

定常性能を考慮した CMAC-PID 制御系の設計

八木 秀幸* 竹内 将雄** 下西 二郎* 山本 透*** 雛元 孝夫****

Design of CMAC-PID Control System Compensating Control Performance in Steady-State

Hideyuki YAGI, Masao TAKEUCHI, Jiro SHIMONISHI,
Toru YAMAMOTO and Takao HINAMOTO

In industrial processes, initiatives of production quality and cost reduction have created a demand. In such the situation, a concern with control performances in steady-state operation has been growing in recent years. On the other hand, Cerebellar Model Articulation Controller(CMAC) that is a kind of artificial neural networks has been proposed to control nonlinear systems. The CMAC has features of simple structure, a few design parameters and high-speed learning. A scheme for tuning PID parameters by using the CMACs has been also proposed. However, this scheme is focused on improvement of transient control performances. In this research, a new design of CMAC-PID control system compensating performance in steady-state is proposed. After weight tables included into the CMACs are learned, the weights that is employed in steady-state are re-tuned. Consequently, desirable control performances are obtained in both transient-state and steady-state.

Key Words : CMAC, PID control, nonlinear system, control performance in steady-state

1. ま え が き

化学プロセスなどに代表されるプロセス産業界では、今なおPID制御法が主に用いられている¹⁾。その理由としては、制御構造が簡単であることや、PIDパラメータの持つ物理的意味が明確であることなどが考えられる。また、PID制御系の性能はPIDパラメータによって大きく左右されるため、その調整法についてはさまざまな手法が提案されている。しかし、プロセス制御ではシステムの持つ非線形性や、その時間的な特性変動の影響のために、適切なPIDパラメータへ調整することは困難である。

この問題への対応として、脳内の神経伝達構造をモデル化したニューラルネットワーク (Neural Network:NN) を用いた制御法が数多く提案されている。NNは非線形システムを精度よく制御できる

有効な手法であるが、十分な学習を行うために多くのデータと時間を必要とする。また、NNの一種として、小脳演算モデル (Cerebellar Model Articulation Controller:CMAC) がAlbusによって提案されている²⁾。CMACは、ある入力点に対応する荷重値をその近傍のいくつかの入力点で共有する構造により非線形関数を学習する。この構造により、学習の際に入力点 (学習点) 近傍の入力に対しても学習効果を波及することができ、効率的な学習を行うことでNNと比べ学習回数を大幅に低減することができる。また構造が簡単で設計パラメータが少ないといった利点も有している。これらの特長を活かし、ロボット制御やプロセス制御、PIDパラメータを調整する手法³⁾ (以降CMAC-PID制御法と呼ぶ) などへ適用されている。

一方、近年プロセス産業界では運転コストの低減や製品品質の向上が望まれており、これを実現するためには時々刻々と変化するプラントシステムを常に最適な状態に制御することが重要であり、特にその定常性能が注目されている。このような状況において、定常状態での稼働データからシステムの制御性能を評価・監視する研究が近年盛んに

原稿受付 平成 20 年 8 月 29 日

* 電気電子工学科

** 専攻科 電子・情報システム工学専攻

*** 広島大学大学院 教育学研究科

**** 広島大学大学院 工学研究科

行われている^{4,5,6)}。しかしながら、NNやCMACなど知的情報処理を用いた制御系に関する研究の多くは過渡状態における制御性能（立ち上がり時間、行過ぎ量、整定時間など）の改善に着目しており、定常状態での制御性能を含めた考察はあまりされていない。

そこで本論文では、定常状態における制御性能を考慮したCMAC-PID制御系の設計について提案する。具体的には、まず最適な過渡応答波形が得られるようCMACをオフラインで学習させる。このとき同時にシステム同定を行い、その結果に基づき一般化予測制御（以降GPCと呼ぶ）により所望の定常特性が得られるPIDパラメータを求める。その値を用いてCMACの該当するデータを書き換える。これにより、過渡状態と定常状態の両面で所望の制御性能が得られるコントローラを構築することができる。本論文の構成を以下に示す。まずCMAC-PID制御系について説明し、定常状態での制御性能評価について詳述する。次にGPC則により定常状態に適したPIDパラメータの導出法について示し、CMAC-PID制御系へ適用する方法について提案する。最後に本手法を非線形システムに適用した数値例を示し、その有用性を検証する。

