

逆スタティクス・ダイナミクスモデルを用いた多関節腕の運動適応

大谷 将司[†] 田地 宏一[†] 宇野 洋二[†][†] 名古屋大学大学院工学研究科 〒464-8603 愛知県名古屋市千種区不老町

E-mail: †{m.otani,taji,uno}@nuem.nagoya-u.ac.jp

あらまし これまで、生体の運動制御を意識した運動制御スキームとして、制御対象の逆モデルを利用した運動適応手法を提案してきた。このスキームは軌道誤差と逆モデルの勾配情報に基づいて、ニュートン法により運動指令を繰り返し修正し、目標軌道を追従する運動指令を得る方法である。本論文では、これを逆モデルの静的なモジュールである逆スタティクスモデル (ISM) と動的なモジュールである逆ダイナミクスモデル (IDM) に分けたシステムへ拡張する。提案する手法では、ISM を事前に獲得することで重力補償が行われ、運動環境のより大きな変化に対し運動適応できることが期待される。2リンクアームの運動制御について、計算機シミュレーションと実機実験により検証を行いこの手法の有効性を確認した。

キーワード 運動適応, 反復制御, 逆スタティクスモデル, 逆ダイナミクスモデル, ニューラルネットワーク

Motion Adaptation of a Multijoint Arm using Inverse Statics and Dynamics Models

Masashi OTANI[†], Kouichi TAJI[†], and Yoji UNO[†][†] Graduated School of Engineering, Nagoya University, Furo-cho, Chikusa-ku, Nagoya, 464-8603 Japan

E-mail: †{m.otani,taji,uno}@nuem.nagoya-u.ac.jp

Abstract We have proposed a motion adaptation scheme using an inverse model of a controlled object based on human motion control. This scheme iteratively corrects the motor command by a Newton-like method based on the gradient of an inverse model and trajectory errors, and gives the motor command for a desired trajectory. In the current study, we extend this iterative control scheme to a modular system that consists of an inverse dynamics model (IDM) and an inverse statics model (ISM). In the proposal scheme, gravity factor is compensated by an acquired ISM a priori, and the motion adaptation is expected to cope with various changes of a movement environment. The effectiveness of the scheme is confirmed by computational simulation and actual experiments of a 2-link arm.

Key words Motion adaptation, Iterative control, Inverse statics model, Inverse dynamics model, Neural network

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」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クスと逆向きになっており、逆モデルと呼ばれている。

運動環境の変化に伴い、制御対象のダイナミ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。これに対応する手法の一つと

して、逆モデルを運動環境にあった形に変更する「学習」と呼ばれる制御スキームがある[1]。しかし、運動環境の変化の度に逆モデルを変更するのは、運動環境を元に戻したときに再び逆モデルを変更する必要があるなど効率が悪い。一方、逆モデルを変更せず逆モデルから出力された運動指令を修正することで運動環境の変化に対応する「適応」と呼ばれる制御スキームが考えられる[2]。

我々は適応に注目し、反復制御法に基づいた運動指令の修正法を提案してきた[3]。提案した手法では、宇野ら[4]が提案した軌道誤差と逆モデルの勾配情報に基づいて、ニュートン法により運動指令の修正量を求める方法を用いる。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。これはまた、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を持っている。しかし、運動指令の更新にニュートン法を用いているため、運動環境の変化が大きくなると、反復制御の初期軌道が不適当となり発散するなど収束能力が落ちる問題があった。

そこで本研究では、この反復制御法を片山ら [5] が提案した並列階層制御神経回路モデルに拡張する。このモデルは軌道制御と力制御を統一的に説明するために提案され、逆モデルとして力制御を担う静的な逆スタティクスモデル (ISM) と、軌道制御を担う動的な逆ダイナミクスモデル (IDM) に分けて、並列階層化したモデルである。本研究で提案する手法では、このモデルをベースにして、まず運動環境の変化に対応して ISM を学習し、その後適応を行う。これは、人間が物を持った際の静的な姿勢制御を比較的簡単に行えることから、ISM は簡単なキャリブレーションにより獲得できると考えるからである。そして、ISM が重力補償を行うことで、精度の良い IDM を獲得でき、運動環境のより大きな変化に対して適応できる可能性がある。

