

# 複数のモデルの学習と切り替えを行う競合連想ネットを用いる適応予測制御†

黒木 秀一\*・西田 健\*

## Adaptive and Predictive Control Using Competitive Associative Net for Learning and Switching of Multiple Models†

Shuichi KUROGI\* and Takeshi NISHIDA\*

So far, the controllers using the switching scheme of multiple models have been proposed for controlling the plants whose parameter values change occasionally and largely owing to wide varieties of environments involving different operating conditions, uncertainty, and so forth. Although the stability of the controllers has been analyzed, the efficiency of the models has not been examined in detail, where the present learning algorithm of competitive associative net called CAN2 is formalized and utilized to minimize the energy defined as the sum of identification errors of switched models so that the models can be efficiently utilized when the identification errors are large. In order for the CAN2 to learn and switch multiple models for reproducing the forward dynamics of the plant, two kinds of CAN2, called CAN2-1 and CAN2-2, are introduced. The former basically is for learning to achieve a piecewise linear approximation of nonlinear plant dynamics, and the latter basically is for learning to achieve adaptive linear approximation of time-varying plant dynamics. The CAN2-1 and CAN2-2 are embodied into the conventional model-based predictive controllers, which are called APC1 (adaptive predictive controller 1) and APC2, respectively. Computer simulation clarifies the behavior of the APC1 and APC2 and shows that the APC1 is useful for nonconvex nonlinear plants while the APC2 is for time-varying plants.

**Key Words:** adaptive predictive control, competitive associative net, multiple models, time-varying plants, nonlinear plants

### 1. はじめに

複数の同定モデルを切り替えて使用する適応制御（以下ではモデル切り替え型適応制御という）<sup>1)~5)</sup>は環境や動作条件の変化などによりパラメタ値が大きく変化するプラントを制御することを目的として提案され研究されている。この制御法はある期間ごとに最小の同定誤差指標を持つモデルを複数のモデルの中から選択し、その選択したモデルを用いて制御を行うものである。この手法の安定性は固定モデル<sup>1)</sup>および適応モデル<sup>2)</sup>を用いる場合について検討されているが、以下に示すような非効率モデルの問題はほとんど検討されていない。

すなわち、まず、複数モデルの個数は理論的には Morse<sup>1)</sup>が仮定するように固定モデルならば無限個あるとよいが、実際には有限個しか使用できないので、Kangら<sup>3)</sup>はクレーンケーブルの制御においてケーブルの長さ  $l = 0.4\text{m}$   $0.7\text{m}$ ,

1.0m, 1.3m に対する 4 個の固定モデルを用い、その中に存在しない長さ  $l = 0.85\text{m}$  の場合でもうまく制御できることを示している。しかし一般に制御性能は制御対象に大きく依存し、Narendra と Balakrishnan<sup>2)</sup>の例では固定および適応モデルの組合せとして、i) 複数の適応モデル、ii) 複数の固定モデル、iii) 複数の固定モデルと一個の適応モデル、iv) 複数の固定モデルと一個の適応モデルおよび一個の再初期化適応モデル、を用いた比較実験の結果、iv) が良いことを示している。ここで再初期化適応モデル ( $M_A$ とする)とはある固定モデル ( $M_j$ とする)がある時刻において選択されると、その時刻において  $M_j$ のパラメタ値に  $M_A$ のパラメタ値が再設定される適応モデルである。この結果はこのモデル切り替え戦略における非効率モデルの問題を示唆している。すなわち評価指標最小を基準とした切り替え機構を用いると、最小の同定誤差指標を持つことのないモデル、たとえばプラントの真のパラメタ値と非常に異なるパラメタ値をもつ固定モデルや変更されて他のモデルとほぼ同じパラメタ値をもつようになった適応モデルなどはそのプラントの制御にまったくあるいはほとんど使用されない非効率なモデルとなる。もしこのような非効率なモデルが数多く存在すると複数のモデルを用いる利点が消失するため、非効率モデルの問題はこの制御手法の

† International Symposium on Nonlinear Theory and its Applications で一部発表 (1999・11)

\* 九州工業大学 工学部 北九州市戸畑区

\* Faculty of Engineering, Kyushu Institute of Technology

(Received April 20, 2000)

(Revised September 11, 2000)

重要な問題のひとつであると言える。

一方、人工ニューラルネットの見地から、この問題は WTA (winner-take-all) ネットの利用不足セル (under-utilized cells)<sup>6),7)</sup> の問題に関係している。ここで WTA ネットとは複数のセルの内部電位を指標とする競争 (競合) により最大または最小の内部電位をもつセルを選択し、選択したセルのみに依存してネットの出力を決定する機構を持つネットである。この機構に起因する利用不足セルの問題はエネルギー関数あるいは歪関数と呼ばれる非負関数を最小化する学習アルゴリズムの問題として解析されているが<sup>6),7)</sup>、この解析とは別に CAN2 (Competitive Associative Nets 2) と呼ばれる WTA ネットで同定誤差が大きくなった時に利用不足セルを再初期化する学習アルゴリズムが提案され、非線形プラントの内部モデル予測制御に適用されている<sup>8)~11)</sup>。それらの研究では、CAN2 は非線形プラントの力学あるいは逆力学を学習し区分的線形近似を行うもの<sup>8)~10)</sup> とパラメタが大きく変動するプラントに適用するもの<sup>11)</sup> があるがその解析は十分ではなかった。

そこで本論文では、まず 2 で、CAN2 とそのエネルギー関数を定義するとともにその学習アルゴリズムを示す。3 では非線形プラントおよび線形パラメタが切り替わる時変プラントをそれぞれ制御するために有効な 2 種類のモデル切り替え型適応制御法を提案する。4 ではシミュレーション実験結果を用いて、提案した 2 種類のモデル切り替え型適応制御法の有効性を示す。

## 2. 競合連想ネットワーク 2 (CAN2)

以下、競合連想ネットワーク 2 (CAN2)<sup>8)~10)</sup> をモデル切り替え型適応制御に応用する立場から定式化する。

### 2.1 プラント

本研究ではプラントが NARX (Nonlinear AutoRegressive with an eXogenous input) モデル<sup>19)</sup> で表され、そのパラメタは一定または緩やかに変化するとする。すなわち離散時刻  $t = \dots, -2, -1, 0, 1, 2, \dots$  のプラントの入出力を  $u(t)$  と  $y(t)$  とするとき、

$$y(t) = f(\mathbf{x}(t)) + d(t) \quad (1)$$

となり、関数  $f$  は時間的に一定または緩やかに変化するとする。ここで  $d(t)$  は平均値がゼロの確率雑音を表し、ベクトル  $\mathbf{x}(t) \in \mathbb{R}^{n \times 1}$  は

$$\mathbf{x}(t) = (1, y(t-1), \dots, y(t-n_y), u(t-1), \dots, u(t-n_u))^T \quad (2)$$

で与えられる。ここで  $n_y$  と  $n_u$  はそれぞれ非負定数であり、 $\mathbf{x}(t)$  の次元は  $n = 1 + n_u + n_y$  となる。なおこの  $\mathbf{x}(t)$  の第 1 成分を 1 としたのは後述の (4) 式の線形近似のバイアス項を生成するためであり、通常の NARX モデルではこの項は必要ないことを注意しておく。

### 2.2 入力空間を分割するネット (CAN2-1)

以下のネットは非線形関数  $y(t) = f(\mathbf{x}(t))$  を区分的に線形

近似することを目的とするものであり、従来の CAN2<sup>8)~10)</sup> と同じものであるが、以後、CAN2-1 と言う。

まずネットは複数のセル  $C_i$  ( $i \in I = \{1, 2, \dots, |I|\}$ ) を持ち、各セル  $C_i$  はプラントの同定モデルを表現する連想行列  $M_i \in \mathbb{R}^{1 \times n}$ 、競合によりモデルを選択するためのベクトル  $\mathbf{x}_i \in \mathbb{R}^{n \times 1}$ 、およびセルの発火回数  $v_i \in \mathbb{Z}$  を記憶する。ここで  $|I|$  は集合  $I$  の要素数、 $\mathbb{R}$  は実数の集合、 $\mathbb{Z}$  は非負整数の集合を表す。なお後述するように、各セルは  $M_i$  を効率よく学習するための 2 つの行列  $\Phi_i, \Psi_i$  ( $\in \mathbb{R}^{n \times n}$ ) も記憶する。

関数値  $y(t)$  の予測は、時刻  $t-1$  において次式の最小化指標に基づく競合に勝つセルの番号

$$c = c(t) = \underset{i \in I}{\operatorname{argmin}} \|\mathbf{x}(t) - \mathbf{x}_i\|^2 \quad (3)$$

