばれるフィードフォワード制御をする機構があると考えられている．この機構は運動パターン（目標軌道）を入力し，運動指令が出力されるので，情報の流れが制御対象のダイナミックスと逆向きになっており，逆ダイナミックスモデル，あるいは単に逆モデルと呼ばれている．人は繰り返し同じ運動を行うことで，内部モデルを学習し，運動のパフォーマンスの向上を行っていると考えられている．

このような学習機構を計算機上で再現し工学的に応用する試みの一つとして，ニューラルネットワークに制御対象の逆モデルを学習することが行われてきた．ニューラルネットワークは，脳の神経細胞のネットワークをモデル化したもので，ニューロン間の結合荷重の値を調整することで制御対象の逆モデルを学習する能力をもつ．Kawato ら [2], [3] によるフィードバック誤差学習や Jordan ら [4] による順逆モデリングに代表されるこれらの試みは，ロボットなどの制御対象を繰返し動かすことで，学習に必要な教師信号を求め，それを用いてニューラルネットワークを学習することで，逆モデルを獲得するものである．

ここで，手に物を持つなどの運動環境の変化に伴い，制御対象のダイナミックスが変化した場合を考える．そこでは，獲得していた逆モデルと望ましい逆モデルの間に誤差が生じるため，目標軌道を実現する運動指令を生成することができなくなる．上記で述べた学習スキームでは，運動環境の変化が起こると，逆モデルを表現するニューラルネットワークを再学習することで新しい環境にあった逆モデルを生成して運動環境に対応する．しかし，運動環境の変化のたびに新しく逆モデルを獲得する，若しくは再学習を繰り返すことは無数の逆モデルが必要となるという問題が生じる．そこで，そのような場合には逆モデルを学習するのではなく，逆モデルから出力される運動指令を修正することで変化に対応していくのではないかと考えることができる．ここでは，このような手法を学習に対し適応と呼ぶ．また，その際，現在保有している内部モデルの情報を利用して運動指令を修正していくことが有用であると考えられる．

ロボット制御の立場から，この問題に対するアプローチがなされ，内部モデルを用いず目標軌道に対する運動指令を求めるスキームが提案されている．代表

[†] 名古屋大学大学院工学研究科，名古屋市
Graduated School of Engineering, Nagoya University, Furo-cho, Chikusa-ku, Nagoya-shi, 464–8603 Japan
a) E-mail: taji@nuem.nagoya-u.ac.jp
b) E-mail: uno@nuem.nagoya-u.ac.jp

的なものとして，Arimoto ら [5] が提案した反復学習制御がある．これは目標軌道と実現した軌道の誤差や誤差の微分に適当なゲイン行列をかけることで，運動指令値の修正量を導出し運動指令値を修正する．これを繰返し行うことで目標とする軌道を達成する運動指令値を求めることができる．反復学習制御では，運動環境の変化に対し，運動指令を直接修正することで適応することができるが，制御対象に対して適切なゲイン行列を見つけなければならないという問題がある．また，スキーム自体に内部モデルを組み込んでおらず，制御対象の情報を事前に獲得していたとしても，その情報を活用できない．

ところで，宇野ら [6] は制御対象の逆モデルを利用することで，運動指令誤差を推定する手法を提案した．この手法では，事前に学習した逆モデルの勾配情報により，運動軌道の誤差から運動指令の修正量を求める．また，永澤ら [7] はこの手法をニューラルネットワーク上に表現した逆モデルを用いた形にまとめ，逆モデルの学習に利用し Forward-propagation 学習則を提案した．

本論文では，永澤らの運動指令誤差の推定法を反復制御における運動指令の修正に用い，運動環境の変化に適応する（運動適応）手法を提案する．提案する手法ではあらかじめ，ニューラルネットワークに制御対象のダイナミックスを学習することで，逆モデルを獲得しておく．そして，運動環境の変化に対して事前に獲得した逆モデルを用いて運動指令の誤差を推定し，修正することで目標軌道を実現する運動指令を求める．この制御の特徴は，逆モデルをニューラルネットワークに表現することで，制御対象のノミナルモデルをもたずに制御を行えることと，軌道誤差から運動指令誤差を推定する誤差変換に逆モデルを利用することにある．

2. で，逆ダイナミックスモデルを用いた反復制御アルゴリズムについて説明し，**3.** で，2 リンクアームについての計算機シミュレーションと，1 リンクアームについての実機を用いた実験の結果を示す．最後に **4.** で，提案手法の有効性と問題点について論ずる．

2. 反復制御のための計算スキーム

目標軌道に対する運動指令を得るために，制御対象を繰返し動かして運動軌道の誤差を計測し，それに基づき運動指令を修正することにより，望ましい運動指令を求める方法を考える．