2 節で、逆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 節で、ISM と IDM を用いた反復制御アルゴリズムを提案する。4 節で、2 リンクアームに対する計算機シミュレーションと実機実験の結果を示す。最後に、5 節で提案手法の有効性と問題点について論ずる。

2. 逆モデルを用いた反復制御

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 $x_d(t)$ が与えられたとき、1 回目の試行では、逆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 $\tau^n(t)$ に対する軌道誤差信号 $\delta x^n(t) = x_d(t) - x^n(t)$ は、逆モデルを利用した誤差変換器によって、運動指令の修正量 $\delta \tau^n(t)$ へと変換される。この誤差変換の詳細については 2.1 節で説明する。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。

$$\tau^{n+1} = \tau^n + \delta \tau^n \quad (1)$$

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、逆モデルの精度を上げることもできる。

2.1 運動指令誤差の推定

逆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

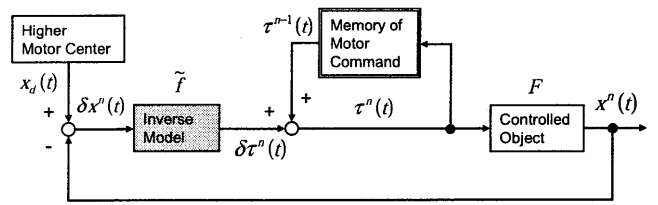


図 1 反復制御スキーム

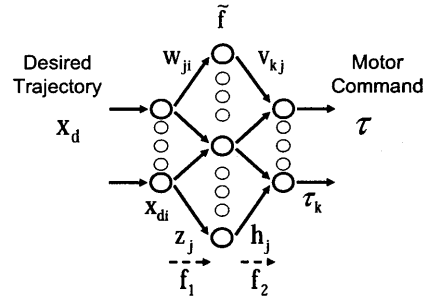


図 2 逆モデルの入出力関係

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。本論文では逆モデルを図 2 に示す 3 層のニューラルネットワークで表現する。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。

$$z_j = \sum_i w_{ji} x_{di} \quad (2)$$

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

$$\tau_k = \sum_j v_{kj} h_j \quad (4)$$

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 $z = f_1(x)$ 、中間層入力から出力層への変換を $\tau = f_2(z)$ 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

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

が成り立つとみなせる。

目標軌道 x_d と実現軌道 x_r の誤差 $\delta x = x_d - x_r$ を小さくするために、ニュートン法を用いて中間層に対する目標信号を逐次的に更新するならば

$$\hat{z} = \left[\frac{\partial \{F \circ f_2(z)\}}{\partial z} \right]^{-1} (x_d - x_r) + z \quad (6)$$

となる。このとき、 $(F \circ f_2)^{-1}$ の近似式として f_1 を利用することができるので、式 (6) のかわりに、次式の広義ニュートン法

$$\hat{z} \simeq \left[\frac{\partial f_1(x)}{\partial x} \right] (x_d - x_r) + z \quad (7)$$

が利用できる。これにより、中間層に対する修正量 δz は次のようにして求められる。

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

$$\begin{aligned} \delta z_j &= \sum_i \frac{\partial f_1(x_{di})}{\partial x_{di}} \delta x_i = \sum_i \frac{\partial (w_{ji} x_{di})}{\partial x_{di}} \delta x_i \\ &= \sum_i w_{ji} \delta x_i \end{aligned} \quad (9)$$