を求め、次式により行う。

$$\hat{y}(t) = M_c \mathbf{x}(t) \quad (4)$$

ここで (3) 式の競合に勝つことを発火するという。またこの競合によりベクトル  $\mathbf{x}_c$  ( $c \in I$ ) は入力  $\mathbf{x} = \mathbf{x}(t)$  が存在する空間  $\mathbb{R}^{n \times 1}$  をポロノイ領域<sup>7)</sup>

$$V_c = \left\{ \mathbf{x} \mid c = \underset{i \in I}{\operatorname{argmin}} i \|\mathbf{x} - \mathbf{x}_i\|, \mathbf{x} \in \mathbb{R}^{n \times 1} \right\}, \quad (5)$$

に分割する。したがって (3)~(5) 式はこのネットが非線形関数  $y(t) = f(\mathbf{x}(t))$  の区分的線形近似を行うことを示している。

この方法では入力  $\mathbf{x}(t)$  の値が同じでも異なる時刻では異なる関数値を出力する関数すなわち時変関数に対しては、後述する忘却率または保持率を用いて古いモデル  $M_i$  を忘却しながら新しいモデルへ更新していくなどの処理が必要になる。

### 2.3 パラメタ空間を分割するネット (CAN2-2)

以下のネットは近似すべき関数  $y(t) = f(\mathbf{x}(t))$  が時間的に一定または緩やかに変化する線形パラメタ  $M(t) \in \mathbb{R}^{1 \times n}$  を用いて  $y(t) = M(t)\mathbf{x}(t)$  で表されると仮定し、この線形パラメタ  $M(t)$  の空間を分割して記憶し使用することを目的とするものである。このネットはパラメタ空間を分割する従来の CAN2<sup>11)</sup> を単純化したものであり、本稿ではこれを CAN2-2 と呼ぶ。このネットは複数のセル  $C_i$  ( $i \in I$ ) をもち、各セル  $C_i$  は連想行列  $M_i \in \mathbb{R}^{1 \times n}$ 、発火回数  $v_i \in \mathbb{Z}$ 、および  $M_i$  を効率よく学習するための行列  $\Phi, \Psi$  ( $\in \mathbb{R}^{n \times n}$ ) を記憶する。関数値  $y(t)$  の予測は時刻  $t-1$  においてそれまでに与えられた  $N_e$  組のデータ  $\mathbf{x}(t-k)$  および  $y(t-k)$  ( $k = 1, 2, \dots, N_e$ ) を用いてその線形予測誤差を最小にする連想行列をもつセルの番号、すなわち

$$c = c(t) = \underset{i \in I}{\operatorname{argmin}} i \sum_{k=1}^{N_e} \left\| y(t-k) - M_i \mathbf{x}(t-k) \right\|^2 \quad (6)$$

を求め、(4) 式を適用することにより行う。従ってパラメタ空間  $\mathbb{R}^{1 \times n}$  は

$$V_c = \left\{ M \mid c = \operatorname{argmin}_{i \in I} \|Y - M_i X\|^2, Y = M X \right\}, \quad (7)$$

により分割される．ここで  $X = [x(t-1), \dots, x(t-N_e)]$  および  $Y = [y(t-1), \dots, y(t-N_e)]$  である．

## 2.4 CAN2のエネルギー関数

時刻  $t$  における関数  $y(t) = f(x(t))$  のセル  $C_i$  ( $i \in I$ ) による近似誤差は

$$e_i(x(t)) = y(t) - M_i x(t) \quad (8)$$

で与えられ，ネットによる近似誤差は競合に勝つセルの番号  $c$  を用いて  $e_c(x(t))$  で与えられる．そこで近似誤差の二乗  $e_c^2(x(t))$  の期待値が時間の増加とともに小さくなるようにネットの記憶データを逐次的に修正することを目的として次のエネルギー関数を考える．すなわち現時刻  $k$  までに第  $c$  セルが (3) 式または (6) 式で競合に勝った (発火した) 時刻を  $t_{cj}$  と表し， $j \in L_c \subseteq \{1, 2, \dots, k\}$  はこのセルの第  $j$  回目の発火であることを表すとすると，各セルの近似誤差の二乗を競合に勝つ時刻  $t_{cj}$  ( $j \in L_c$ ) ごとに忘却率  $\rho \in [0, \infty)$  あるいは保持率  $\eta = e^{-\rho} \in (0, 1]$  で忘却しながら現時刻まで加算した

$$E = \frac{1}{\gamma} \sum_{c \in I} \sum_{j \in L_c} e_c^2(x(t_{cj})) \eta^{|L_c| - j} \quad (9)$$

をエネルギー関数とする．ここで  $\gamma = 1 + \eta + \eta^2 + \dots + \eta^{k-1}$  である．この式はデータ数が十分多いとき，

$$E = \sum_{c \in I} \int_{V_c} e_c^2(x) p(z) dz \quad (10)$$

と表すことができる．ここで  $z \in V_c$  は (5) 式の  $x$  または (7) 式の  $M$  であり，微小領域  $dz \subset V_c$  内に  $z$  が出現する発火時刻  $t_{cj}$  の添字  $j$  の集合を  $dL_c(z)$  として

$$p(z) dz = \frac{1}{\gamma} \sum_{j \in dL_c(z)} \eta^{|L_c| - j} \quad (11)$$

と置いたものである．結局，(10) 式は保持率  $\eta$  により重み付けした密度関数  $p(z)$  を用いた  $e_c^2(x)$  の期待値を表している．

## 2.5 学習アルゴリズム

以下，学習手順の概要と具体的アルゴリズムを順に示す．

### 2.5.1 学習手順の概要

まずエネルギー  $E$  を逐次的に小さくするために，各時刻  $t$  において (3) 式または (6) 式の競合に勝つ第  $c$  セルに対して次の2つの記憶更新処理を施す．

(i)  $M_c$  の更新：(9) 式の第  $c$  セルに対応する項

$$E_c = \sum_{j \in L_c} \|y(t_{cj}) - M_c x(t_{cj})\|^2 \eta^{|L_c| - j} \quad (12)$$

に再帰的最小二乗法<sup>12)</sup>を適用し  $M_c$  を更新する．

(ii) 領域  $V_c$  の更新：CAN2-1 では (10) 式に  $x_c$  に関する勾配法を適用して  $V_c(x_c)$  を更新する．CAN2-2 では  $V_c$  が  $M_c$  により決定されるので，上の処理 (i) により  $V_c$  が更新される．

以上の記憶更新は基本的にエネルギー関数の勾配を用いる手法

であり，一般に最小解を求めることはできない．特に，これらは競合学習法すなわち競合に勝つ第  $c$  セルだけに対して適用される学習法であり，競合に勝つことのないセルの更新が困難となる．そこで次の手順を追加する．

(iii) 非効率モデルの再初期化：その時刻の近似誤差が大ききときは発火回数  $v_i$  が最小のセルを探し，そのセルが有効利用できるように再初期化する．

ここでセルの発火回数は (iii) で用いると同時に (ii) でも  $x_c$  の更新に使用し，この更新は他の記憶データの更新後に

$$v_c := v_c + 1 \quad (13)$$

により行う．ここで記号 := は右辺で計算した値を左辺の記憶変数に代入することを表す．また発火回数の初期値は  $v_i = 0$  ( $i \in I$ ) である．

以下，処理 (i)，(ii)，(iii) の具体的手続きを述べた後，データの初期化を含めた全体の学習アルゴリズムを示す．

### 2.5.2 再帰的最小二乗法による $M_c$ の更新

領域  $V_c$  が一定と仮定し，(12) 式を最小二乗法により最小化するための再帰的学習アルゴリズム<sup>12)</sup>

$$M_c := M_c + (y - M_c x) p^T \quad (14)$$

を適用する．ここで  $y = y(t)$  および  $x = x(t)$  を表す．また

$$p = \begin{cases} (x^T \Phi_c^T \Phi_c x)^{-1} \Phi_c x & \text{if } \|\Phi_c x\| > \theta_0 \\ (\eta + x^T \Psi_c x)^{-1} \Psi_c x & \text{otherwise} \end{cases} \quad (15)$$

であり， $\theta_0$  はある小さな正定数である．行列  $\Phi_c$  および  $\Psi_c$  は指数荷重形最小二乗法<sup>13)</sup>により次のように更新する．すなわち  $\|\Phi_c x\| > \theta_0$  のとき

$$\Phi_c := \Phi_c - \Phi_c x p^T \quad (16)$$

$$\Psi_c := \eta^{-1} (\Psi_c + (\eta + x^T \Psi_c x) p p^T - \Psi_c x p^T - p x^T \Psi_c^T) \quad (17)$$