反復制御のスキームを図 1 に示す．目標軌道 $\boldsymbol{x}_d(t)$ が与えられたとき，1 回目の試行では，図 2 に示す 3 層からなるニューラルネットワークにより表現された逆ダイナミックスモデルによる近似逆変換 $\tilde{f}$ を実行し，その出力を制御対象と反復システム内の運動指令メモリに送る．2 回目以降の n 回目から $n+1$ 回目の試行は以下のように決められる．運動指令 $\boldsymbol{\tau}^n(t)$ に対する軌道誤差信号 $\delta\boldsymbol{x}^n(t) = \boldsymbol{x}_d(t) - \boldsymbol{x}^n(t)$ は，逆モデルを利用した誤差変換によって，運動指令の修正量 $\delta\boldsymbol{\tau}^n(t)$ へと変換される．この誤差変換の詳細については **2.1** で説明する．誤差変換により得た運動指令の修正量を用いて，n 回目から $n+1$ 回目で運動指令が次式のように更新される．

$$\boldsymbol{\tau}^{n+1}(t) = \boldsymbol{\tau}^n(t) + \delta\boldsymbol{\tau}^n(t) \tag{1}$$

この制御過程を繰り返し行うことで，目標軌道に対する望ましい運動指令が求められる．また，得られた運動指令を教師信号として，ニューラルネットワークの学習を行えば，その目標軌道を追従する運動指令を与える逆モデルを得ることができる．

2.1 運動指令誤差の推定

逆ダイナミックスモデルを用いた運動指令誤差の推定について説明する．この手法は制御対象の逆モデルが近似的に形成されている（近似逆モデル）とき，入力信号の誤差から出力信号の誤差を導く手法である．

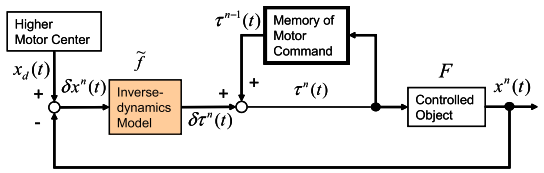

図 1 反復制御スキーム
Fig. 1 An iterative control scheme.

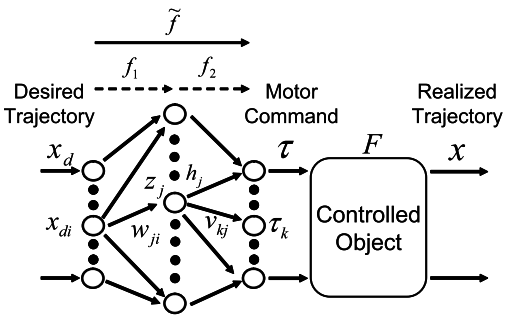

図 2 逆モデルの入出力関係
Fig. 2 Input-output relation of inverse model.

逆ダイナミックスモデルを表現する入力層 l 個，中間層 m 個，出力層 n 個のニューロンをもつ 3 層からなるニューラルネットワークへの入力信号を x_{di}，中間層への入力信号を z_j，z_j に対してシグモイド関数を施した中間層の出力を h_j，出力層の出力を τ_k とすると各層における信号は次のようになる．

$$z_j = \sum_{i=1}^{l} w_{ji} x_{di} \qquad (j = 1 \ldots m) \quad (2)$$

$$h_j = \frac{1}{1 + e^{-z_j}} \qquad (j = 1 \ldots m) \quad (3)$$

$$\tau_k = \sum_{j=1}^{m} v_{kj} h_j \qquad (k = 1 \ldots n) \quad (4)$$

ただし，w_{ji} は入力層の i 番目のニューロンから中間層の j 番目のニューロンへの結合荷重，v_{kj} は中間層の j 番目のニューロンから出力層の k 番目のニューロンへの結合荷重である．

入力層から中間層入力への変換を $\boldsymbol{z} = f_1(\boldsymbol{x})$，中間層入力から出力層への変換を $\boldsymbol{\tau} = f_2(\boldsymbol{z})$ とする．ニューラルネットワークにおける変換 $f_2 \circ f_1$ に制御対象 F の近似逆モデル $\tilde{f}$ が形成されていると仮定すれば，目標軌道から見た実現軌道について

$$F \circ \tilde{f} = F \circ f_2 \circ f_1 \simeq I \quad \text{(恒等写像)} \qquad (5)$$

が成り立つとみなせる．

目標軌道 $\boldsymbol{x}_d$ と実現軌道 $\boldsymbol{x}_r$ の誤差 $\delta\boldsymbol{x} = \boldsymbol{x}_d - \boldsymbol{x}_r$ を小さくするために，ニュートン法を用いて中間層への入力信号 $\boldsymbol{z}$ に対する目標信号 $\hat{\boldsymbol{z}}$ を逐次的に更新するならば

$$\hat{\boldsymbol{z}} = \left[\frac{\partial\{F \circ f_2(\boldsymbol{z})\}}{\partial \boldsymbol{z}}\right]^{-1} (\boldsymbol{x}_d - \boldsymbol{x}_r) + \boldsymbol{z} \qquad (6)$$

となる．一般に，逆関数のヤコビ行列は元の関数のヤコビ行列の逆行列になるので，式 (5) より $\left[\{\partial F \circ f_2(\boldsymbol{z})\}/\partial \boldsymbol{z}\right]^{-1}$ の近似として，$\left[\partial f_1(\boldsymbol{x})/\partial \boldsymbol{x}\right]$ を用いることができるので，式 (6) の代わりに，次式の広義ニュートン法

$$\hat{\boldsymbol{z}} \simeq \left[\frac{\partial f_1(\boldsymbol{x})}{\partial \boldsymbol{x}}\right] (\boldsymbol{x}_d - \boldsymbol{x}_r) + \boldsymbol{z} \qquad (7)$$