2. CMAC-PID 制御系

2.1 CMAC の概要

CMACは人の小脳皮質内における情報処理メカニズムを数学的にモデル化したものである。CMACは入力空間における量子化された入力値をラベル集合に変換し、そのラベルに対応した荷重表を参照し、その参照した荷重の総和を出力値とする。

具体的な例として、1入力1出力、荷重表数3の場合をFig.1に示す。実際の入力値に対し量子化を行うことで $s = 7$ が得られたとする。それに対応したラベル集合 $m^* = \{C, H, K\}$ へ変換され、このラベル集合の各要素で表わされるそれぞれの荷重表から荷重 $a^* = \{6, 4, 8\}$ を参照し、これらの総和 $\sum a^* = 18$ を出力値とする。このとき、教師信号20と出力値との差2を荷重表の数3で割った値、すなわち $2/3$ を参照した荷重表の各要素それぞれに加え、荷重表を修正する。

入力空間内の2つの入力値の距離が近い場合（例えば、Fig.2の $s = 7$ における $s = 6$ または $s = 8$ ）、一部重複したラベル集合が得られる。距離が遠くなるにつれて重複度は減少し、一定距離以上離れると全て異なったアドレスを参照する。これによ

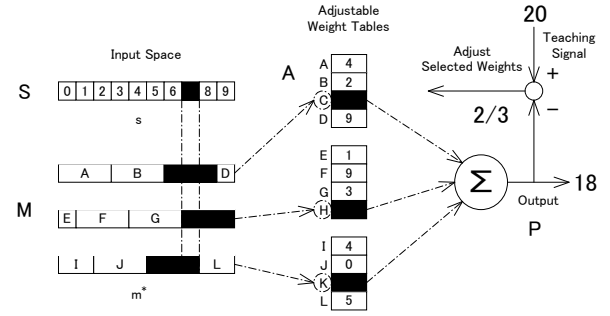


Fig. 1: CMAC model with one input, one output and three weight tables.

り、CMACは非線形性の強いモデルを少ない学習回数で近似することができる。また、CMACでは出力計算および荷重の修正において荷重表の一部のみを参照するため、比較的少ない時間で学習が可能である。

2.2 PID 制御への適用

本節では、CMACをPID制御に適用する方法³⁾について説明する。Fig.2にCMAC-PID制御系の概略図を示す。次式で与えられるPID制御則を考える。

$$\Delta u(t) = K_I e(t) - K_P \Delta y(t) - K_D \Delta^2 y(t) \quad (1)$$

ここで、 $u(t)$ は制御入力、 $y(t)$ はシステム出力である。 Δ は差分演算子を表し、 $\Delta = 1 - z^{-1}$ で定義される。 z^{-1} は時間遅延演算子を表し、 $z^{-1}u(t) = u(t-1)$ を意味する。式(1)はいわゆる速度型PID制御則(I-PD制御則)である。CMACの出力となるのは、このPIDパラメータ K_P, K_I, K_D であり、各パラメータを出力する3つのCMACを用意する。また、CMACへの入力信号は、ステップ状で与えられる目標値 $w(t)$ 、制御誤差 $e(t), e(t-1)$ の3変量とする。ここで、 $e(t)$ は次式で表される。

$$e(t) = w(t) - y(t) \quad (2)$$

参照モデルの出力 $y_m(t)$ と制御対象の出力の差が小さくなるようなPIDパラメータを出力するよう荷重を修正し学習を行う。そこで次の評価規範を考える。

$$J = \frac{1}{2} \varepsilon^2(t) = \frac{1}{2} \{y_m(t) - y(t)\}^2 \quad (3)$$

式(3)で表わされる評価規範を減少させるように、

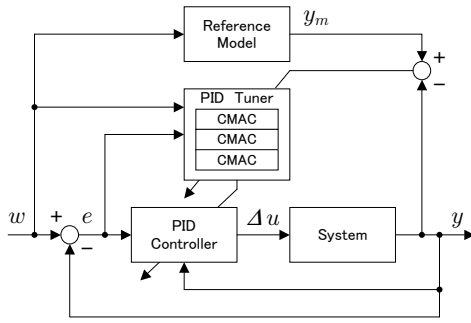


Fig. 2: Block diagram of CMAC-PID controller.

