PID 制御器のデータベース駆動型チューニング†

太 田 純*・山 本 茂**

Database-Driven Tuning of PID Controllers†

Jun OHTA* and Shigeru YAMAMOTO**

In this paper, we propose a tuning method for PID controllers based on a database-driven function estimation technique. The basic idea of the proposed method is to build a local weighted model from all observed data stored in a database so that a PID parameter meets given design specifications. With several simulations for process control problems, we illustrate the usefulness of the proposed method.

Key Words: Model-on-Demand, Just-in-Time modeling, Lazy Learning, PID

1. はじめに

コントローラのパラメータを制御対象の特性や制御仕様に応じて適切に調整（チューニング）することは重要である。産業界で広く用いられている PID 制御器において、このチューニングがコントローラ側で自動的に行なわれるセルフチューニング PID は、これまでに多種多様な手法が提案され研究が進められてきた。その主たる目的は (a) 制御パラメータ調整作業の自動化, (b) 調整時間の短縮, (c) オペレータやエンジニアの省力化, (d) 調整結果の個人差の解消などである¹⁾。一般にセルフチューニング機能を実現するためには、制御対象の動特性の同定とその結果に基づく PID パラメータの決定とを組み合わせる必要があると考えられる。本論文では、このチューニング過程を一種の非線形関数として捉え、その関数を推定することで補償器の調整を行なう手法を提案する。

非線形関数の推定は、現在まで広く研究されているが、最近のコンピュータ技術の発展に伴い、大量のデータベースを活用した手法が注目されている。それらは、Model-on-Demand (MoD)²⁾、Just-In-Time モデリング³⁾、Lazy learning⁴⁾ と呼ばれる手法である。これらは、対象から得られた大量の観測データをデータベースに蓄えておき、要求があったときのみ必要なデータをデータベースから検索し、そのデータから局所モデルを構築する方法である。これらの

モデリングは、人工知能、マシン学習、ロボット制御など広範な分野で応用されている^{5)~7)}。

本論文で提案する手法では、閉ループ制御時に得られた応答波形の特徴量をデータベースに蓄えておき、MoD などのデータベースを活用した手法によってパラメータ調整に相当する非線形関数を推定することで、所望の制御性能を得ることのできる PID パラメータを推定することを目的とする。データベース中に蓄えるデータは過去の試行錯誤の結果得られたものや、新たなチューニング作業によって生じたもの、通常の操業で得られるものなどである。また本稿では、いわゆる次元の呪いを避けるために、一般に広範囲の制御対象に用いられているパラメータ数が少ない 1 自由度の PID 補償器に限定して述べる。有用性を検証するため一次遅れ+むだ時間系と二次遅れ系が混在するプロセス制御系と電気油圧サーボ系に対するシミュレーションを行ない、PID パラメータの自動抽出によって、それぞれ所望の制御性能が得られることを確認する。

2. Model-on-Demand

データベースを活用した非線形関数の推定法として、以下に Model-on-Demand 手法の説明をする。

非線形関数

$$Y_i = m(X_i) + \epsilon_i \quad (1)$$

を考える。 X_i が入力で、 Y_i はノイズ ϵ_i が加わった出力である。ある新しい入力 x が与えられたとき十分に大きな観測データセット $\{X_i, Y_i\}_{i=1}^N$ を用いて、非線形関数 $m(x)$ の推定値を与えることである。

MoD の基本的な考え方は、すべての観測値をデータベースに蓄え、そして必要が生じたときにその要求に応じて、要求点の近傍を選び出し、その近傍から局所モデルを作るというものである。

† 第 3 回適応学習シンポジウムで発表 (2003・3)

* 大阪大学大学院基礎工学研究科システム人間系専攻
豊中市待兼山町 1-3

** 大阪大学大学院基礎工学研究科システム創成専攻
豊中市待兼山町 1-3

Department of Systems and Human Science, Osaka University, 1-3 Machikaneyama, Toyonaka, Osaka