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、制御対象から中間層入力までの変換 $(f_1 \circ F)^{-1}$ の近似式として f_2 を利用し、中間層入力に対する修正量 δz を前向きに伝播させることにより、出力層に対する修正量 τ_k の導出が可能である。

$$\begin{aligned} \hat{\tau} &= \left[\frac{\partial \{f_1 \circ F(\tau)\}}{\partial \tau} \right]^{-1} \delta z + \tau \\ &\simeq \left\{ \frac{\partial f_2(z)}{\partial z} \right\} \delta z + \tau \end{aligned} \quad (10)$$

$$\begin{aligned} \delta \tau_k &= \sum_j \left\{ \frac{\partial f_2(z_j)}{\partial z_j} \right\} \delta z_j = \sum_j \left\{ \frac{\partial \{v_{kj} h_j(z_j)\}}{\partial z_j} \right\} \delta z_j \\ &= \sum_j \{h_j(1-h_j)v_{kj}\} \delta z_j \end{aligned} \quad (11)$$

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)を用いて式(11)は

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

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、経験的により良い推定を促すことが確認されている [6]。

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。

$$\delta \tau = \frac{\partial f_2(z)}{\partial z} \cdot \frac{\partial f_1(x)}{\partial x} \delta x = \frac{\partial \tilde{f}(x)}{\partial x} \delta x \quad (13)$$

以上の式により、軌道誤差 δx から運動指令の修正量 $\delta \tau$ を求めることができる。

3. ISM と IDM を用いた反復制御

制御対象のダイナミクスは慣性項、コリオリ項、粘性項、重力項を用いて次式のように表されるものとする。

$$\tau = M(\theta)\ddot{\theta} + C(\theta, \dot{\theta}) + D(\dot{\theta}) + G(\theta) \quad (14)$$

ただし、 θ は位置または角度を表す変数である。右辺の前3項の慣性項、コリオリ項、粘性項は制御対象の動的な要素を含んでいるので IDM としてまとめ f_D とし、第4項の重力項は静的な要素のみで構成されているので ISM として f_S とすると、式(14)は次のように表すことができる。

$$\tau = f(\theta, \dot{\theta}, \ddot{\theta}) = f_D(\theta, \dot{\theta}, \ddot{\theta}) + f_S(\theta) \quad (15)$$

$$f_D = M(\theta)\ddot{\theta} + C(\theta, \dot{\theta}) + D(\dot{\theta}) \quad (16)$$

$$f_S = G(\theta) \quad (17)$$

また、運動指令の修正量は式(15)の変分を取ることより求め

ることができる。

$$\delta \tau = \frac{\partial f}{\partial \theta} \delta \theta + \frac{\partial f}{\partial \dot{\theta}} \delta \dot{\theta} + \frac{\partial f}{\partial \ddot{\theta}} \delta \ddot{\theta} \quad (18)$$

f_S の近似逆モデルを \tilde{f}_S 、 f_D の近似逆モデルを \tilde{f}_D として、これを用いた反復制御系を図3に表す。これは図1の逆モデルを IDM と ISM に分けて並列化したものになり、2節の反復制御と同様な計算で運動指令を修正していく。

目標軌道 θ_d が与えられたとき、1回目の試行では、ISM と IDM の近似逆変換の和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 $n+1$ 回目の試行は反復制御と同様に式(1)の更新式を用いて行われる。この時、 n 回目の修正量は式(15)、式(18)より以下のように求められる。

$$\delta \tau^n = \frac{\partial \tilde{f}_D}{\partial \theta} \delta \theta^n + \frac{\partial \tilde{f}_D}{\partial \dot{\theta}} \delta \dot{\theta}^n + \frac{\partial \tilde{f}_D}{\partial \ddot{\theta}} \delta \ddot{\theta}^n + \frac{\partial \tilde{f}_S}{\partial \theta} \delta \theta^n \quad (19)$$