となり，それ以外 ( $\|\Phi_c x\| \leq \theta_0$ ) のとき

$$\Phi_c := \Phi_c \quad (18)$$

$$\Psi_c := \eta^{-1} (\Psi_c - \Psi_c x p^T) \quad (19)$$

とする．ただしこれらの行列の初期値はすべての  $i \in I$  について  $M_i = O_{1 \times n}$  (零行列)， $\Phi_i = I_{n \times n}$  (単位行列)， $\Psi_i = O_{n \times n}$  とする．

### 2.5.3 勾配法による $x_c$ の更新

CAN2-1 では  $M_c$  を一定として勾配法により  $x_c$  を更新する．(10) 式のエネルギー  $E$  の  $x_c$  の変化に対する勾配は

$$\frac{\partial E}{\partial x_c} = \sum_{i \in A_c} \int_{W_{ci}} (e_c^2(x) - e_i^2(x)) p(x) \frac{x - x_c}{\|x_i - x_c\|} dx \quad (20)$$

となる (付録 A 参照)．ここで  $A_c$  は領域  $V_c$  の隣接領域  $V_i$  の添字  $i$  の集合であり，境界  $W_{ci}$  は  $V_c$  と  $V_i$  の境界である．よって各時刻  $t$  において  $x$  が境界  $W_{ci}$  に入る時のみこの勾配を用いて荷重を修正すればよい．しかし幅が 0 の境界  $W_{ci}$  に  $x$  が入

る確率は非常に低いので、幅 $\epsilon \in (0, 1]$ を考え、

$$\frac{|(2\mathbf{x} - \mathbf{x}_i - \mathbf{x}_c)^T(\mathbf{x}_i - \mathbf{x}_c)|}{\|\mathbf{x}_i - \mathbf{x}_c\|^2} \leq \epsilon \quad (21)$$

を満たす場合に

$$\mathbf{x}_c := \mathbf{x}_c - \alpha(e_c^2(\mathbf{x}) - e_i^2(\mathbf{x})) \frac{\mathbf{x} - \mathbf{x}_c}{\|\mathbf{x}_i - \mathbf{x}_c\|} \quad (22)$$

により $\mathbf{x}_c$ を更新する。ここで $\alpha$ はこのセルの発火回数 $v_c$ の増大とともに減少する関数 $\alpha = \exp(-v_c/\tau_v)$ とし、 $\tau_v$ は定数とする。なお(21)式の条件は厳密にはすべての隣接ベクトル $\mathbf{x}_i$  ( $i \in A_c$ )について調べないといけないが、本アルゴリズムではその近似として $\mathbf{x}_c$ に最も近い順に $N_c = n$ 個( $\mathbf{x}$ の次元数個)の $v_i > \theta_v$ となる $\mathbf{x}_i$ について調べ、初めに(21)式を満たした $\mathbf{x}_i$ に対してのみ(22)式を適用した。ここで $\theta_v$ は各セルが十分な個数のデータを記憶しているかを判断するための正定数である。

#### 2.5.4 非効率モデルの再初期化

競合に勝った第 $c$ セルが十分多くのデータを記憶している( $v_c \geq \theta_v$ )にも関わらず、近似誤差が大きい( $|e_c(\mathbf{x})| \geq \theta_e$ )ときは、最小の発火回数 $v_l$ をもつ第 $l$ セルを探しその記憶データに第 $c$ セルの記憶データをすべて複製し、 $c$ を $l$ に置き換えた(13)~(22)式を用いて第 $l$ セルの学習を行う。ここで $\theta_e$ は許容誤差を表す正定数である。この処理は競合学習法で更新困難な利用頻度最小の第 $l$ セルの非効率モデル $M_l$ を、十分多くのデータが学習されているにも関わらず近似誤差が大きい領域 $V_c$ を細分割するモデルとして再初期化することにより、競合学習法の欠点を補うものであると言える。

#### 2.5.5 データの初期化と学習アルゴリズム

以上のデータ更新手続きはデータの初期化手続きを付加した次の学習アルゴリズムに統合し使用する。

(1) 各時刻 $t$ において、(3)式または(6)式により競合に勝つ第 $c$ セルを求める。

(2)  $v_c = 0$ ならば、このセルのデータ $v_c$ ,  $M_c$ ,  $\Phi_c$ ,  $\Psi_c$ を(13)~(19)式を用いて更新するとともに、CAN2-1については $\mathbf{x}_c := \mathbf{x}$ とする。これを第 $c$ セルのデータの初期化という。

(3)  $v_c < \theta_v$ または $|e_c(\mathbf{x})| < \theta_e$ のときは、(13)~(22)式を用いて学習し、そうでないときは以下の処理を行う。もし $v_i = 0$ となるセルがある場合は、そのなかのひとつのセルを選び、そのセルのデータの初期化を行う。そうでない場合は非効率モデルの再初期化を行う。

### 3. 制御方法：予測制御

以上のCAN2によるモデルの学習と切り替えにより、その時刻のプラントの同定モデルが求まると、種々の制御手法が適用可能となる。たとえばNarendraとBalakrishnan<sup>2)</sup>はモデル規範型、Kangら<sup>3)</sup>は極配置法を用いているが、本研究ではむだ時間を含むプラントや非線形プラントにも適用可能な予測制御<sup>14)~16), 18)</sup>に基づいて定式化する。

#### 3.1 予測制御

時刻 $t$ においてプラントの同定モデルが(1)式の形式で求まると、時刻 $t+k$  ( $k = 1, 2, \dots$ )のプラントの出力 $y(t+k)$ は $t$ 以前の入出力 $u(t-i-1)$ と $y(t-i)$  ( $i = 0, 1, 2, \dots$ )および $t$ 以後の仮想的な入力すなわち予測入力 $\hat{u}(t+j-1)$  ( $j = 1, 2, \dots, k$ )を用いて計算できるので、次の評価関数

$$J = \sum_{j=N_1}^{N_2} \hat{e}^2(t+j) + \lambda_u \sum_{j=1}^{N_u} \Delta \hat{u}^2(t+j-1) \quad (23)$$

を最小にする $\hat{u}(t)$ を計算し、プラントへの操作量 $u(t) = \hat{u}(t)$ とすることができる。ここで第1項は目標値 $r(t+j)$ と予測出力 $\hat{y}(t+j)$ の差、すなわち予測される追従誤差 $\hat{e}(t+j) = r(t+j) - \hat{y}(t+j)$ を小さくするための項である。第2項は操作量の変動 $\Delta \hat{u}(t+j-1) = \hat{u}(t+j-1) - \hat{u}(t+j-2)$ を抑制するための項である。さらにプラントが線形モデルとして同定される場合は一般化予測制御(GPC)の手法を用いることにより上の評価関数は次式に変形できる。<sup>14)</sup>

$$J = (\mathbf{r} - \mathbf{G}\Delta\hat{\mathbf{u}} - \mathbf{p})^T (\mathbf{r} - \mathbf{G}\Delta\hat{\mathbf{u}} - \mathbf{p}) + \lambda_u \Delta\hat{\mathbf{u}}^T \Delta\hat{\mathbf{u}} \quad (24)$$

ここで、

$$\mathbf{r} = (r(t+N_1), \dots, r(t+N_2))^T \quad (25)$$

$$\mathbf{p} = (p(t+N_1), \dots, p(t+N_2))^T \quad (26)$$

$$\Delta\hat{\mathbf{u}} = (\Delta\hat{u}(t), \dots, \Delta\hat{u}(t+N_u-1))^T \quad (27)$$

であり、 $p(t+j)$  ( $j = N_1, N_1+1, \dots$ )は時刻 $t$ 以降の操作量 $\hat{u}(t+j-1)$ を $u(t-1)$ と等しくして計算した場合の第 $(t+j)$ 時刻におけるプラントの仮想出力である。行列 $\mathbf{G} \in \mathbb{R}^{(N_2-N_1+1) \times N_u}$ の第 $ij$ 成分は $G_{ij} = g_{i-j+N_1}$ であり、 $g_k$  ( $k = \dots, -2, -1, 0, 1, 2, \dots$ )は $y(i) = u(i) = 0$  ( $i < 0$ )および $u(i) = 1$  ( $i \geq 0$ )としたときのこのプラントの出力 $y(k)$ すなわち単位ステップ応答である。すると $J$ を最小にする $\Delta\hat{\mathbf{u}}(t)$ は

$$\Delta\hat{\mathbf{u}} = (\mathbf{G}^T \mathbf{G} + \lambda_u \mathbf{I})^{-1} \mathbf{G}^T (\mathbf{r} - \mathbf{p}) \quad (28)$$