** Department of Systems Innovation, Osaka University, 1-3 Machikaneyama, Toyonaka, Osaka

要求点の近傍の選出は、まず、各データの入力ベクトルと要求点のベクトル間のノルムにより距離を測り、最適なサンプル数をクロスバリデーションや赤池の FPE (Final Prediction Error; 最終予測誤差) などの規範を用いて決定する。局所モデルの同定は、LWR (Locally Weighted Regression; 重み付け局所回帰法) や LWA (Locally Weighted Average; 重み付け線形平均法) などがある。重みとしては、以下の性質を満たすカーネル関数 $K(u)$ が用いられる。

$$\int K(u)du = 1, \quad \int uK(u)du = 0 \quad (2)$$

一般的には、Gaussian Kernel や Tricube Kernel などのカーネルが用いられる。

データベース $\{X_i, Y_i\}_{i=1}^N$ が与えられているときの代表的な MoD のアルゴリズムをつぎに示す。

MoD アルゴリズム

入力: データベース $\{X_i, Y_i\}_{i=1}^N$ および要求点 X_q 。

出力: 要求点 X_q に対する出力 Y_q の推定値 $\hat{Y}_q$ 。

Step1: データベース中のすべてのデータを入力ベクトル $X_i (i=1, \dots, N)$ と要求点 X_q との距離 $d(X_i, X_q)$ が小さいものから順に並び替える。距離 $d(X_i, X_q)$ は、

$$d(X_i, X_q) = \sqrt{(X_i - X_q)^T M (X_i - X_q)} \quad (3)$$

で与えられ、対角行列 $M \in \mathbb{R}^{2n \times 2n}$ は、対角成分に入力ベクトル X の共分散行列の逆行列の対角成分をもつ。

Step2: 要求点 X_q の最適な近傍 $\Omega_{k_{\text{opt}}}$ を赤池の FPE を用いて求める。

$$k_{\text{opt}} = \arg \min_k V_k(K(u), \Omega_k) \frac{k+n}{k-n} \quad (4)$$

$$\Omega_{k_{\text{opt}}} = \{X_i, Y_i\}_{i=1}^{k_{\text{opt}}} \quad (5)$$

ただし、 V_k は損失関数で、データベースに記録されている i 番目の出力の実績値 Y_i とカーネル関数 $K(u)$ を用いて、局所重み付平均法によって与えられる予測値 $\hat{Y}_i$ の二乗誤差平均

$$V_k(K(u), \Omega_k) = \frac{1}{k} \sum_{i=1}^k (Y_i - \hat{Y}_i)^2 \quad (6)$$

$$\hat{Y}_i = \sum_{i=1}^k K\left(\frac{d(X_i, X_q)}{h}\right) Y_i \quad (7)$$

$$h = \max\{d(X_i, X_q) \mid i \in \Omega_k\} \quad (8)$$

とする。

Step3: 局所重み付平均法より Y_q の推定値 $\hat{Y}_q$ を求める。

$$\hat{Y}_q = \sum_{i=1}^{k_{\text{opt}}} K\left(\frac{d(X_i, X_q)}{h_{\text{opt}}}\right) Y_i \quad (9)$$

$$h_{\text{opt}} = \max\{d(X_i, X_q) \mid i \in \Omega_{k_{\text{opt}}}\} \quad (10)$$

3. データベース駆動型 PID 制御パラメータ抽出アルゴリズム

3.1 PID パラメータと制御結果の関係

制御対象 $P(s)$ に対して所望の制御結果を得るような PID 制御器のパラメータを抽出する問題を考える。今、制御対象 $P(s)$ に対して、任意の PID パラメータ $u_i = \{K_{Pi}, K_{Ii}, K_{Di}\}$ とそれを用いて得られる制御結果の特微量 r_i との間に次式の関係が成り立つような非線形関数 $f(\cdot)$ が存在すると仮定する。

$$r_i = f(P(s), u_i) + \epsilon_i \quad (11)$$

ここで、 ϵ_i はノイズである。このとき、(11) 式の関係から、 $f(P(s), \cdot)$ の逆関数 $f^{-1}(P(s), \cdot)$ を推定することにより、目標とする特微量 r_q に対する PID パラメータ $u_q = \{K_{Pq}, K_{Iq}, K_{Dq}\}$ を $u_q = \hat{f}^{-1}(P(s), r_q)$ によって求めることは、制御対象 $P(s)$ が一般に未知であるため困難である。したがって、観測できるデータを用いて $P(s)$ の特性を表すベクトル s_i に置き換えることを考える。実際のチューニング過程において、オペレータは、PID 制御器のパラメータ $u_i = \{K_{Pi}, K_{Ii}, K_{Di}\}$ とそれを用いて得られる制御結果の特微量 r_i から制御対象の特性を把握して調整を行なうことから、 u_i と r_i は制御対象の特性を含んだデータといえる。そこで s_i を