ただし、 $\frac{\partial \tilde{f}_D}{\partial \theta}$ 、 $\frac{\partial \tilde{f}_D}{\partial \dot{\theta}}$ 、 $\frac{\partial \tilde{f}_D}{\partial \ddot{\theta}}$ 、 $\frac{\partial \tilde{f}_S}{\partial \theta}$ の値は、繰り返しの番号 n に関わらず、目標軌道 $\theta = \theta_d$ 、 $\dot{\theta} = \dot{\theta}_d$ 、 $\ddot{\theta} = \ddot{\theta}_d$ に対して計算された値を用いる。さらに、 $x = (\theta, \dot{\theta}, \ddot{\theta})^T$ とおくと式(19)は次の形に書き直される。

$$\delta \tau^n = \delta \tau_D^n + \delta \tau_S^n = \frac{\partial \tilde{f}_D}{\partial x_d} \delta x^n + \frac{\partial \tilde{f}_S}{\partial \theta_d} \delta \theta^n \quad (20)$$

ISM, IDM をそれぞれニューラルネットワークを用いて作成しておくことで、この計算は3節で示した誤差推定法を用いて行うことができる。ただし、本論文では、ISM が IDM に比べて簡単に作れるので、 $\delta \tau_s$ は4.1.2節で述べる推定法を用いる。

4. 実験

提案した手法の有効性を確認するため、鉛直面内を動く2リンクアームの運動制御について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。実験では近似逆モデルを獲得した後に、近似逆モデルを利用した反復制御により運動適応を行い有効性を確認する。また、ISM と IDM を分けない従来法 [3] と比較する。

(a) 逆モデルの獲得

本論文で用いるスキームでは、あらかじめ、制御対象の近似逆モデルが必要である。そこで、まず、適当な振幅と位相をもつ駆動トルクの時間波形を5種類用意してアームを動かす、それぞれに対応する関節角軌道を求めた後、これらの5種類の関節角軌道と駆動トルクの対を訓練パターンとして、誤差逆伝播則によりネットワークの学習を行う（以後、事前学習と呼ぶ）。ただし、ISM は先に学習しておき、事前学習は IDM のみに行う。この ISM の学習法は後ほど、4.1.2節に示す。

さらに、この近似逆モデルを用いて反復制御を行い、目標軌道に対して望ましい運動指令を求め再び IDM を学習することで、逆モデルを更新する。

(b) 運動環境の変化と ISM の学習

おもりを付けることで、制御対象のダイナミクスを変化させる。また、新しい運動環境に対して ISM を新たに学習する。

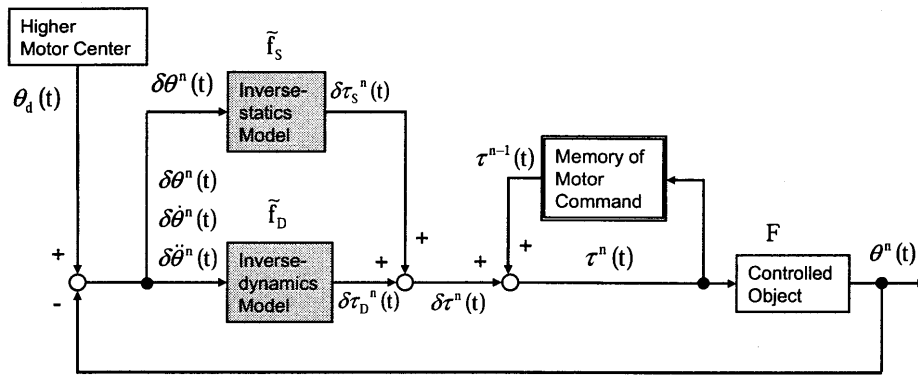


図3 ISM と IDM を用いた反復制御スキーム

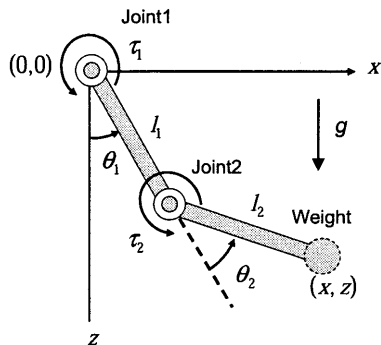


図4 鉛直面内を動く 2 リンクアーム

(c) 反復制御による運動適応

制御対象のダイナミクスの変化に対して、(a) で得た IDM と (b) で得た ISM を利用して、反復制御を行うことで変化した運動環境において目標軌道を追従する運動指令を獲得し、運動適応を行う。本論文では、この運動適応の結果を示す。