次式に示す勾配に基づいて学習を行う。

$$\frac{\partial J}{\partial K_P} = \varepsilon(t) \Delta y(t) \frac{\partial y(t)}{\partial u(t-1)} \quad (4)$$

$$\frac{\partial J}{\partial K_I} = -\varepsilon(t) e(t) \frac{\partial y(t)}{\partial u(t-1)} \quad (5)$$

$$\frac{\partial J}{\partial K_D} = \varepsilon(t) \Delta^2 y(t) \frac{\partial y(t)}{\partial u(t-1)} \quad (6)$$

ここで、 $\partial y(t)/\partial u(t-1)$ はシステムヤコビアンである。

CMAC は入力をラベルに変換し、ラベルに基づき参照した荷重の合計を出力とする。このとき、参照した荷重を次式により更新することで学習を行う。

$$W_{P,h}^{\text{new}}(t) = W_{P,h}^{\text{old}}(t) - g_P \frac{\partial J}{\partial K_P} \frac{1}{N_w} \quad (7)$$

$$W_{I,h}^{\text{new}}(t) = W_{I,h}^{\text{old}}(t) - g_I \frac{\partial J}{\partial K_I} \frac{1}{N_w} \quad (8)$$

$$W_{D,h}^{\text{new}}(t) = W_{D,h}^{\text{old}}(t) - g_D \frac{\partial J}{\partial K_D} \frac{1}{N_w} \quad (9)$$

ここで、 W^{old} は参照した荷重であり、 W^{new} はそれを更新した荷重である。また、 $h = 1, 2, \dots, N_w$ で、 N_w は参照する荷重の総数、つまり荷重表の数を表す。さらに、 g_P, g_I, g_D は荷重修正の割合を決定する学習係数を表わしている。

CMAC-PID 制御では、学習を行う前に適切な初期荷重表を作成する必要がある、オーバーシュートがない過制動な応答波形が得られる比較的低感度の PID パラメータを初期値として選べばよいことが分かっている。しかし、式 (4)~(6) からわかるように、定常状態では各勾配量は小さく、荷重表の修正量はほぼ 0 である。また、学習回数を増やすと過渡状態において不安定な挙動を示し、過学習という結果を生じる。つまり、少ない学習回数で得られた荷重表では、所望の定常特性が得られないことを意味する。そこで、過渡特性と定常

特性を同時に満足する CMAC-PID 制御則について次節にて考察する。

3. 定常性能を考慮した CMAC-PID 制御系の設計

3.1 概要

石油精製プロセスなどのプロセス産業界では、その運用状況からわかるように常に安定した稼働状態が必要不可欠である。また、コスト削減や省エネルギー化についても考慮する必要がある。そのため、システムの稼働データである制御誤差と制御差分入力の両方を用いて、PID 制御系の能力が十分に発揮されているかどうか評価・監視することは非常に有用である。過渡状態での応答波形による制御性能を改善する手法については数多く研究されているが、それらの手法を用いて得られる PID パラメータを使用しても、定常状態において満足する制御性能が得られるとは限らない。そこで、定常状態で所望の制御性能を得るために、次のステップで求めていく。

- 1° 定常状態のシミュレーションデータを用いてシステム同定を行い、システムのパラメータを推定する。
- 2° 所望の定常特性が得られるような PID パラメータを GPC 制御則を用いて調整する。

以上の機能を有する新しい CMAC-PID 制御系について、次節にて考察していく。

3.2 定常状態における制御性能

まず、式 (24) の詳細モデルに対応した次の離散時間モデルを考える。

$$\alpha(z^{-1})y(t) = z^{-(d+1)}\beta(z^{-1})u(t) + \xi(t)/\Delta \quad (10)$$

$$\left. \begin{aligned} \alpha(z^{-1}) &= 1 + \alpha_1 z^{-1} \\ \beta(z^{-1}) &= \beta_0 + \beta_1 z^{-1} \end{aligned} \right\} \quad (11)$$

ここで d は離散時間におけるむだ時間を表し、 $\xi(t)$ は平均 0、分散 σ_ξ^2 のガウス性白色雑音とする。

このとき、システムが定常状態であるときの制御誤差 $e(t)$ と制御入力 $\Delta u(t)$ の分散は、 H_2 ノルムを用いて次式で計算できる⁷⁾。

$$\hat{\sigma}_e^2 = E[\{e(t)\}^2] = \left\| -\frac{1}{T(z^{-1})} \right\|_2^2 \cdot \sigma_\xi^2 \quad (12)$$

$$\hat{\sigma}_u^2 = E[\{\Delta u(t)\}^2] = \left\| -\frac{L(z^{-1})}{T(z^{-1})} \right\|_2^2 \cdot \sigma_\xi^2 \quad (13)$$

ただし, $T(z^{-1})$ は次式で定義している.

$$T(z^{-1}) = \Delta\alpha(z^{-1}) + z^{-(d+1)}\beta(z^{-1})L(z^{-1}) \quad (14)$$

また, 実際の稼働データから得られる制御誤差の分散 $\sigma_e^2(t)$, 制御差分入力の分散 $\sigma_u^2(t)$ は, それぞれデータ長 M_e, M_u の二乗平均で求められ, それぞれ $\hat{\sigma}_e^2, \hat{\sigma}_u^2$ の周りに分布する.