となり、その時刻 $t$ の操作量を $u(t) = u(t-1) + \Delta\hat{u}(t)$ として求めることができる。

この制御法で調整できるパラメタは $N_1$ ,  $N_2$ ,  $N_u$ および $\lambda_u$ であり、これらの値に依存する追従性能や安定性の解析<sup>14)</sup>、およびある条件下の制御対象に対し $\lambda_u > 0$ として操作量変動を抑えかつ制御量のオフセットを抑えるための手法の提案<sup>17)</sup>などが行われている。しかしこれらの性能や提案は制御対象に種々の条件を仮定したとき有効になるものであり、実際の制御対象には許容される操作量変動やオフセットなどの要求仕様を考慮してこれらのパラメタ値を調節する必要がある。

#### 3.2 CAN2を用いる予測制御

パラメタ空間を分割するCAN2-2では、学習する関数が線形モデルで表されると仮定するので(6)式で選択される

$c = c(t)$  に対する線形モデル  $M_{c(t)}$  を上述の GPC の手法に適用することができる。一方、入力空間を分割する CAN2-1 は学習する関数が非線形であると仮定するので、(24) 式でなく (23) 式を直接評価し、かつ予測ステップ  $t+j$  ( $j = 1, 2, \dots$ ) ごとに  $M_{c(t+j)}$  を求める必要もある。高橋<sup>16)</sup> はニューラルネットを用いる予測制御の評価関数の最小値探索法としてシンプレックス法を用いているが、後述の非線形プラントの例では評価関数が非凸となり、シンプレックス法では求解できなかった。そこで CAN2-1 を用いる場合は許容される入力区間  $[u_{\min}, u_{\max}]$  を  $N_s$  等分した全点  $u_k$  ( $k = 0, 1, 2, \dots, N_s$ ) に対する全点探索を行う。この場合  $N_u$  や  $N_s$  の増大とともに計算時間が非常に増大することが問題になる。したがって  $N_u$  と  $N_s$  の値は使用計算機の能力と制御系の要求仕様を考慮して設定する必要がある。なお、以後、CAN2-1 と CAN2-2 を用いる予測制御を、それぞれ APC1 (Adaptive Predictive Controller 1) および APC2 とよぶ。

### 3.3 ランダム学習とオンライン学習

従来のモデル切り替え型適応制御では、プラントのパラメータ値が分布する場合<sup>2)</sup> や連続値を取り得る場合<sup>3)</sup> などを扱うため、事前実験によりその値の範囲を調査するとともに、目標軌道などに応じてどのような複数モデルをパラメータ空間上に配置すべきかを事前に検討する必要があった。

本手法は以上の手続きを CAN2 の学習機能を用いて行うものである。まずプラントに許容入力区間  $[u_{\min}, u_{\max}]$  内の値をとるランダムな入力を加えたときの入出力データを用いてネットの学習を行う。この学習 (以後、ランダム学習という) は従来手法の事前実験に相当し、パラメータ空間上で大まかに配置された複数のモデルを形成する。次に、目標軌道を用いて制御しながらネットの学習を行う。この学習 (以後、オンライン学習という) はランダム学習で生成したモデルを目標軌道により適したものに修正する。最終的な制御はこれらの学習を行ったネットを用いて行うが、この時にもオンライン学習を行う。

なお、ここで、ランダム学習は CAN2 のモデル  $M_i$  ( $i \in I$ ) の初期値を零行列としたことにより必要になった手続きであることを注意しておく。すなわちもしこの初期値を用いてオンライン学習のみを行うと CAN2 の出力あるいはプラントへの操作量は零となり、このときの上述のプラントの出力はほとんど零となり、プラントの入出力関係のモデル化誤差に基づくネットの学習がうまく機能しなくなるからである。これに代わる方法としては、モデル  $M_i$  の初期値にランダムな値を設定することなども考えられるが、そのときは  $\Phi_i$  や  $\Psi_i$  などを注意深く設定する必要がある。

## 4. 数値実験

いくつかの数値例を用いて提案手法の挙動を示す。目標軌道  $r(t)$  ( $t = 1, 2, \dots$ ) は時刻  $t = 1$  のとき 0 であり、以後  $t = 25, 75, 125, 175$  でそれぞれ  $0.5, 0.5, -0.5, -0.5$  ずつステップ状に変化する軌道とし、制御区間は  $0 < t \leq 200$  と

した。

### 4.1 パラメータが切り替わる線形プラント

パラメータが切り替わる線形プラント  $p_1$  :

$$y(t) = \begin{cases} 0.9y(t-1) + 0.1u(t-1) + d(t), & t \leq 50 \\ 0.9y(t-1) - 0.1u(t-1) + d(t), & 50 < t \leq 100 \\ 1.1y(t-1) + 0.2u(t-1) + d(t), & 100 < t \leq 150 \\ 1.1y(t-1) - 0.2u(t-1) + d(t), & 150 \leq t \end{cases} \quad (29)$$

を考える。ここで雑音  $d(t)$  は  $\pm 0.005$  の範囲の乱数とした。

このプラントに対しランダム学習とオンライン学習は時刻  $t = 1, 2, \dots, 200$  に対する 200 ステップを繰り返しながら行った。以下では簡単のためこの 200 ステップ 1 回を 1 イタレーションといい、さらにランダム学習を開始してから第  $i$  ( $i = 1, 2, \dots$ ) イタレーションにおける時刻  $t$  ( $t = 1, 2, \dots, 200$ ) における操作量、制御量、目標値をそれぞれ  $u_i(t)$ ,  $y_i(t)$ ,  $r_i(t) = r(t)$  で表し、ネットへの入力を  $\mathbf{x}_i(t)$ , 選択されたモデルの番号を  $c_i(t)$  で表す。またランダム回数を  $N_R$ , オンライン学習回数を  $N_O$  で表す。

ネットが (29) 式のプラントを学習して再現するにはネットへの入力ベクトル  $\mathbf{x}(t) = (1, y(t-1), u(t-1))^T$  の値が同じでも異なる時刻には異なる値  $y(t)$  を出力する能力が必要となる。CAN2-1 ではこれに対処するため (9) 式の保持率  $\eta$  を調節することが考えられる。しかし CAN2-1 を用いる APC1 がこの目標軌道に対してこのプラントをうまく制御するパラメータ値を  $\eta = 1.00, 0.95, 0.90, 0.85, 0.80, \dots$  などに設定して探したが見つけることはできなかった。この原因は次節 4.2 の  $\eta$  に関する考察を参照のこと。

これに対し CAN2-2 を用いる APC2 ではかなり広範囲のパラメータ値に対してうまく学習し、制御できた。その一例として 4.4 に示す設定方針に従って制御パラメータを設定し、ランダム学習回数を  $N_R = 200$  およびオンライン学習回数を  $N_O = 1800$  としたときの、同定誤差  $E^I(k)$  と追従誤差  $E^C(k)$  の推移を Fig. 1 に示す。ここで  $E^I(k)$  と  $E^C(k)$  は各イタレーション  $i$  ( $i = 1, 2, \dots, 10$ ) ごとに時刻  $t$  ( $t = 1, 2, \dots, 200$ ) までの各誤差の和を

$$E_i^I(t) = \sum_{j=1}^t (y_i(j) - M_{c_i(j)} \mathbf{x}_i(j))^2 \quad (30)$$

$$E_i^C(t) = \sum_{j=1}^t (r_i(j) - y_i(j))^2 \quad (31)$$

として求め、時刻の総数  $k = 200(i-1) + t$  の関数  $E^I(k) = E_i^I(t)$  および  $E^C(k) = E_i^C(t)$  として表したものである。この  $E^I(k)$  は、(9) 式で保持率を  $\eta = 1$  としたネットのエネルギーを各イタレーションごとに初期化しているものとみることができ、これによりネットの学習能力を評価することができる。まず各イタレーションの最終時刻すなわち  $k = 200, 400, \dots, 2000$  における同定誤差の変化を見ると、