4.1 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) について計算機シミュレーションを行う。

4.1.1 実験条件

2 リンクアームのダイナミクスは次式で表される。

$$\tau = M(\theta)\ddot{\theta} + C(\theta, \dot{\theta}) + D(\dot{\theta}) + G(\theta) \quad (21)$$

$$\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}$$

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}$$

$$C = \begin{bmatrix} -m_2 l_1 s_2 \dot{\theta}_2 (2\dot{\theta}_1 + \dot{\theta}_2) \sin \theta_2 \\ m_2 l_1 s_2 \dot{\theta}_1^2 \sin \theta_2 \end{bmatrix}$$

$$D = \begin{bmatrix} b_1 \dot{\theta}_1 \\ b_2 \dot{\theta}_2 \end{bmatrix}$$

$$G = \begin{bmatrix} m_1 g s_1 \sin \theta_1 + m_2 g \{l_1 \sin \theta_1 + s_2 \sin(\theta_1 + \theta_2)\} \\ m_2 g s_2 \sin(\theta_1 + \theta_2) \end{bmatrix}$$

$$M_{11} = m_1 s_1^2 + I_1 + m_2 (l_1^2 + s_2^2 + 2l_1 s_2 \sin \theta_2) + I_2$$

$$M_{12} = M_{21} = m_2 (s_2^2 + l_1 s_2 \sin \theta_2) + I_2$$

表 1 2 リンクアームのパラメータ

	$m[\text{kg}]$	$l[\text{m}]$	$s[\text{m}]$	$I[\text{kg} \cdot \text{m}^2]$	$b[\text{kg} \cdot \text{m}/\text{s}]$	$g[\text{m}/\text{s}^2]$
Link1	1.59	0.3	0.0992	0.0477	0.4	9.81
Link2	1.44	0.35	0.1223	0.162	0.4	

$$M_{22} = m_2 s_2^2 + I_2$$

ここで、リンクの長さ l 、質量中心までの長さ s 、質量 m 、関節周りの慣性モーメント I 、粘性抵抗 b 、重力加速度 g であり、各パラメータの値を表 1 のように設定した。このアームの IDM 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_{D1}, τ_{D2} 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、ISM は角度 θ_1, θ_2 が入力され、関節トルク τ_{S1}, τ_{S2} が出力される。アームの運動軌道は 4 次の Runge-Kutta 法を用いて計算する。目標軌道は図 5 の実線に示す各関節の角加速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 [sec]、サンプリング周波数 100 [Hz] である。また、運動環境を変化させる際には、アームの手先におもりを持たせた。

4.1.2 ISM の学習

アームが静止しているとき、関節角速度、角加速度はゼロとなる。よって、各関節角と関節トルクの関係は式 (21) より以下の式で表すことができる。

$$\tau_{S1} = m_1 g s_1 \sin \theta_1 + m_2 g \{l_1 \sin \theta_1 + s_2 \sin(\theta_1 + \theta_2)\} \quad (22)$$