となる．これを用いると，中間層に対する修正量 δz_j は次のようにして求められる．

$$\delta x_i = x_{di} - x_{ri} \qquad (8)$$

$$\delta z_j = \sum_i \frac{\partial f_1(x_{di})}{\partial x_{di}} \delta x_i = \sum_i \frac{\partial (w_{ji} x_{di})}{\partial x_{di}} \delta x_i$$
$$= \sum_i w_{ji} \delta x_i \qquad (9)$$

同様に中間層の出力について $f_1 \circ F \circ f_2 \simeq I$ という関係が成り立つので，$\left[\{\partial f_1 \circ F(\boldsymbol{\tau})\}/\partial \boldsymbol{\tau}\right]^{-1}$ の近似として，$\left[\partial f_2(\boldsymbol{z})/\partial \boldsymbol{z}\right]$ を用いることができるので，中間層入力に対する修正量 $\delta\boldsymbol{z}$ を前向きに伝搬させることにより，出力層に対する修正量 $\delta\tau_k$ の導出が可能である．

$$\hat{\boldsymbol{\tau}} = \left[\frac{\partial\{f_1 \circ F(\boldsymbol{\tau})\}}{\partial \boldsymbol{\tau}}\right]^{-1} \delta\boldsymbol{z} + \boldsymbol{\tau}$$
$$\simeq \left[\frac{\partial f_2(\boldsymbol{z})}{\partial \boldsymbol{z}}\right] \delta\boldsymbol{z} + \boldsymbol{\tau} \qquad (10)$$

$$\delta\tau_k = \sum_j \frac{\partial f_2(z_j)}{\partial z_j} \delta z_j = \sum_j \frac{\partial\{v_{kj} h_j(z_j)\}}{\partial z_j} \delta z_j$$
$$= \sum_j \{h_j(1 - h_j) v_{kj}\} \delta z_j \qquad (11)$$

ここで，$h_j(1-h_j)$ 項の値域は (0,0.25] であり，小さな正定数 ϵ で近似できる．すると，式 (9) より式 (11) は

$$\delta\tau_k = \epsilon \sum_j v_{kj} \delta z_j = \epsilon \sum_j \sum_i v_{kj} w_{ji} \delta x_i \qquad (12)$$

となる．この近似は，アルゴリズムを単純化するだけでなく，良い推定を促すことが数値実験により確認されている [8]．

この一連の式は以下のような形にも書き下すことができる．

$$\delta\boldsymbol{\tau} = \left[\frac{\partial f_2(\boldsymbol{z})}{\partial \boldsymbol{z}}\right] \cdot \left[\frac{\partial f_1(\boldsymbol{x})}{\partial \boldsymbol{x}}\right] \delta\boldsymbol{x} = \left[\frac{\partial \tilde{f}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right] \delta\boldsymbol{x} \qquad (13)$$

以上の式により，逆モデルのニュールネットワーク内で軌道誤差 $\delta\boldsymbol{x}$ を前向きに伝搬させることによって運動指令の修正量 $\delta\boldsymbol{\tau}$ を求めることができる．

2.2 初期結合荷重の選定

上記で述べた誤差変換は式 (5) で表される近似逆モデルを獲得しているという仮定の上にある．しかし，一般の学習法ではネットワークの結合荷重の初期値は小さなランダムな値を与えるので，必ずしも適切な近似逆モデルとはならない．先行研究 [6] では，制御対象に適当な振幅と位相をもつ正弦波形の運動指令を入

力し，それぞれに対応する軌道を出力させ，これらの対を訓練パターンとして，ネットワークの学習を行い近似逆モデルを獲得していた．

本論文では新たに，初期結合荷重を適切に設定することで，**2.1** で述べた誤差変換法を用いて運動指令の誤差を推定する方法を提案する．簡単のため，1 入力 1 出力のシステムを考える．逆ダイナミックスモデルは，位置，速度，加速度である $\theta, \dot{\theta}, \ddot{\theta}$ の三つが入力され，運動指令 τ が出力される．ここで，位置，速度，加速度の誤差にゲインをかけたものが出力されるように結合荷重を設定する（図 3 参照）．このように，ある入力に対して一つの中間層ニューロンへの結合荷重を大きくし，更にそのニューロンから関連する出力への結合荷重を大きくしておくと，大きい結合荷重の値をとる項が支配的になる．つまり，式 (12) は以下のように書き下せる．ただし，大きな値をとる結合荷重は，他の結合荷重より十分大きいものとする．経験的には 10 倍程度の差があればよい．

$$\begin{aligned}\delta\tau &\simeq \epsilon(w_{11}v_{11}\delta x_1 + w_{22}v_{12}\delta x_2 + w_{33}v_{13}\delta x_3)\\ &= \epsilon(w_{11}v_{11}\delta\theta + w_{22}v_{12}\delta\dot{\theta} + w_{33}v_{13}\delta\ddot{\theta})\\ &= K_P\delta\theta + K_D\delta\dot{\theta} + K_A\delta\ddot{\theta} \qquad (14)\end{aligned}$$

ここで，K_P, K_D, K_A は，位置，速度，加速度ゲインであり，結合荷重を調整することで，このゲインの値を決めることができる．以上の式より，結合荷重を調整することで，逆モデルを用いた運動指令誤差の推定を，軌道誤差にゲインをかけた形，つまり線形 PDA 制御器に近似した．この手法は多自由度系に対しても拡張可能である．