そこで, 以下の制御性能評価指数を定義する⁸⁾.

$$\eta_e(t) = \hat{\sigma}_e^2 / \sigma_e^2(t), \quad \eta_u(t) = \hat{\sigma}_u^2 / \sigma_u^2(t) \quad (15)$$

ここで, $\eta_e(t), \eta_u(t)$ は 1 に近いほど制御性能がよく, 0 または無限大へ近づくにつれ制御性能が悪いといえる.

3.3 システム同定

式 (11) の係数 $\alpha_1, \beta_0, \beta_1$ の推定値 $\hat{\alpha}_1(t), \hat{\beta}_0(t), \hat{\beta}_1(t)$ は, 次の逐次最小二乗法により求めることができる.

$$\begin{aligned} \hat{\theta}(t) &= \hat{\theta}(t-1) \\ &+ \frac{\mathbf{F}(t-1)\boldsymbol{\psi}(t-1)}{\rho + \boldsymbol{\psi}^T(t-1)\mathbf{F}(t-1)\boldsymbol{\psi}(t-1)} \cdot \varepsilon_p(t) \end{aligned} \quad (16)$$

$$\begin{aligned} \mathbf{F}(t) &= \frac{1}{\rho} \mathbf{F}(t-1) \\ &- \frac{\mathbf{F}(t-1)\boldsymbol{\psi}(t-1)\boldsymbol{\psi}^T(t-1)\mathbf{F}(t-1)}{1 + \boldsymbol{\psi}^T(t-1)\mathbf{F}(t-1)\boldsymbol{\psi}(t-1)} \end{aligned} \quad (17)$$

$$\varepsilon_p(t) = \Delta y(t) - \hat{\theta}^T(t-1)\boldsymbol{\psi}(t-1) \quad (18)$$

$$\hat{\theta}(t) = [\hat{\alpha}_1(t), \hat{\beta}_0(t), \hat{\beta}_1(t)]^T \quad (19)$$

$$\begin{aligned} \boldsymbol{\psi}(t-1) &= [-\Delta y(t-1), \\ &\Delta u(t-d-1), \Delta u(t-d-2)]^T \end{aligned} \quad (20)$$

ここで, ρ は忘却係数 (forgetting factor: $0 < \rho \leq 1$) を表し, $\varepsilon_p(t)$ は予測誤差を表している. このとき, システムパラメータの推定値 $\hat{K}, \hat{T}, \hat{L}$ は, 次式により計算できる.

$$\hat{T}(t) = -\frac{T_s}{\log_e(-\hat{\alpha}_1(t))} \quad (21)$$

$$\hat{K}(t) = \frac{\hat{\beta}_0(t) + \hat{\beta}_1(t)}{1 + \hat{\alpha}_1(t)} \quad (22)$$

$$\hat{L}(t) = \left\{ \frac{\hat{\beta}_1(t)}{\hat{\beta}_0(t) + \hat{\beta}_1(t)} + d \right\} \cdot T_s \quad (23)$$

3.4 モデル予測型 PID 制御

対象とする制御システムの記述モデル (詳細モデル) を「一次遅れ+むだ時間」系として与える.

$$G(s) = \frac{K}{1 + Ts} e^{-Ls} \quad (24)$$

ここで, システムパラメータ K, T, L はそれぞれシステムゲイン, 時定数, むだ時間を表している. さらに, むだ時間を一次遅れ系で近似し, 次式で表わされる離散時間伝達関数モデルを用いる.

$$A(z^{-1})y(t) = B(z^{-1})u(t-1) \quad (25)$$

$$\left. \begin{aligned} A(z^{-1}) &= 1 + a_1 z^{-1} + a_2 z^{-2} \\ B(z^{-1}) &= b_0 + b_1 z^{-1} \end{aligned} \right\} \quad (26)$$

次に, 式 (1) で与えられる PID 制御則を考える. 以下の考察を簡単にするため, 式 (1) を次式として書き換える.