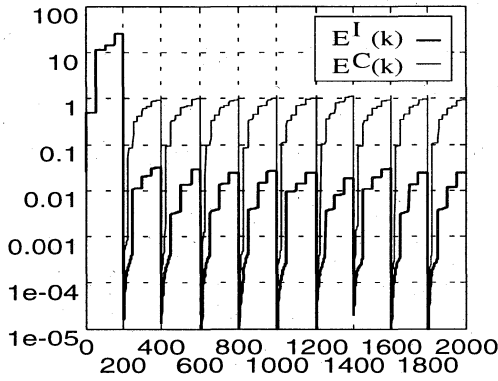
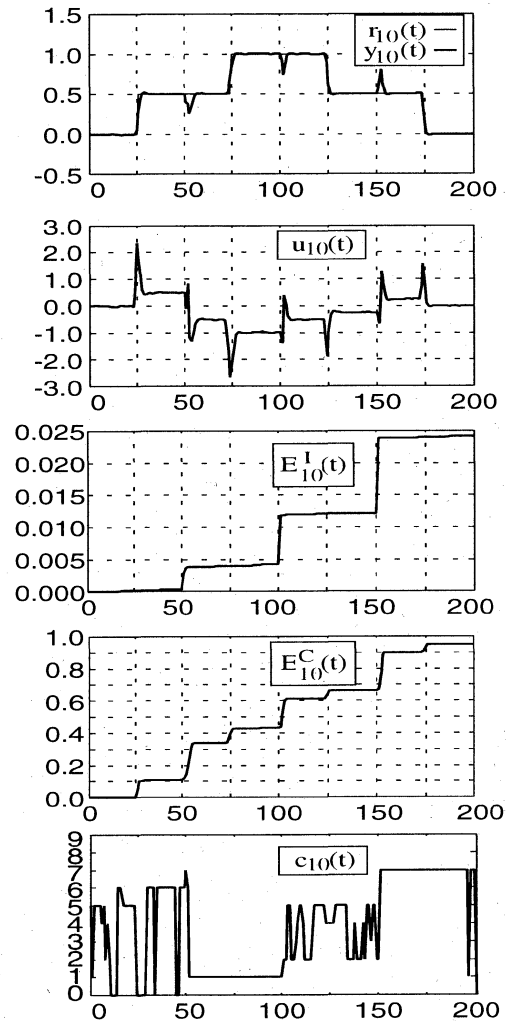


Fig. 1 The identification error  $E^I(k)$  and the tracking error  $E^C(k)$  of the APC2 applied to  $p_1$ . The parameter values of the APC2 are as follows:  $|I| = 10$ ,  $n_y = n_u = 1$ ,  $N_e = 4$ ,  $\eta = 1.0$ ,  $\theta_0 = 10^{-13}$ ,  $\tau_v = 100$ ,  $\theta_e = 0.1$ ,  $\theta_v = n = 3$ ,  $N_1 = 1$ ,  $N_2 = 3$ ,  $\lambda_u = 0.01$ ,  $u_{\max} = -u_{\min} = 10$ .

ランダム学習 ( $k = 200$ ) からオンライン学習 ( $k = 400$ ) に移るとき同定誤差が大きく減少していることがわかる (誤差は対数目盛で表示されていることに注意). これはオンライン学習時には目標軌道に沿ったデータのみが学習されて評価されるのでオンライン学習時の同定誤差はランダム学習時よりも一般に小さくなるからである. オンライン学習の期間では  $k = 400, 600, \dots, 2000$  における同定誤差および追従誤差はともにわずかに振動するがほぼ一定となっている. これはオンライン学習の最初のイタレーション後 ( $k = 400$ ) に学習がほぼ収束していることを意味する.

次に各イタレーション ( $200(i-1) < k < 200i$ ) における同定誤差を見ると, 誤差が階段状に大きく増加している部分が3箇所あることがわかる. これを詳しく調べるため最終の第  $i = 10$  イタレーション ( $1800 < k < 2000$ ) における制御結果を Fig. 2 に示す. Fig. 2 の  $E_{10}^I(t)$  からプラントのパラメタが切り替わる  $t = 50, 100, 150$  の直後に同定誤差が大きく増加していることが分る. これらはパラメタが切り替わる直前のデータでモデルを同定し, 切り替わった直後のデータでその同定誤差を評価するために生じた誤差であり, 基本的にネットの学習能力に起因するものではない. 逆に, CAN2はこのような誤りを含むデータに対応するモデルを再初期化を有する学習により複数モデルのひとつとして形成することにより, それまでに学習された正しいモデルへの影響を防ぐ機能をもつ. 具体的に  $0 < k < 2000$  の範囲で再初期化による学習が起こった時刻を調べると  $k = 902, 952, 1352, 1452, 1752, 1952$  であり, 再初期化されたセルの番号はそれぞれ 8, 4, 3, 6, 2, 9 であった. ここでこれらの時刻とセルが不規則な原因を調べるため雑音を  $d(t) = 0$  とすると再初期化は  $k = 852$  のとき第4セルに対してのみ行われた. さらに  $d(t)$  が零または非零にかかわらず, 使用するセルの個数  $|I|$  が8以上のときはうまく制御できたが, 7以下では発散した. これはプラントの4つの真のモデルの他に4つのモデルが必要だったことを



$i$	$M_{i0}$	$M_{i1}$	$M_{i2}$
0	0.00084	0.89976	0.09995
1	-0.00029	0.89894	-0.09993
2	0.00088	1.09925	0.19795
3	0.00083	1.09917	0.19842
4	0.00076	1.09932	0.19871
5	-0.00013	1.09999	0.19998
6	0.00070	0.90005	0.09922
7	-0.00078	1.10000	-0.20001
8	-0.05827	0.87081	-0.12950
9	0.07836	1.08374	0.25883

Fig. 2 Results of the APC2 applied to  $p_1$  at the 10th iteration, where, from the top, the output  $y_{10}(t)$ , the input  $u_{10}(t)$ , the tracking error  $E_{10}^C(t)$ , the identification error  $E_{10}^I(t)$ , the index of switched model  $c_{10}(t)$ , and all the models after this iteration  $M_i = (M_{i0}, M_{i1}, M_{i2})$  for  $i = 0, 1, \dots, 9$ .

表す. なお  $\theta_e = 100$  として再初期化が行われないようにすると, ひとつのセルのみが学習を繰り返す, うまく制御することはできなかった. 結局, CAN2の再初期化を有する複数モデルの学習機能は誤りのある同定誤差を吸収しつつ正しいモデルを形成し, プラントをうまく制御することに寄与したものである. Narendra と Balakrishnan<sup>2)</sup>が複数の固定モデ

ルと1個の適応モデルおよび1個の再初期化適応モデルを用いてうまく制御できたのもこれと同様の原理であると考えられるが、彼らの手法では固定モデルをあらかじめうまく配置する必要があり、本手法の有効性のひとつが理解できる。

Fig. 2の追従誤差和  $E_{10}^C(t)$  はプラントのパラメタが切り替わった直後および目標軌道がステップ状に変化した直後に増加するがその他の区間ではほとんど増加しないことが分る。また(29)式とこの図の  $M_i$  ( $i = 0, 1, 2, \dots, 9$ )を対比すると、 $t$ の区間がそれぞれ  $t \leq 50$ ,  $50 < t \leq 100$ ,  $100 < t \leq 150$ ,  $150 < t$ におけるプラントの近似モデルが  $i = 0, 6$ ,  $i = 1$ ,  $i = 2, 3, 4, 5$ ,  $i = 7$ に対する  $M_i$ により学習されていることがわかる。一方、競合に勝ったセルの番号  $c_{10}(t)$  をみると各区間でこの対応とは異なるモデルも選択されているが  $y_{10}(t)$  と  $u_{10}(t)$  をみるとうまく制御できている。これはプラントの入出力が一定値、たとえば  $u(t) = \bar{u}$  と  $y(t) = \bar{y}$ 、をとるようになると、(29)式のプラントにおいて同定すべきモデル  $M_i = (M_{i0}, M_{i1}, M_{i2})$  が満たすべき条件は、 $\bar{y} = M_{i0} + M_{i1}\bar{y} + M_{i2}\bar{u}$  となり、さまざまな  $M_{ij}$  ( $j = 0, 1, 2$ ) がこの条件を満たし得るからである。これらは目標軌道に依存して縮退したプラントモデルと言えるが、プラント出力を予測する上では問題ないと言える。たとえば区間  $0 < t < 25$  において  $r(t) = y(t) = u(t) = 0$  のときは、モデル  $M_i$  の第1成分  $M_{i0}$  が小さい方がより良い予測を行うことができる。