Arimoto ら [5] によれば，制御対象の慣性行列を $M(\theta)$ とすると，次の条件を満たすとき，反復制御は収束することが示されている．

$$\|I - M(\theta)^{-1}R\|_\infty < 1 \qquad (15)$$

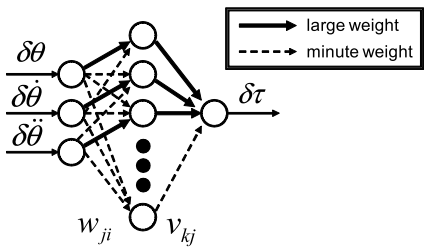

図 3　初期結合荷重の設定
Fig. 3　Set of initial weights.

R は反復制御におけるゲイン行列で，式 (14) の K_A を対角に並べたものに対応している．ここで，慣性行列 $M(\theta)$ を対角行列で近似すると，式 (15) より，加速度ゲイン K_A が慣性項成分の 2 倍より小さいとき，反復制御は収束する．また，K_D, K_P は収束の速さや安定性に関係がある．本論文では，K_A を適当な値からはじめて，収束しなければ小さな値に設定し直した．その後，K_D は収束が速くなるよう試行錯誤的に決めた．また，本論文では，K_P については収束の早さに大きな影響を与えなかったため，簡単のため他の結合荷重と同様に小さな値を取ることとした．

従来研究 [6] では，獲得した近似逆モデルは訓練パターンに用いる正弦波の振幅の大きさやパターン数に依存するため，入出力の汎化性は比較的高いが，目標とする運動軌道において反復制御の収束を必ずしも満たさない．提案手法では，物理パラメータにあったゲインを決めれば，任意の運動軌道において反復制御が収束することが保証されている．しかし，Arimoto ら [5] の反復学習制御と同様に運動環境の変化に対して，ゲインを探索し直さなければならない場合があることに注意する．

3. 実　　験

提案した手法の有効性を確認するため，計算機シミュレーションと実機を用いた実験を行う．各実験は以下の (a), (b), (c) の手順で行うものとする．実験では **2.2** で提案した初期結合荷重を線形 PDA 制御器としたときの運動誤差の推定と，近似逆モデルを利用した反復制御による運動適応の有効性を確認する．

(a) 初期結合荷重からの反復制御

逆モデルが線形 PDA 制御器となるように調整した初期結合荷重から反復制御を行い，運動指令を求める．

(b) 逆ダイナミックスモデルの学習

(a) で得た運動指令を教師信号として，誤差逆伝搬学習則 [9] を用いてニューラルネットワークを学習し，近似逆モデルを獲得する．

(c) 近似逆モデルを利用した反復制御（運動適応）

制御対象におもりを付けることでダイナミックスを変化させ，運動環境を変化させる．そして，新しい運動環境に対して，(b) で得たおもりを付けないときの近似逆モデルをそのまま利用して，反復制御を行う．

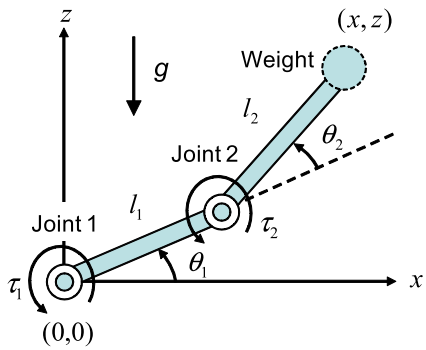

図 4　鉛直面内を動く 2 リンクアーム
Fig. 4　A 2-link arm moving a vertical plane.

表 1　2 リンクアームのパラメータ
Table 1　Physical parameters of a 2-link arm.

	m [kg]	l [m]	s [m]	I [kg· m^2]	g [m/s^2]
Link1	1.59	0.3	0.0992	0.0477	9.81
Link2	1.44	0.35	0.1223	0.162	

3.1　計算機シミュレーション

鉛直面内を動く 2 リンクアーム（図 4）について，目標軌道（関節角度，角速度，角加速度）から運動指令（関節トルク）を求める逆ダイナミックスモデルを用いた運動適応について，計算機シミュレーションを行った．

2 リンクアームのダイナミックスは次式で表される．

$$\tau_1 = M_{11}\ddot{\theta_1} + M_{12}\ddot{\theta_2} + h_{122}\dot{\theta_2}^2 + 2h_{112}\dot{\theta_1}\dot{\theta_2} + G_1$$
$$\tau_2 = M_{21}\ddot{\theta_1} + M_{22}\ddot{\theta_2} + h_{211}\dot{\theta_1}^2 + G_2$$
$$M_{11} = m_1 s_1^2 + I_1 + m_2(l_1^2 + s_2^2 + 2l_1 s_2 \cos\theta_2) + I_2$$
$$M_{12} = M_{21} = m_2(s_2^2 + l_1 s_2 \cos\theta_2) + I_2$$
$$M_{22} = m_2 s_2^2 + I_2$$
$$h_{122} = h_{112} = -h_{211} = -m_2 l_1 s_2 \sin\theta_2$$
$$G_1 = m_1 g s_1 \cos\theta_1 + m_2 g\{l_1 \cos\theta_1 + s_2 \cos(\theta_1 + \theta_2)\}$$
$$G_2 = m_2 g s_2 \cos(\theta_1 + \theta_2) \qquad (16)$$