$$L(z^{-1})y(t) - L(1)w(t) + \Delta u(t) = 0 \quad (27)$$

$$\begin{aligned} L(z^{-1}) &= K_P + K_I + K_D \\ &- (K_P + 2K_D)z^{-1} + K_D z^{-2} \end{aligned} \quad (28)$$

である. ここでは, GPC 則に基づいて PID パラメータ (K_P, K_I, K_D) を調整する⁹⁾. GPC 則では次の評価関数を最小化することを考える.

$$\begin{aligned} J(t) &= \mathbf{E} \left[\sum_{j=1}^N \{y(t+j) - w(t)\}^2 \right. \\ &\quad \left. + \lambda \sum_{j=1}^N \{\Delta u(t+j-1)\}^2 \right] \end{aligned} \quad (29)$$

ここで, N は予測区間, λ はシステム入力の増分に対する重み係数, $\mathbf{E}[\cdot]$ は期待値を表している. 式 (29) を最小にする制御則は次式のように得られる.

$$\begin{aligned} \sum_{j=1}^N \{p_j F_j(z^{-1})\} y(t) - \sum_{j=1}^N p_j w(t) \\ + \left\{ 1 + z^{-1} \sum_{j=1}^N p_j s_j \right\} \Delta u(t) = 0 \end{aligned} \quad (30)$$

ここで, $F_j(z^{-1})$ と s_j は, 次の Diophantine 方程式を解くことで得られる.

$$1 = \Delta A(z^{-1})E_j(z^{-1}) + z^{-j}F_j(z^{-1}) \quad (31)$$

$$E_j(z^{-1})B(z^{-1}) = R_j(z^{-1}) + z^{-j}s_j \quad (32)$$

ここで,

$$E_j(z^{-1}) = 1 + e_1 z^{-1} + \cdots + e_{j-1} z^{-(j-1)} \quad (33)$$

$$F_j(z^{-1}) = f_{j,0} + f_{j,1} z^{-1} + f_{j,2} z^{-2} \quad (34)$$

$$R_j(z^{-1}) = r_0 + r_1 z^{-1} + \cdots + r_{j-1} z^{-(j-1)} \quad (35)$$

であり, p_j は次式で与えられる.

$$\begin{aligned} [p_1, p_2, \dots, p_N] \\ = [1, 0, \dots, 0] (\mathbf{R}^T \mathbf{R} + \lambda \mathbf{I})^{-1} \mathbf{R}^T \end{aligned} \quad (36)$$

ここで、行列 $\mathbf{R}$ は式 (35) の係数からなる行列で、次式で表される。

$$\mathbf{R} = \begin{bmatrix} r_0 & & & & \\ r_1 & r_0 & & & \\ \vdots & & \ddots & & \\ r_{N-1} & r_{N-2} & \cdots & r_0 & \end{bmatrix} \mathbf{0} \quad (37)$$

いま、式 (30) 左辺第 3 項を定常ゲインに置き換え、次式を得る。

$$\tilde{F}(z^{-1})y(t) - \tilde{F}(1)w(t) + \Delta u(t) = 0 \quad (38)$$

$$\begin{aligned} \tilde{F}(z^{-1}) &= \frac{\sum_{j=1}^N p_j F_j(z^{-1})}{1 + \sum_{j=1}^N p_j s_j} \\ &= \tilde{f}_0 + \tilde{f}_1 z^{-1} + \tilde{f}_2 z^{-2} \end{aligned} \quad (39)$$

である。式 (27) と式 (38) を比較することにより、近似的に GPC 則と PID 制御則が等価となる。

$$K_P = -(\tilde{f}_1 + 2\tilde{f}_2) \quad (40)$$

$$K_I = \tilde{f}_0 + \tilde{f}_1 + \tilde{f}_2 \quad (41)$$

$$K_D = \tilde{f}_2 \quad (42)$$

以上により、GPC 則に基づいて PID パラメータを調整することができる。GPC 則の設計パラメータである予測区間 N は、実用的には時定数 T とむだ時間 L の和に相当するステップ数あたりに設定するのが良いとされている⁹⁾。重み係数 λ は定常状態における制御性能の観点から、閉ループ系におけるシステムのステップ応答出力波形の最大行過ぎ量 $M_0[\%]$ に基づいて調整する¹⁰⁾。