$$\tau_{S2} = m_2 g s_2 \sin(\theta_1 + \theta_2) \quad (23)$$

ここで、パラメータ a, b を以下のようにおくと、

$$a = m_1 g s_2 + m_2 g l_1 \quad (24)$$

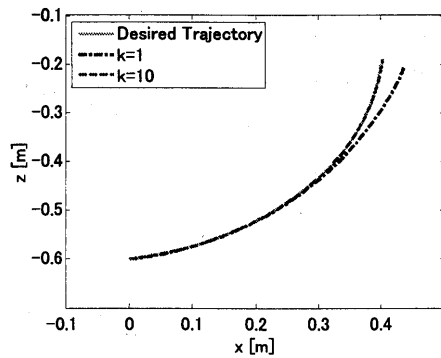
$$b = m_2 g s_2 \quad (25)$$

式 (22)、(23) は次の式で書き換えることができる。

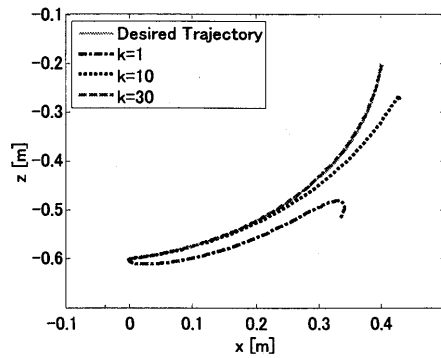
$$\tau_{S1} = a \sin \theta_1 + b \sin(\theta_1 + \theta_2) \quad (26)$$

$$\tau_{S2} = b \sin(\theta_1 + \theta_2) \quad (27)$$

また、式 (26)、式 (27) のヤコビアンを求めることで ISM によ



(a) 提案法



(b) 従来法

図5 計算機シミュレーションの反復制御による2リンクアームの手先軌道.

る運動指令の修正量は以下のように計算できる.

$$\delta\tau_S = \begin{pmatrix} a \cos \theta_1 + b \cos(\theta_1 + \theta_2) & b \cos(\theta_1 + \theta_2) \\ b \cos(\theta_1 + \theta_2) & b \cos(\theta_1 + \theta_2) \end{pmatrix} \begin{pmatrix} \delta\theta_1 \\ \delta\theta_2 \end{pmatrix} \quad (28)$$

各関節の角度とトルクは得ることができるので、パラメータ a, b の値は2リンクアームのいくつかの姿勢データから最小二乗法より推定する.

4.1.3 実験結果と考察

逆モデルを獲得した後、アームの手先に0.5 [kg]のおもりを持たせ運動環境を変化させ、図3のスキームに基づく運動適応を行った。その時の手先軌道の結果を図5(a)に示す。ISMは新たな運動環境に対して学習するがIDMは変更しないので、逆モデルから出力される運動指令は1回目の試行(一点鎖線)では目標軌道を追従できない。そこで、反復制御を行った結果、10回目の試行(破線)により目標軌道を追従することができた。比較として行ったISMとIDMを分けない従来法(図1)による反復制御の結果を図5(b)に示す。従来法では1回目の試行(一点鎖線)から軌道誤差が大きく、10回目の試行(点線)では目標軌道を追従できず、収束に30回の試行(破線)を必要とした。

また、手先につけるおもりを(0.5 [kg], 2 [kg], 5 [kg])と変え、ダイナミクスの変化に対し運動適応を行い、その時の軌道誤差を比較する。この結果を図6に示す。図の細い線が従来法、太い線が本論文で提案した手法を用いた場合の誤差を示す。提案法は従来法と比べ、2つの点で優位であると言える。第1