リンク i に対し $\theta_i, \dot{\theta_i}, \ddot{\theta_i}$ は関節角度，角速度，角加速度，M_{ii} は有効慣性，M_{ij} は相互慣性，h_{ijj} は遠心加速度係数，$h_{ijk}(j \neq k)$ はコリオリ加速度係数，G_i は重力成分を表す．また，リンクの長さ l_i，質量中心までの長さ s_i，質量 m_i，重心周りの慣性モーメント I_i，重力加速度 g であり，各値を表 1 のように設定した．j, k はそれぞれ角速度または角加速度に対応する添字である．

このアームの逆ダイナミックスモデルは，目標軌道における各関節の角度 θ_1, θ_2，角速度 $\dot{\theta_1}, \dot{\theta_2}$ と角加速度 $\ddot{\theta_1}, \ddot{\theta_2}$ が入力され，関節トルク τ_1, τ_2 が出力される．そこで，入力層，中間層，出力層がそれぞれ，6 個，10 個，2 個のニューラルネットワークを用意した．また，式 (12)，式 (14) に示す正定数とゲインは $\epsilon = 0.2, K_A = 0.02, K_D = 0.1$ とした．アームの運動軌道は 4 次の Runge-Kutta 法を用いて計算する．目標軌道は運動開始手先位置 (0.0,0.6) [m]，運動終了手先位置 (0.4,0.1) [m] を結ぶ各関節の角加速度の微分の総和が最小となる関節角躍度最小軌道とした．運動時間 1.0 [s]，サンプリング周波数 100 [Hz] である．また，ダイナミックスを変化させる際には，アームの手先に 0.5 [kg] のおもりを持たせた．

線形 PDA 制御器に近似するよう調整したニューラルネットワークを用いて，反復制御を行った際の手先軌道を図 5 に示す．このとき，図より，50 回の反復制御を行った結果，目標軌道（実線）と実現軌道（点線）が重なっており，目標軌道を描く運動指令を求めることができたことが分かる．また，反復制御を 50 回行って得られた運動指令を教師信号とし，誤差逆伝搬則を用いてニューラルネットワークの学習を行い，近似逆モデルを得た．

次に，アームの手先のおもりを持たせたことでダイナミックスを変化させる．このとき，先ほど得た逆ダイナミックスモデルから出力される運動指令では図 6 の 1 回目の試行（一点鎖線）に示すように目標軌道を追従できなくなる．そこで，おもりを付けないときに獲得した近似逆モデルを用い，提案した反復制御を行い，再び目標軌道を追従する運動指令を得るため運動適応を行った．その結果，図 6 に示すように，20 回の反復制御により目標軌道（実線）と実現軌道（破線）が重なっており，再び望ましい運動指令を得ることができた．

本論文では，運動環境の変化に対しての適応に注目してきたが，提案手法を用いることで，運動タスクの変化への適応も行うことができる．逆ダイナミックスモデルが制御対象に対して完全な形で学習されていれば，目標軌道や運動時間など運動タスクの違いにかかわらず望ましい運動指令を逆モデルから得ることができる．しかし，一般に完全な形で逆モデルを獲得することは難しいことから，運動タスクが変化したとき逆モデルから出力される運動指令と望ましい運動指令に誤差が生じる．そこで，運動環境変化への適応と同様に，運動タスクへの変化に対して反復制御により適応

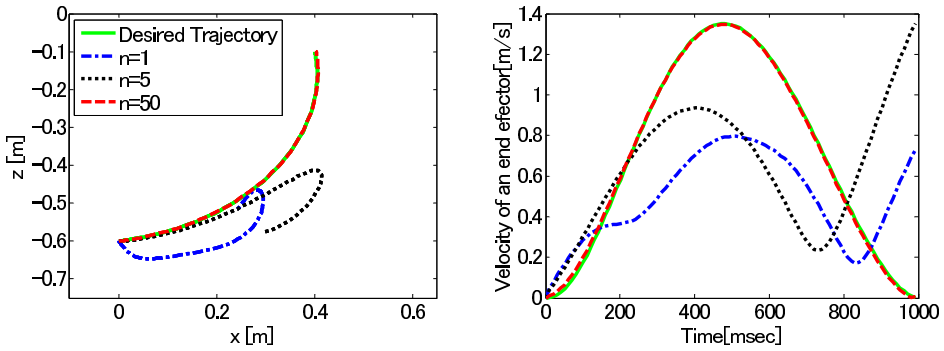

図 5　おもりを持たない場合の手先軌道（左）と速度形状（右）
Fig. 5　Trajectories (left) and velocity profiles (right) of an end effector without a weight.