3.5 荷重表の更新方法

定常状態で望ましい性能が得られる PID パラメータ ($\hat{K}_P, \hat{K}_I, \hat{K}_D$) が求められたとき、CMAC-PID 制御器の荷重表において現在の目標値 $w(t)$ および $e(t) = 0, e(t-1) = 0$ に相当する荷重値を次式により変更する。

$$W_{P,h} = \hat{K}_P / N_w, \quad (h = 1, 2, \dots, N_w) \quad (43)$$

$$W_{I,h} = \hat{K}_I / N_w, \quad (h = 1, 2, \dots, N_w) \quad (44)$$

$$W_{D,h} = \hat{K}_D / N_w, \quad (h = 1, 2, \dots, N_w) \quad (45)$$

ここで、 $W_{P,h}, W_{I,h}, W_{D,h}$ はオフラインで学習させた後に得られる荷重表を表している。前述したとおり、CMAC は入力値に応じて参照する荷重値が異なる。つまり、式 (43)–(45) で定常状態での荷重値を書き換えることにより、過渡特性に影響を及ぼすことはなく目標値に近づくにつれ定常特性が改善されることを意味する。これにより、過渡状態と定常状態の両面において所望の制御特性となるコントローラが得られる。

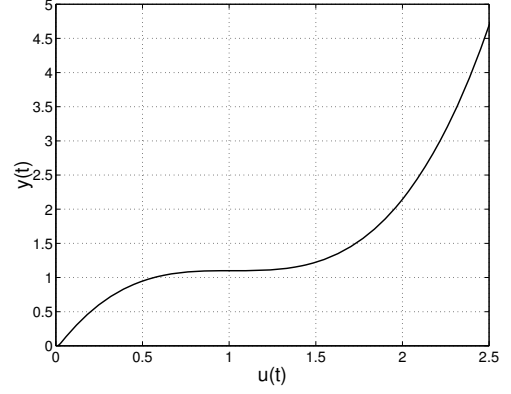


Fig. 3: Static property of hammerstein system.

Table 1: User-specified parameters

CMAC	Width of quantization	0.25
	Number of weight tables	$N_w = 4$
	Learning coefficient	$g_P = 3 \times 10^{-3}$ $g_I = 10^{-4}$ $g_D = 10^{-1}$
GPC	Prediction horizon	$N = (T + L)/T_s$
System Identification	Forgetting factor	$\rho = 1.00$
	Data length	300

3.6 定常性能を考慮した CMAC-PID コントローラ

これまでに考察した手続きをまとめ、以下のようにアルゴリズムを構築する。

[アルゴリズム]

- 1° 荷重表を初期化する。
- 2° オフラインにて CMAC-PID コントローラの荷重表を学習させる。
- 3° シミュレーションデータから式 (16)–(20) を用いてシステム同定を行う。
- 4° 制御性能評価に基づいて λ を求め、GPC-PID 制御則に基づいて PID パラメータを求める (式 (40)–(42))。
- 5° CMAC-PID コントローラの荷重表を再調整する (式 (43)–(45))。
- 6° オンラインで稼働させる。

4. 数値例

本論文で提案したアルゴリズムの有効性をシミュレーションにより検証する。

制御対象として、以下の式で表わされる Hammerstein モデルを考える。

$$\left. \begin{aligned} y(t) &= 0.6y(t-1) - 0.1y(t-2) \\ &\quad + 1.2x(t-d-1) - 0.1x(t-d-2) + \frac{\xi(t)}{\Delta} \\ x(t) &= 1.5u(t) - 1.5u^2(t) + 0.5u^3(t) \end{aligned} \right\} \quad (46)$$

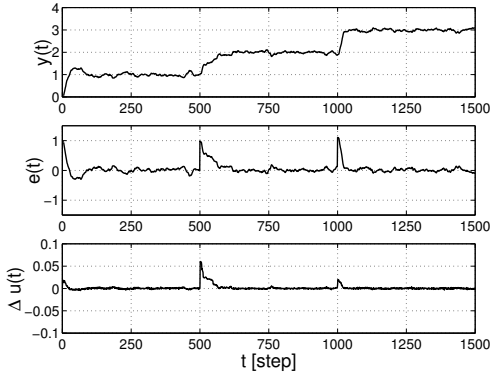


Fig. 4: Learning result using the conventional CMAC-PID controller.