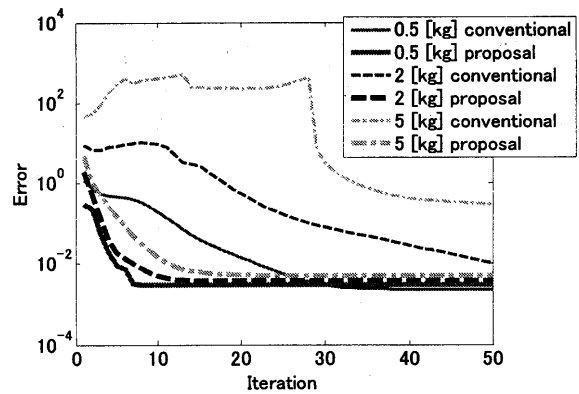


図6 軌道誤差の収束の比較

点は1回目の試行における誤差が少ないことが挙げられる。これは、おもりの質量が大きくなるにつれ顕著になっている。本論文で用いる運動指令の更新にニュートン法を用いるため初期解が近いことは、収束性の向上を意味する。第2点は収束速度の速さが挙げられる。これは逆モデルをISMとIDMに分け、さらにISMは運動環境の変化に対し新たに獲得しているので、逆モデルの誤差が小さいことが要因であると考えられる。以上のことから、提案法は従来法に比べ運動環境の変化に対する適応能力において優れていると言える。

4.2 実機実験

図7に示す2リンクアームの実機を用いて実験を行う。

4.2.1 実験条件

実験機はECモータにより電流制御(トルク制御)が可能であり、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時を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4.1節と同様に関節角躍度最小軌道とした。運動時間1.0 [sec]、サンプリング周波数100 [Hz]である。また、ダイナミクスを変化させる際には、図7にあるように0.5 [kg]のおもりを取り付けた。