さらにこのプラントの出力  $y(t)$  には  $\pm 0.005$  の範囲の確率雑音  $d(t)$  が加わるので、これによるモデル切り替えが  $c_{10}(t)$  で観測されるが、うまく制御できている。この結果は本手法の以下の諸機能がうまく働いたからであると考えられる。まずゼロ平均の確率雑音のモデル学習への影響は最小二乗法により軽減され、モデル同定への影響は(7)式の  $N_e$  を大きくすることにより軽減できる。しかし  $N_e$  を大きくすると適切なモデルの同定が遅れることになり、不安定系では正しく同定する前に発散する可能性がでてくる。このプラントも  $t > 100$  で不安定な系であり  $N_e$  を5以上にすると発散した。次に、一般に制御のためのモデルが切り替えられると操作量が振動する可能性がでてくるが、本手法では制御系内に状態量(たとえばPI制御の積分動作など)を保持しないことおよび(23)式の  $\lambda_u$  を調節して入力変動を抑制する機能をもつことから、モデル切り替えによる操作量振動はある程度抑制できる。

#### 4.2 むだ時間と時定数が変化する線形プラント

伝達関数が

$$G(s) = \frac{e^{-Ls}}{s + \alpha} \quad (32)$$

で与えられ、そのむだ時間  $L$  とパラメタ  $\alpha$  が

$$L = T(1.0 - \cos(4\pi t/200)) \quad (33)$$

$$\alpha = 20 + 10 \cos(2\pi t/200), \quad (34)$$

のように変化するプラントを周期  $T = 0.01$  で1次ホールドによりサンプリングして  $\pm 0.005$  の範囲の確率雑音  $d(t)$  を加

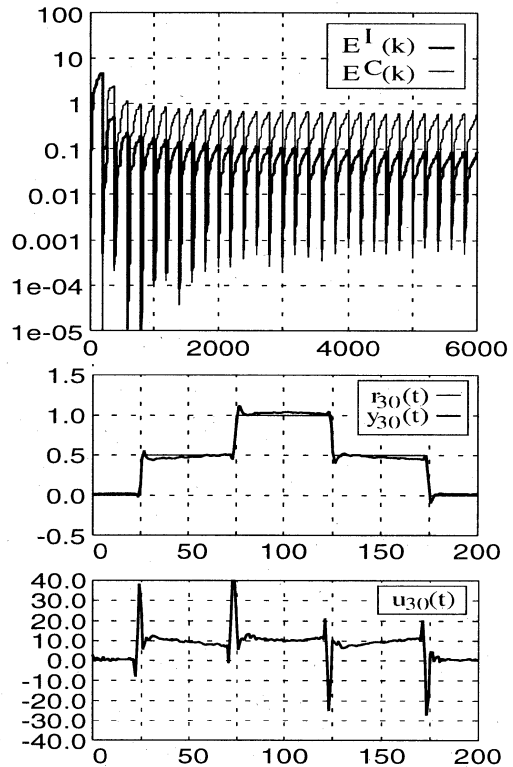


Fig. 3 Results of the APC1 with  $|I| = 1$  applied to  $p_2$ . The parameter values are as follows:  $n_y = 1$ ,  $n_u = 3$ ,  $N_e = 10$ ,  $\eta = 1.0$ ,  $\theta_0 = 10^{-13}$ ,  $\tau_v = 100$ ,  $\theta_e = 0.02$ ,  $\theta_v = n = 5$ ,  $N_1 = 1$ ,  $N_2 = N_u = 5$ ,  $\lambda_u = 1.0e - 5$ ,  $u_{\max} = -u_{\min} = 40$ .

えた離散時間プラント  $p_2$  :

$$y(t) = a_1 y(t-1) + b_K u(t-K) + b_{K+1} u(t-K-1) + d(t) \quad (35)$$

を考える。ここで

$$a_1 = e^{-\alpha T} \quad (36)$$

$$b_K = (1 - e^{-(K-L/T)\alpha T}) / \alpha \quad (37)$$

$$b_{K+1} = (e^{-(K-L/T)\alpha T} - e^{-\alpha T}) / \alpha \quad (38)$$

であり、 $K = \lfloor L/T \rfloor$ 、および  $[x]$  は  $x (\in \mathbb{R})$  を越えない最大の整数を表す。(33)式からわかるようにむだ時間  $L$  はサンプリング周期  $T$  を越える  $0 \leq L \leq 2T$  で変動し、 $K$  は 0, 1 または 2 となる。

このプラントは前例よりもパラメタの変化が緩やかであるが、複数のセルをもつ APC1 では制御できなかった。1個のセルを用いる APC1 では、保持率を  $\eta = 1.00$  とし、ランダム学習を1イタレーション ( $N_R = 200$ )、オンライン学習を29イタレーション ( $N_O = 5800$ ) 行うと Fig. 3の結果が得られた。ここで1個のセルを用いる APC1 は1個のセルを用いる APC2 と同じであり、最小二乗法によりプラントパラメタを同定して予測制御を行う単純な適応制御であることを注意しておく。図の同定誤差  $E^I(k)$  と追従誤差  $E^C(k)$  は最終の第30イタレーションではほぼ収束しているが、 $y_{30}(t)$  は

プラントパラメタの変動により目標値付近で緩やかに変動している。そこで保持率を $\eta = 1.00, 0.95, 0.90, 0.85, 0.80$ と変化させて実験を行ったが、 $\eta < 1.00$ でオンライン学習のイタレーションを繰り返すと、追従誤差は一度は減少するがあるイタレーション回数を越えると発散した。この原因は本実験では200ステップの学習を繰り返すのでその影響が最小二乗法のバイアス<sup>13)</sup>として同定モデルに残るからではないかと考えられる。なお、前節4.1のプラント $p_1$ がAPC1で制御できなかったのも、これと同様の原因、およびプラント $p_1$ ではプラントパラメタが瞬時に大きく切り替わるために $\eta < 1$ による適応がより困難であったものと考えられる。なお、次節のプラント $p_3$ の $\eta$ に関する考察も参照のこと。

一方、複数のセルを用いるAPC2ではFig. 4の結果が得られ、この図から第30イタレーションにおける出力 $y_{30}(t)$ の累積追従誤差 $E_C(6000)$ はFig. 3よりも小さいことがわかる。出力 $y_{30}(t)$ および入力 $u_{30}(t)$ には小さな振動が見られるが、これは確率雑音 $d(t)$ とモデルの切り替えの影響によるものである。 $N_e = 10$ としているので $d(t)$ の影響は前例の $N_e = 4$ の場合より小さくなっていると考えられるが、連続的に変化するパラメタをもつプラントを複数の線形モデルの切り替えにより同定しているため、前例と異なり $\lambda_u$ を大きくすると滑らかな入力は得られたが、追従誤差が大きくなった。

#### 4.3 非凸な非線形プラント

非凸な入出力関数をもつプラント $p_3$ ：

$$y(t+1) = 15u(t)^3 - 16u(t)^2 + 5u(t) + d(t) \quad (39)$$

をAPC2で制御する過程をFig. 5を用いて考える。時刻 $t$ のとき図中の2点 $(u(t-2), y(t-1))$ と $(u(t-1), y(t))$ から $N_e = 2$ のときの予測モデル $y = \hat{M}(t)u$ が得られるので、目標値 $r(t+1)$ に達するための入力は、図の $\hat{u}(t)$ となり、これはその最適解 $u^*(t)$ と異なる。すると $u(t) = \hat{u}(t)$ によるプラント出力 $y(t+1)$ は $r(t+1)$ に一致せず、さらにこの過程を続けていくと発散することになる。

これに対し入力空間を分割するAPC1はこの非線形系を制御できた。その結果の一例をFig. 6に示す。図の $c_{30}(t)$ を見ると $|I| = 20$ 個のセルの内のはほとんどは使用されていないが、うまく制御するにはこの数が必要であった。これはそれらのセルが再初期化に使用されたためであり、再初期化を抑制するためにしきい値 $\theta_e$ を大きくするとセル数を少なくできたが追従誤差が大きくなった。また、保持率 $\eta$ の影響を調べるため1.0から0.5刻みに減少させて実験したが、 $\eta = 0.65$ まではほぼ同等の制御結果が得られ、0.60では発散した。この結果は4.1と4.2の結果も合わせると、1より小さい保持率 $\eta$ を用いるのは時変プラントに対しては注意が必要であることを示している。この原因の詳細な説明は今後の課題である。

#### 4.4 制御パラメタの設定方針

本手法は多くの制御パラメタをもち、その値に依存して

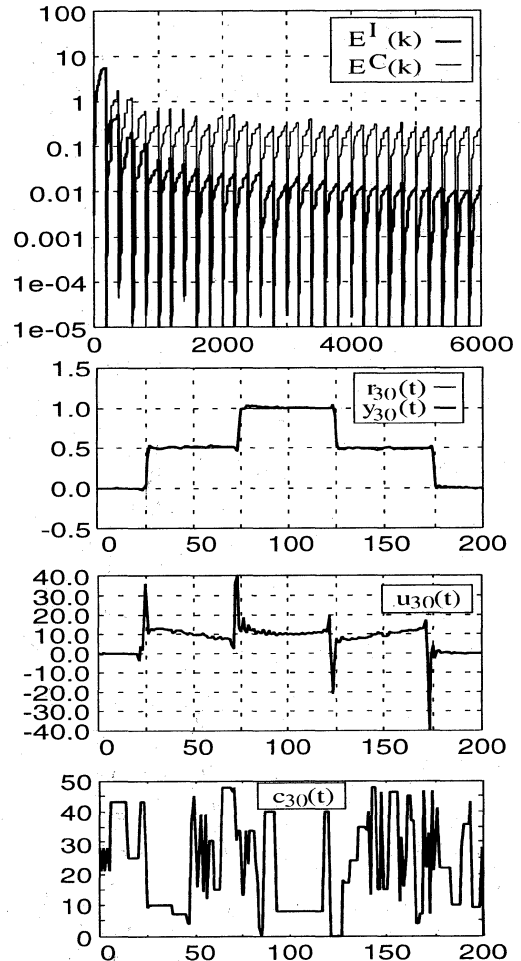


Fig. 4 Results of the APC2 with  $|I| = 50$  applied to  $p_2$ . The parameter values except  $|I|$  are the same as in Fig. 3.