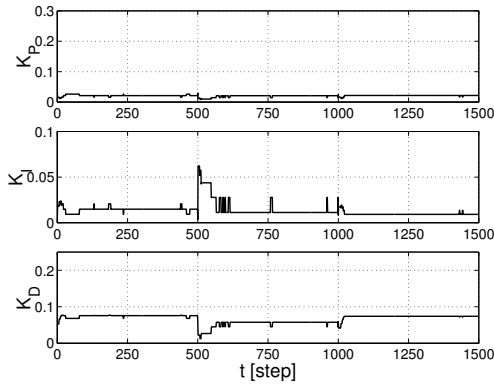


Fig. 5: Trajectories of PID parameters corresponding to Fig.4.

ここで、 ξ は平均0、分散 0.01^2 のガウス性白色雑音を用いた。このモデルの静特性を Fig.3 に示す。また、目標値 $w(t)$ は以下のように設定した。

$$w(t) = \begin{cases} 1.0 & (0 \leq t < 800) \\ 2.0 & (800 \leq t < 1600) \\ 3.0 & (1600 \leq t \leq 2400) \end{cases} \quad (47)$$

さらに、参照モデルとして次式を用いた。

$$y_m(t) = 0.8y_m(t-1) + 0.2w(t-d-1) \quad (48)$$

また、サンプリング時間を $T_s = 1.0[\text{sec}]$ とし、むだ時間は既知であると仮定し $d = 4$ とした。さらに、各種パラメータを Table 1 に示すよう設定した。CMAC の初期 PID パラメータは、Fig.3 の $u = 3$ におけるシステムパラメータをシステム同定により導出し、CHR 法により求められた値として設定した。GPC 制御則の λ は $M_0 = 5[\%]$ となるように設定した。制御性能評価指数に関して、 $M_e = M_u = 300$ とし、 $50[\text{step}]$ ごとに求めた。

上記のパラメータによりシミュレーションを行った。まず従来法である CMAC-PID 制御を用いてオ

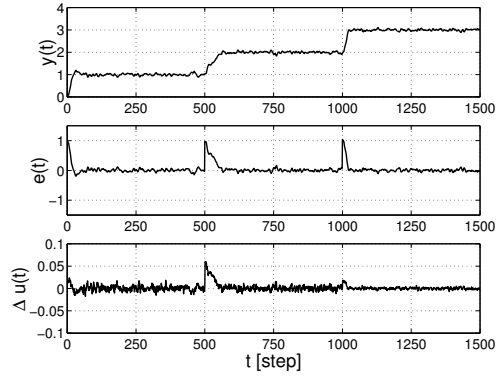


Fig. 6: Control result using the proposed control scheme.

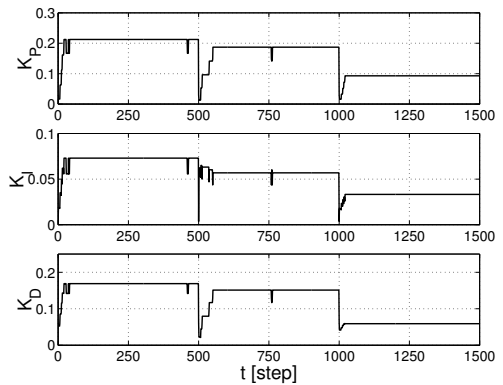


Fig. 7: Trajectories of PID parameters corresponding to Fig.6.

フラインにて荷重表の学習を行った。20回の学習を行ったときの制御結果を Fig.4 に、そのときの PID ゲインの時間推移を Fig.5 にそれぞれ示す。次にオフラインにて作成された荷重表を用いて提案した手法を適用した。その結果を Fig.6 に、そのときの PID ゲインの時間推移を Fig.7 に、制御性能評価指数の時間推移を Fig.8 にそれぞれ示す。Fig.8 より、制御性能指数が 1 付近で落ち着いていることがわかる。このことからシステムパラメータが適切に推定され、所望の制御性能が得られるように PID パラメータが調整されていることがわかる。また、誤差信号と差分入力信号との間にはトレードオフの関係があるためこれら 2 つの値を同時に小さくすることは難しいが、Fig.4 と Fig.6 より本手法により定常状態における誤差信号の分散を低く抑えられていることがわかる。この結果から、本手法を用いることにより従来法による過渡特性を維持したまま、定常特性を改善させることができることが確かめられた。

さらに比較のため、セルフチューニング GPC-PID 制御系により同様のシミュレーションを行っ

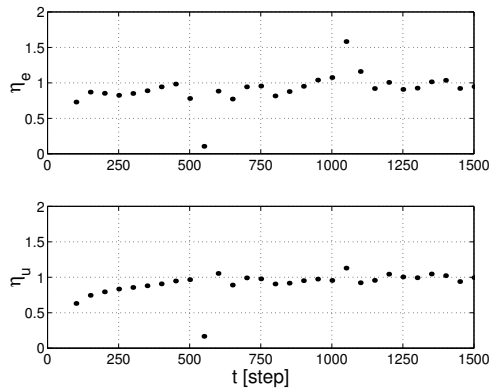


Fig. 8: Trajectories of CPA indices using the proposed scheme.