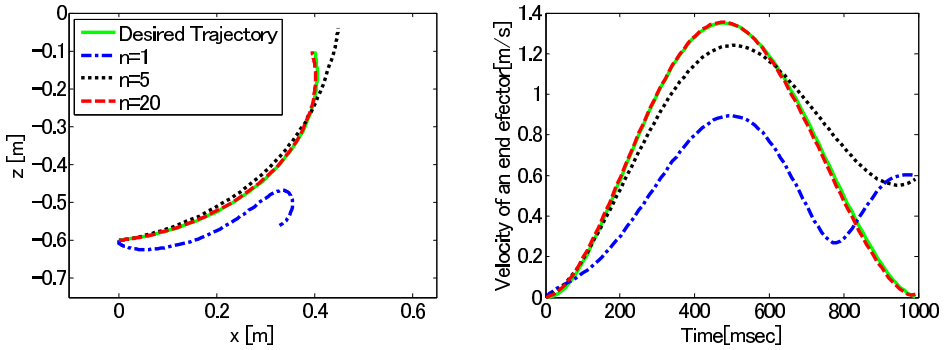

図 6　おもりを持つ場合の手先軌道（左）と速度形状（右）
Fig. 6　Trajectories (left) and velocity profiles (right) of an end effector with a weight.

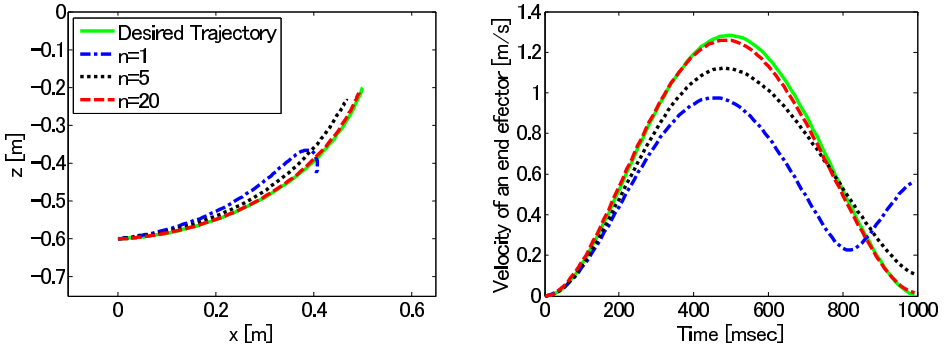

図 7　目標軌道を変更した場合の手先軌道（左）と速度形状（右）
Fig. 7　Trajectories (left) and velocity profiles (right) of an end effector when desired trajectory is changed.

を行うことで望ましい運動指令を得ることができる．目標軌道の運動終了手先位置を (0.5,−0.2) [m] に変更し，先ほど得た近似逆モデルを用い反復制御を行った結果を図 7 に示す．運動環境が変化した場合と同様に 20 回の反復制御の後，目標軌道（実線）と実現軌道（破線）が重なっており，再び望ましい運動指令を得ることができた．

また，運動環境の変化が大きい場合，獲得している逆モデルと理想的な逆モデルの間の誤差が大きくなり，収束性能が低下し，運動指令値を修正するだけでは対

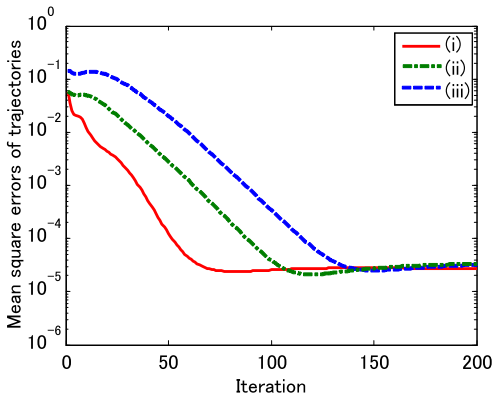

図 8　学習効果の比較
Fig. 8　Comparing the effectiveness of learning.

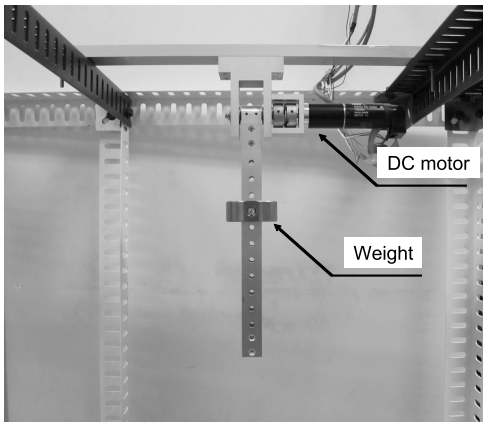

図 9　1 リンクアーム
Fig. 9　A 1-link arm.

応できないこともあると考えられる．そのような場合には，運動環境の大きな変化に対し，その変化より小さな運動環境の変化において逆モデルを再学習し対応することが有効であると考えられる．そこで，運動環境の大きな変化（おもり 10 [kg]）に対し，以下に示す 3 種類の実験を行った．

(i)　一度小さな運動環境の変化（おもり 5 [kg]）において反復制御を行い，そのときに得た運動指令を用いて逆モデルを再学習する．更に，大きな運動環境の変化（おもり 10 [kg]）に対し獲得した逆モデルを用いて反復制御を行う．

(ii)　小さな運動環境の変化（おもり 5 [kg]）において反復制御を行い，運動指令を獲得するが再学習を行わず，そのときに得た運動指令を初期出力として大きな運動環境の変化（おもり 10 [kg]）に対し反復制御を行う．

(iii)　小さな運動環境の変化（おもり 5 [kg]）に対する反復制御は行わず，はじめから大きな運動環境の変化（おもり 10 [kg]）に対して反復制御を行う．