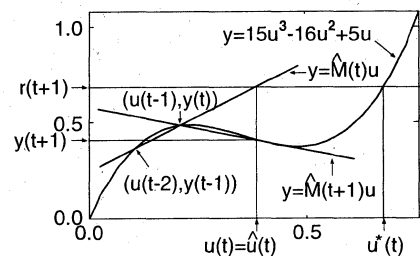


Fig. 5 Illustrative control process of APC2 for  $p_3$ .

様々な応答が得られるが、本シミュレーションでは以下の設定方針に基づく制御パラメタ値を用いた。これとは異なる設定方針の議論や、パラメタ間の相互関係の解析も重要であるが、より複雑な議論が必要であり、今後の課題である。

- (1) ネットへの入力ベクトルの次元 $n_y$ と $n_u$ ：(2) 式の $n_y$ と $n_u$ は既知としてそれらの値を用いた。またそれらが変動する場合はそれらの最大値を用いた。これらはより大きく設定してもより多くの学習を行えば制御性能にはほとんど影響を与えなかったが、これは各セルが誤差の二乗を最小化するように線形モデル(連想行列)を学習することから理解できる。



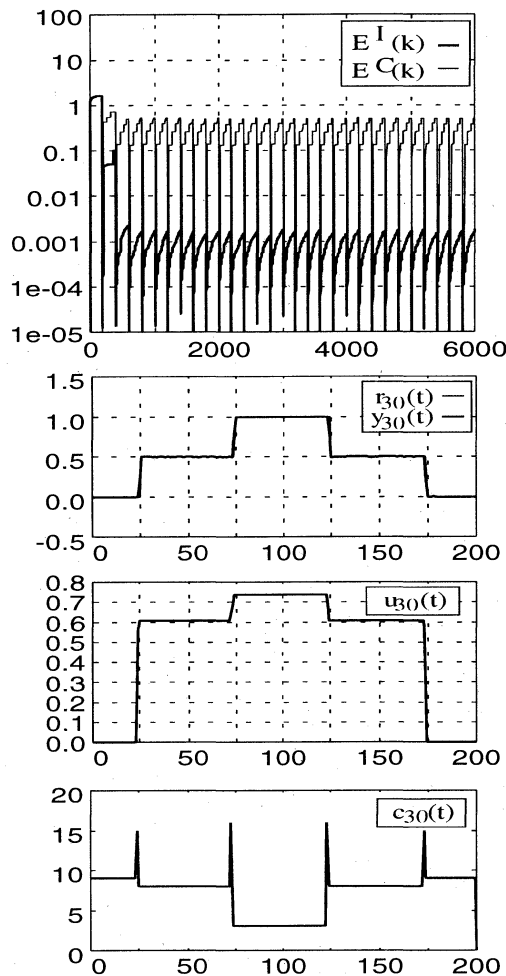


Fig. 6 Results of the APC1 applied to  $p_3$ ,  $|I| = 20$ ,  $n_y = 0$ ,  $n_u = 1$ ,  $N_e = 2$ ,  $\eta = 1.0$ ,  $\theta_0 = 10^{-13}$ ,  $\tau_v = 100$ ,  $\theta_e = 0.02$ ,  $\theta_v = n = 2$ ,  $N_1 = 1$ ,  $N_2 = 2$ ,  $N_u = 1$ ,  $\lambda_u = 0$ ,  $u_{\max} = 0.8$ ,  $u_{\min} = 0$  and  $N_s = 100$ .

(2) セルの個数  $|I|$ : プラントを同定するために最終的に必要な線形モデルの個数が分かっている場合はその個数の2~3倍程度の値を設定する。これはそれまでに記憶された正しいモデルが再初期化機能により変更されないようにするためである。連続領域内のパラメタ値をもつプラントを同定する場合の  $|I|$  の決定は難しいが、本数値例では制御区間が  $0 < t \leq 200$  であるので200個以上のモデルは最終的には必要ないのでこの2~3倍程度を限度とした。

(3) モデル同定のためのデータ数  $N_e$ : 外乱が小さい場合は  $N_e = \dim(\mathbf{x}(t)) = n$  で十分な同定が可能であるが、外乱が大きい場合はそれより少し大きくする。

(4) エネルギーの保持率  $\eta$ : 基本的に  $\eta = 1$  とする。

(5) (14) 式の  $p(t)$  を計算するための  $\theta_0$ : 基本的には0を表す値であるが、計算機の演算精度によってはあまり小さくはできない。本実験では64ビットの浮動小数点演算に対し  $\theta_0 = 10^{-13}$  とした。

(6) (22) 式の  $\alpha$  を決定する時定数  $\tau_v$ : 数イタレーションで  $\mathbf{x}_v$  の学習が収束するように  $\tau_v = 100$  とした。

(7) 再初期化のためのしきい値  $\theta_e$  と  $\theta_v$ : 目標値や外乱の大きさおよびプラントの性質などに依存して決めるが、これらが不適切に設定されると再初期化が起こり過ぎたり、起こらなかったりする。本実験では  $\theta_e = 0.02$  から0.1の値を設定した。 $\theta_v$  は外乱が小さいときは入力ベクトルの次元数  $n = \dim(\mathbf{x}(t))$  の独立なベクトルがあればよいと考えられるので  $\theta_v = n$  とした。

(8) 予測制御に用いるパラメタ:  $N_1 = 1$ ,  $N_2 = n$  とした。APC2では  $N_u = n$  とした。APC1では  $N_u = n$  についても実験したが、大きな  $N_u$  を用いるとAPC1では時間がかかり過ぎるのでより実用的な  $N_u = 1$  の場合のみを示した。 $\lambda_u$  は操作量の振動を見て調整した。

(9) 予測入力計算のための範囲  $[u_{\min}, u_{\max}]$ : 実際のプラントに依存して決定すべきであるが、本実験では1ステップで目標値に到達できる入力値をもとに  $u_{\max}$  と  $u_{\min}$  を決定した。

## 5. おわりに

本論文では複数のモデルの学習と切り替えを行う競合連想ネットワーク CAN2 を用いる予測制御の方法を示した。従来の切り替え型適応制御において複数の適応モデルを用いる場合、効率の悪いモデルが学習されるとうまく制御できない可能性があることを指摘し、これを解決する CAN2 の学習法を提案した。また複数の固定モデルを使用する場合、従来は対象プラントのモデル分布をあらかじめ調査してモデルを設定する必要があったが、本学習法はこの設定を自動化するものである。従来手法と同様に同定誤差に基づいてモデルを切り替える APC2 はパラメタ値が変化するプラントに、入力に依存してモデルを切り替える APC1 は非凸な非線形プラントに、それぞれ有効であることがわかった。今後、ネットに必要なセル数や記憶容量を減少させる方法、より効率的な学習アルゴリズムなどについて検討する予定である。なお本研究の一部は文部省科研費 (12680389) の援助を受けた。