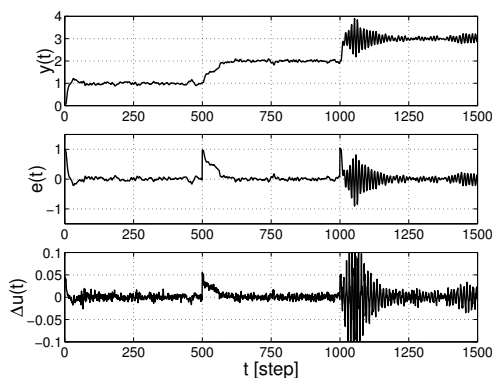


Fig. 9: Transition response result using self-tuning GPC-PID controller.

た. ここで, GPC 則で用いられる設定パラメータを $N = 20$, $\lambda = 300$ と固定した. この結果を Fig.9 に示す. 過渡状態における速応性, 定常状態における追従性の両面において本手法の方が優れていることがわかる. これより, 本手法の有効性が確かめられた.

5. まとめ

本論文では, CMAC の学習後に得られる荷重表を再調整し, 定常状態において所望の制御性能が得られる CMAC-PID 制御法について提案した. 本手法の特徴は, オフラインにおいて非線形システムを学習させることで, 比較的少ない学習回数で所望の過渡特性が得られる. さらに, 定常状態に対応する荷重表を修正することにより所望の定常特性が得られる. これにより, 従来の CMAC-PID コントローラの問題点であった定常状態における制御性能を改善することができ, 特に定常特性を重視するプロセス制御に適した制御系を構築することができる. コンピュータシミュレーションにより, 非線形システムにおいて過渡状態および定

常状態の両面から所望の制御特性が得られることを確認した. しかし, CMAC の特徴から過渡特性が特別に優れているとは言えず, 考察を重ねていく必要がある. また, 実際のプラントではそのシステムの特徴が時々刻々と変化するため, これに対応させていくことも今後の課題である.

参考文献

- 1) 須田信英: PID 制御, 朝倉出版 (1992).
- 2) J.S.Albus: A New Approach to Manipulator Control: The Cerebellar Model Articulation Controller (CMAC), Trans. ASME, J. of Dynamic Systems, Measurement, and Control, Vol. 97, No. 9, pp. 220–227 (1975).
- 3) 黒住亮太, 山本透: 小脳演算モデルを用いたインテリジェント PID 制御系の一設計, 電気学会論文誌, Vol. 125C, No. 4, pp. 607–615 (2005).
- 4) T.J.Harris: Assessment of Control Loop Performance, Canadian Journal of Chemical Engineering, Vol. 67, No. 5, pp. 856–861 (1989).
- 5) B.Huang and S.L.Shah: Performance Assessment of Control Loops: Theory and Applications, Springer-Verlag, London (1999).
- 6) B.Huang: A pragmatic approach towards assessment of control loop performance, Int. J. Adapt. Control Signal Process., Vol. 17, pp. 589–608 (2003).
- 7) 山本透: モデリング性能評価に基づくパフォーマンス・アダプティブ PID 制御系の一設計, 電気学会論文誌, Vol. 127C, No. 12, pp. 2101–2108 (2007).
- 8) L. Desborough and T. Harris: Performance Assessment Measures for Univariate Feedback Control, Canadian Journal of Chemical Engineering, Vol. 70, No. 6, pp. 1186–1197 (1992).
- 9) 増田士朗, 山本透, 大嶋正裕: モデル予測制御 III 一般化予測制御 (GPC) とその周辺, システム/制御/情報, Vol. 46, No. 9, pp. 578–584 (2002).
- 10) 八木秀幸, 下西二郎, 山本透, 雛元孝夫: モデル予測機能を有するパフォーマンス駆動型 PID 制御系の一設計, 計測自動制御学会論文集, Vol. 44, No. 10, pp. 809–818 (2008).