図 8 に大きな運動環境の変化で反復制御を行ったときの手先軌道の誤差二乗和を示す．(i)，(ii)，(iii) いずれの場合にも収束したが，(i) のように再学習した後反復制御を行うと，1 回目の試行における誤差が少なく，また，反復ごとの誤差の減少が大きく，収束性が向上していることが分かる．

3.2　実 機 実 験

実験には図 9 に示す 1 リンクアームを用いた．この実験機が有する DC モータにより電流制御（トルク制御）が可能である．アームは質量 100 [g], アーム長 300 [mm] であり，おもりを取り付けることが可能である．また，アームの関節角度は一番下にあるときを 0 [rad] とした．

アームの関節角度はモータに内蔵されたエンコーダにより取得し，関節角速度，関節角加速度は関節角度より計算した．目標軌道は **3.1** と同様に関節角躍度最小軌道とした．運動時間 1.0 [s]，サンプリング周波数 100 [Hz] である．また，ダイナミックスを変化させる際には，図 9 にあるようにアームに 0.5 [kg] のおもりを取り付けた．

このアームの逆ダイナミックスモデルは，関節角度，関節角速度と関節角加速度が入力され，モータへの指令電流値が出力される．そこで，入力層，中間層，出力層にそれぞれ，3 個，10 個，1 個のニューラルネットワークを用意した．また，式 (12)，式 (14) に示す正定数とゲインは $\epsilon = 0.03$, $K_A = 15, K_D = 50$ とした．

線形 PDA 制御器に近似するよう調整したニューラルネットワークを用いて，反復制御を行った際の手先軌道を図 10 に示す．図より，20 回の反復制御を行った結果，関節角度，角速度，角加速度ともに目標値（実線）に近い実現値（破線）を示しており望ましい運動指令を求めることができたことが分かる．このとき，計算機シミュレーションと異なり 1 回目の試行ではアームが動いていない．これは実機ではモータに摩擦があることと，ニューラルネットワークからの出力がモータへの指令電流値であり，初期状態での出力が大きな値をとらず，モータのトルクが小さいためである．

次に，求めた運動指令を教師信号とし，誤差逆伝搬則を用いてニューラルネットワークの学習を行い，近

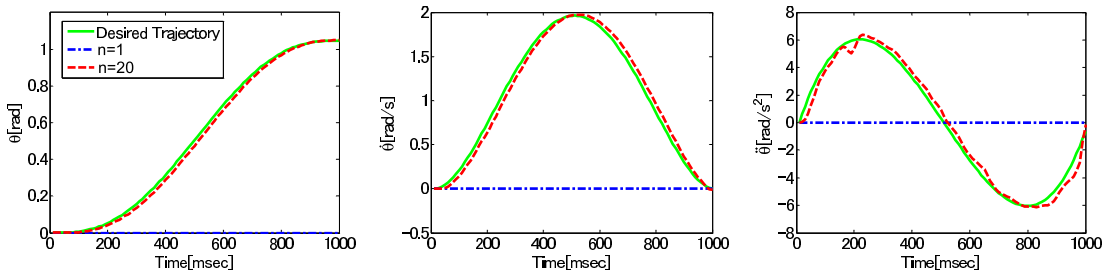

図 10　おもりを付けていない場合の関節角度，角速度，角加速度
Fig. 10　Joint angles, angular velocity and angular acceleration without a weight.

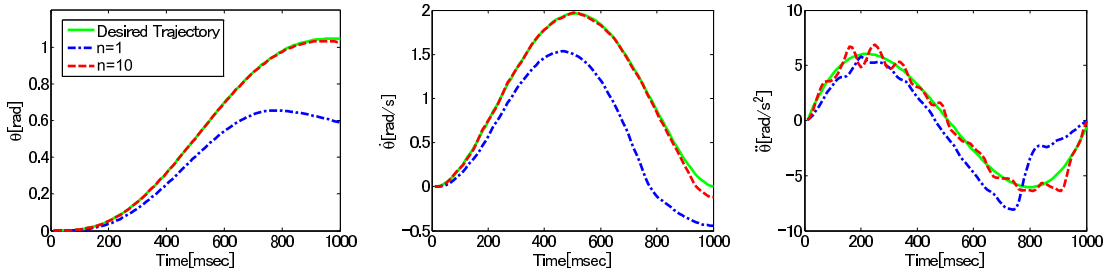

図 11　おもりを付けた場合の関節角度，角速度，角加速度
Fig. 11　Joint angles, angular velocity and angular acceleration with a weight.

似逆モデルを得た．更に，アームにおもりを付けることでダイナミックスを変化させた後に，おもりを付けないときに獲得した近似的モデルを用い，提案した反復制御を行い，再び目標軌道を追従する運動指令を得るため運動適応を行った．その結果を，図 11 に示す．10 回の反復制御により関節角度，角速度については，再び望ましい値を実現することができた．しかし，関節角加速度には振動成分が残ってしまった．これは，おもりを付けたことにより運動の際の機械振動が増加したため，その影響が現れたものと考えられる．

4. むすび

本論文では，逆ダイナミックスモデルを用いた運動指令誤差の推定を利用し，反復制御を行うことで運動適応を行う手法を提案した．また，提案手法を計算機シミュレーションと実機を用いた実験により検証し，有効性を確かめることができた．