## 参考文献

- 1) A.S.Morse: Supervisory Control of Families of Linear Set-Point Controllers-Part 1: Exact Matching, IEEE Trans. on Automatic Control, **41**-10, 1413/1431 (1996)
- 2) K.S.Narendra and J.Balakrishnan: Adaptive Control Using Multiple Models, IEEE Trans. on Automatic Control, **42**-2, 171/187 (1997)
- 3) Z. Kang, S.Fujii, C.Zhou and K.Ogata: Adaptive Control of a Planar Gantry Crane by the Switching of Controllers, Trans. of the Society of Instrument and Control Engineers, **35**-2, 253/261 (1999)
- 4) 藤田政之, 平田研二: スイッチング制御, 計測と制御, **38**-3, 176/181 (1999)
- 5) 池田建司: 切り替え型適応制御, 計測と制御, **38**-3, 182/188 (1999)
- 6) N.Ueda and R.Nakano: A New Competitive Learning Approach Based on an Equidistortion Principle for Designing Optimal Vector Quantizers, Neural Networks, **7**-8, 1211/1227 (1994)
- 7) S.Ren and S.Kurogi: Learning Algorithms Using Firing

- Numbers of Weight Vectors for WTA Networks in Rotation Invariant Pattern Classification, IEICE TRANS. on FECCS, **E81-A-1**, 175/182 (1998)
- 8) S.Kurogi and S.Ren: Competitive Associative Networks for Function Approximation and Control of Plants, Proceedings 1997 International Symposium on Nonlinear Theory and its Applications, 2, 775/778 (1997)
  - 9) S.Ren and S.Kurogi: Competitive Associative Networks for Control of Linear and Nonlinear Plants, Proceedings of the Third International Symposium on Artificial Life and Robotics (AROB 3rd '98), 2, 723/726 (1998)
  - 10) S.Kurogi, A.Ikushima and S.Ren: Neuro-Controllers Using Competitive Associative Nets Requiring Neither Parameterization of Plants Nor Special Training, International Conference on Neural Information Processing, 2, 679/682 (1998)
  - 11) S.Kurogi, H.Hirata, K.Nishida and T.Sakamoto: Model-Switching Adaptive Control Using Competitive Associative Net, Proceedings of 1999 International Symposium on Nonlinear Theory and its Applications, 2, 755/7768 (1999)
  - 12) T.Kohonen: Associative Memory, A system-theoretical approach, Springer-Verlag (1977)
  - 13) 谷萩隆嗣: デジタル信号処理の理論3, コロナ社 (1986)
  - 14) D. W. Clarke and C. Mohtadi: Properties of Generalized Predictive Control, *Automatica*, **25-6**, 859/875 (1989)
  - 15) 高津 春雄: 内部モデル予測制御, コンピュートロール, 51/58, コロナ社 (1989)
  - 16) 高橋 安人: ニューラルネットワークによる非線形及び非定常系の最適予測適応制御, 科学技術社 (1992)
  - 17) 高橋 一成: 時変係数システムのための一般化予測制御, 計測自動制御学会論文集, **34-2**, 105/111 (1998)
  - 18) K.J.Hunt, D.Sbarbaro, R.Zbikowski and P.J.Gawthrop: Neural Networks for Control Systems-A Survey, *Automatica*, **28-6**, 1083/1112 (1992)
  - 19) J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P. Y. Glorennec, H. Hualmarsson and A. Juditsky: Nonlinear black-box modeling in system identification: a unified overview, *Automatica*, **31-12**, 1691/1724 (1995)
  - 20) M.Iri, K.Murota and T.Ohya: A Fast Voronoi-Diagram Algorithm With Applications to Geographical Optimization Problems, Lecture Notes in Controls and Information Science, 59, System Modelling and Optimization, eds. P.Thoft-Christensen, 273/288, Springer-Verlag (1984)

## 《付 録》

### A. エネルギー勾配の導出

(10) 式で  $e_i^2(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}_i\|^2$  とした場合のテンソル形式による解析<sup>20)</sup>を参考に, ここでは  $e_i^2(\mathbf{x})$  が (8) 式で与えられる場合のエネルギー勾配をベクトル形式で導出する. まずエネルギー  $E$  を微小領域におけるエネルギーの和とし,  $\mathbf{x}_c$  で偏微分すると次式を得る.

$$\frac{\partial E}{\partial \mathbf{x}_c} \simeq \frac{\partial}{\partial \mathbf{x}_c} \sum_{i \in I} \sum_{\Delta \mathbf{x} \in V_i} e_i^2(\mathbf{x}) p(\mathbf{x}) \Delta \mathbf{x} \quad (\text{A.1})$$

$$= \sum_{i \in I} \sum_{\Delta \mathbf{x} \in V_i} \left( \frac{\partial(e_i^2(\mathbf{x}) p(\mathbf{x}))}{\partial \mathbf{x}_c} \Delta \mathbf{x} + e_i^2(\mathbf{x}) p(\mathbf{x}) \frac{\partial \Delta \mathbf{x}}{\partial \mathbf{x}_c} \right) \quad (\text{A.2})$$

ここで (A.2) 式の第1項は  $e_i^2(\mathbf{x}) p(\mathbf{x}) = \|\mathbf{y} - \mathbf{M}_i \mathbf{x}\|^2 p(\mathbf{x})$  が  $\mathbf{x}_c$  に無関係なので零となる. 第2項は領域  $V_c$  の変化に

関する項であり,  $V_c$  の隣接領域  $V_i$  ( $i \in A_c$ ) との境界  $W_{ci}$  における値のみが残る. 以下この値を求める. 境界ベクトル  $\mathbf{x} \in W_{ci}$  は  $\mathbf{x}_c$  と  $\mathbf{x}_i$  の垂直二等分超平面上にあることから  $(\mathbf{x} - (\mathbf{x}_i + \mathbf{x}_c)/2)^T (\mathbf{x}_i - \mathbf{x}_c) = 0$  満たすのでこの式を  $\mathbf{x}_c$  で偏微分すると,

$$\left( \frac{\partial \mathbf{x}}{\partial \mathbf{x}_c} \right) (\mathbf{x}_i - \mathbf{x}_c) = \mathbf{x} - \mathbf{x}_c \quad (\text{A.3})$$

となる. ここで  $\partial \mathbf{x} / \partial \mathbf{x}_c$  は  $\partial x_j / \partial x_{ci}$  を  $i$  行  $j$  列とする行列である. 今,  $\mathbf{x}$  を正規直交基底ベクトル  $\mathbf{e}_k$  ( $k = 1, 2, \dots, n$ ) を用いて  $\mathbf{x} = \sum_k x_k \mathbf{e}_k$  で表し, かつその第1基底ベクトル  $\mathbf{e}_1$  が境界  $W_{ci}$  に直交して  $V_i$  の方向を向いている, すなわち  $\mathbf{e}_1 = (\mathbf{x}_i - \mathbf{x}_c) / \|\mathbf{x}_i - \mathbf{x}_c\|$ , として上式に代入すると

$$\frac{\partial x_j}{\partial x_c} = \begin{cases} \frac{\mathbf{x} - \mathbf{x}_c}{\|\mathbf{x}_i - \mathbf{x}_c\|}, & j = 1 \\ 0, & j = 2, 3, \dots, n \end{cases} \quad (\text{A.4})$$

が得られる. 微小変位  $\partial x_1 \mathbf{e}_1$  が  $V_i$  の領域の方向へ変化するとき,  $e_c^2(\mathbf{x})$  の領域  $V_c$  が増え  $e_i^2(\mathbf{x})$  の領域  $V_i$  が減るので, 結局

$$\frac{\partial E}{\partial \mathbf{x}_c} \simeq \sum_{i \in I} \sum_{\Delta \mathbf{x} \in V_i} e_i^2(\mathbf{x}) p(\mathbf{x}) \frac{\partial \Delta \mathbf{x}}{\partial \mathbf{x}_c} \quad (\text{A.5})$$

$$= \sum_{i \in A_c} \sum_{\Delta \mathbf{x} \in W_{ci}} (e_c^2(\mathbf{x}) - e_i^2(\mathbf{x})) p(\mathbf{x})$$

$$\frac{\mathbf{x} - \mathbf{x}_c}{\|\mathbf{x}_i - \mathbf{x}_c\|} \Delta x_2 \Delta x_3 \dots \Delta x_n \quad (\text{A.6})$$

となる. よって (20) 式が得られる.

## [著者紹介]

### 黒木 秀一 (正会員)



1980年九州工業大学工学部電気工学科卒業, 82年東京工業大学大学院総合理工学研究科システム科学専攻修士課程修了, 85年同大学院総合理工学研究科システム科学専攻博士課程修了, 同年九州工業大学工学部制御工学科助手を経て91年同助教授. 主にニューラルネットの研究に従事. 工学博士. 日本神経回路学会, 電子情報通信学会などの会員.

### 西田 健 (学生会員)



1998年九州工業大学工学部設計生産工学科卒業, 99年同大学院工学研究科博士前期課程修了(設計生産工学専攻), 同年同博士後期課程に進学し現在に至る. 主にニューラルネットによるパターン認識の研究に従事.