このアームのIDMは、各関節の関節角度、角速度と角加速

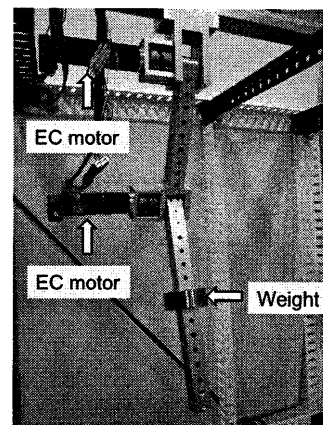


図7 2リンクアームの実機

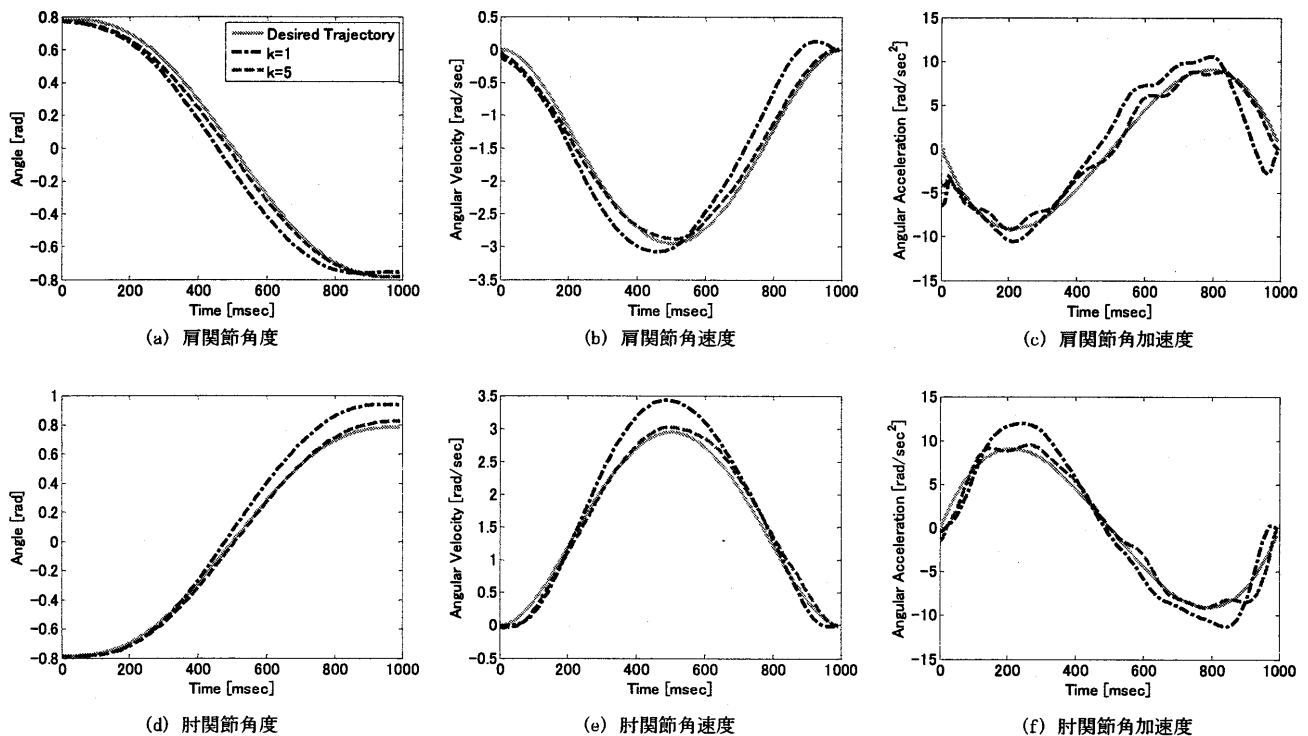


図 8 実機実験の反復制御による 2 リンクアームの関節角度、角速度、角加速度。

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、出力層がそれぞれ、6 個、15 個、2 個のニューラルネットワークを用意した。また、ISM は各関節の関節角度が入力され、モータへの指令電流値が出力される。

4.2.2 ISM の学習

ISM の学習には 4.1.2 節と同様の方法を用いる。ただし、シミュレーションでは ISM からの出力は関節トルクであったが、実機実験では指令電流値になる。

また、ISM の推定にはいくつかの姿勢での関節角度と指令電流値が必要であるが、実験に用いたアームでは静止状態では静止摩擦の影響が大きく出る。そこで、アームを 90 [deg/min] と低速度で動かしたときの値を用いた。これにより、静止摩擦の影響をなくし慣性項、粘性項、コリオリ項の影響を少なくした。

4.2.3 実験結果と考察

逆モデルを獲得した後、アームの手先に 0.5 [kg] のおもりを持たせた運動環境を変化させ、反復制御を行った際の各関節の角度、角速度、角加速度を図 8 に示す。運動環境の変化に対して、1 回目の試行（一点鎖線）から目標軌道（実線）に近い実現軌道を描くことができている。反復制御により 5 回の試行（破線）で目標軌道に近い実現軌道を描く運動指令が得られた。これにより、提案手法が実際の制御対象に対しても有効であると言える。

5. むすび

本論文では、逆スタティクス・ダイナミクスモデルを用いた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実験により検証し、

有効性を確かめることができた。また、従来法と比較することで、運動環境の変化に対して収束性が改善されることが示せた。

本論文では ISM が正しい形で学習されていることを前提とし、IDM の学習の際に反復制御により獲得した運動指令から ISM の出力を除くことで IDM の出力を求めた。しかし、ISM が完全でない場合、反復制御より得た運動指令からいかにして ISM と IDM にそれぞれの出力を求めるか検討する必要がある。

謝 辞

本研究に対し、貴重なご助言をいただきました名古屋大学の香川高弘助教に感謝いたします。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] Kawato M., Furukawa K. and Suzuki S., "A hierarchical neural-network model for control and learning of voluntary movement", *Biol.Cybern.*,57,pp.169-185,1998.
- [2] Suguru Arimoto, Sadao Kawamura, Fumio Miyazaki, "Bettering Operation of Robots by Learning", *Journal of Robotic Systems*, 1(2), 123-140, 1984.
- [3] 大谷将司, 田地宏一, 宇野洋二, "逆ダイナミクスモデルを用いた運動指令誤差の推定と運動適応", NC2010-86, pp.115-120, 2010.
- [4] 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", *電子情報通信学会論文誌 (D-II)*, J76-DII :140-148, 1993.
- [5] 片山正純, 川人光男, "筋肉骨格系の運動制御を行う並列階層制御神経回路モデル", *電子情報通信学会論文誌 (D-II)*, J73-DI I, pp.1328-1335, 1990.
- [6] 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*, 134-135, 2003.