本論文で提案した手法の特徴は，逆モデルを変更せず，運動指令を修正することにある．この手法では，式 (1) にあるように前回の運動指令値に修正量を足し合わせる簡単な形で更新ができる．一方，フィードバック誤差学習 [2] のように逆モデルを学習することで運動環境の変化に対応する場合には，ニュールネットワークの結合荷重，つまり逆モデルの関数系の変更を勾配法を用いた更新により行う必要があり，多くの試行を行わなければならない．よって，制御対象を実際に動かす回数が少ない提案手法は効率的であると思われる．また，反復制御において Arimoto ら [5] の方法では，運動指令の修正量を求めるためにゲイン行列の探索を行う必要があるが，本研究の方法では内部モデルの情報を用いて運動指令の修正量を求めることができる．また，逆モデルをニューラルネットワークで表現するので，対象の正確な物理パラメータ値を必要とせずに，制御を行えるという特徴ももっている．

本論文の計算スキームを実行する上で問題となるのは，運動指令の誤差を推定する上で，どの程度正確な逆ダイナミックスモデルが存在しなければならないか分かっていないことにある．よって，どの程度の近似逆モデルが誤差を収束させるのに必要か検討しなければならない．また，目標軌道に近い軌道を追従する運動指令を出力するような逆ダイナミックスモデルは必要ではあるが，反復制御では逆ダイナミックスモデルが関数空間において軌道に対する運動指令の勾配をいかによく近似できるかがより重要であると考えられる．そのために，どのような構造のニューラルネットワークで逆ダイナミックスモデルを学習すべきであるかが

今後の研究課題として挙げられる．

また，本論文ではニューラルネットワークの学習に誤差逆伝搬学習則を用いたが，この学習法は生理学的に発見されていない神経回路の逆伝搬を用いる．しかし，提案手法ではネットワークの入力層に軌道誤差を入力し神経回路を前向きに伝搬することで出力層の誤差（運動指令誤差）の推定する．そのため，中間層への教師信号も誤差の順伝搬により獲得が可能であると考えられる．よって，そのような形でニューラルネットワークの学習を行うことが求められる．

謝辞　本研究に対し，貴重な御助言を頂きました名古屋大学の香川高弘助教に感謝致します．本研究は一部，科学研究費補助金基盤研究 (B) No.21300092 の補助を受けている．

文　　献

[1] 川人光男，脳の計算理論，産業図書，2000.
[2] M. Kawato, K. Furukawa, and S. Suzuki, “A hierarchical neural-network model for control and learning of voluntary movement,” Biol. Cybern., vol.57, pp.169–185, 1998.
[3] H. Miyamoto, M. Kawato, T. Setoyama, and R. Suzuki, “Feedback-error-learning neural network for trajectory control of a robotic manipulator,” Neural Netw., vol.1, no.3, pp.251–265, 1988.
[4] M.I. Jordan and D.E. Rumelhart, “Forward models: Supervised learning with a distal teacher,” in Advanced Neural Computers, ed. R. Eckmiller, pp.365–272, North Holland, 1990.
[5] S. Arimoto, S. Kawamura, and F. Miyazaki, “Bettering operation of robots by learning,” J. Robotic Systems, vol.1, no.2, pp.123–140, 1984.
[6] 宇野洋二，川西康之，菅田　誠，鈴木良次，“多関節腕の繰り返し制御と逆ダイナミクスモデルの学習，” 信学論（D-Ⅱ），vol.J76-D-Ⅱ, no.1, pp.140–148, Jan. 1993.
[7] 永澤和行，福村直博，宇野洋二，“多層神経回路が逆モデルを学習するための Forward-Propagation 則，” 信学論（D-Ⅱ），vol.J85-D-Ⅱ, no.6, pp.1066–1074, June 2002.
[8] 大濱吉紘，福村直博，宇野洋二，“簡素化された Forward-propagation 則と正則化は効率的な学習を実現する，” 日本神経回路学会第 13 回全国大会講演論文集，pp.134–135, 2003.
[9] D.E. Rumelhart, J.L. McCell, and PDP Research Group, Parallel Distributed Processing vol.1, MIT Press, 1987.

（平成 23 年 3 月 28 日受付，8 月 5 日再受付）

大谷　将司

平 21 名大・工卒．平 23 同大大学院工学研究科修士課程了．同年 4 月（株）豊田自動織機入社，現在に至る．大学では人間の運動制御に関するに研究に従事．IEEE 会員．

田地　宏一

平 5 京都大学大学院・工・博士・中退．同年奈良先端大・情報助手，平 10 阪大・基礎工・講師，平 15 同助教授，平 16 名大・工・助教授，平 19 同准教授となり現在に至る．最適化アルゴリズムの理論及びシステム論などへの応用に関する研究に従事．京都大学博士（工学）．日本オペレーションズ・リサーチ学会，計測自動制御学会，SIAM 等各会員．

宇野　洋二　（正員）

昭 63 大阪大学大学院基礎工学研究科博士課程了．工博．同年東大・工・計数助手．平 3 同講師．平 4 ATR 人間情報通信研究所主任研究員．平 8 豊橋技科大情報工学系教授．同 18 名大・工・機械教授．現在に至る．生体の運動機能の研究に従事．日本神経回路学会，計測自動制御学会各会員．