$$s_i = [u_i, r_i] \quad (12)$$

とし、新たに調整した PID パラメータを $u_{i+1} = \{K_{Pi+1}, K_{Ii+1}, K_{Di+1}\}$ とすると、(11) 式と (12) 式により制御結果 r_{i+1} は次式によって関係付けられる。

$$r_{i+1} = f(s_i, u_{i+1}) + \epsilon_{i+1} \quad (13)$$

したがって、 $f(s, \cdot)$ の逆関数を推定することができれば、所望の制御結果が r_q となるような PID パラメータ u_q は $u_q = f^{-1}(s_i, r_q)$ によって求めることができる。新たに

$$x_{i,q} = [s_i, r_q] \quad (14)$$

とすると

$$u_q = f^{-1}(x_{i,q}) \quad (15)$$

となる。

3.2 PID パラメータ抽出手順

データベース $D(N) = \{s_i, r_{i+1}, u_{i+1}\}_{i=1}^N = \{x_{i,i+1}, u_{i+1}\}_{i=1}^N$ が与えられたとき、目標の制御結果となるような PID パラメータを以下の手順に従い抽出する。

PID パラメータ抽出アルゴリズム

入力: データベース $D(N)$ 、現在使用中の PID パラメータ u_k とそれによる制御結果 r_k および目標とする制御結果 r_q 。

出力: 目標とする制御結果となる PID パラメータ u_q の推定値 $\hat{u}_q$ 、データベース $D(N+1) = \{x_{i,i+1}, u_{i+1}\}_{i=1}^{N+1}$ 。

Table 1 Plants and control specifications

plant	parameters				control specifications		
	K	T	L	ζ	A_{inf}	A_p	T_s
$P_1(s)$	1	10	1	-	1	$\leq 10\%$	≤ 20 sec
$P_2(s)$	1	10	5	-	1	$\leq 10\%$	≤ 25 sec
$P_3(s)$	1	10	10	-	1	$\leq 10\%$	≤ 30 sec
$P_4(s)$	1	10	15	-	1	$\leq 10\%$	≤ 35 sec
$P_5(s)$	1	1	-	0.3	1	$\leq 10\%$	≤ 10 sec
$P_6(s)$	1	5	-	0.3	1	$\leq 10\%$	≤ 15 sec
$P_7(s)$	1	10	-	0.3	1	$\leq 10\%$	≤ 20 sec
$P_8(s)$	1	15	-	0.3	1	$\leq 10\%$	≤ 25 sec

Step1: $s_k = [u_k, r_k]$, $x_{k,q} = [s_k, r_q]$ を求める.

Step2: MoD アルゴリズムを用い PID パラメータの推定値 $\hat{u}_q$ を求める.

$\hat{u}_q = \text{MoD アルゴリズム } (D(N), x_{k,q})$

Step3: PID パラメータ $\hat{u}_q$ を用いて制御を実行し, 得られた制御結果を r_q^* とする.

Step4: s_k , u_q , r_q^* をデータベースに蓄える.

$$x_{N+1, N+2} = [s_k, r_q^*], u_{i+1} = u_q \quad (16)$$

$$D(N+1) = \{x_{i,i+1}, u_{i+1}\}_{i=1}^{N+1} \quad (17)$$

4. 一次遅れ+むだ時間系と二次遅れ系が混在する場合

制御対象の特性が一次遅れ+むだ時間系と二次遅れ系で表わせる場合を考える. 以下に制御対象のモデルと制御目標を示す.

4.1 制御対象のモデルと制御目標

一次遅れ+むだ時間系を次式で与える.

$$P(s) = \frac{K}{1+Ts} e^{-sL} \quad (18)$$

また, 二次遅れ系を次式で与える.

$$P(s) = \frac{K}{T^2s^2 + 2\zeta Ts + 1} \quad (19)$$

また, PID 補償器 $C(s)$ は次式で与える.

$$C(s) = \frac{K_D s^2 + K_P s + K_I}{s} \quad (20)$$

ここでは, 一次遅れ+むだ時間系のプラントと二次遅れ系のプラントをそれぞれ 4 つ, 合計 8 個のプラントを用意した. 各プラントのパラメータとステップ応答における制御仕様を Table 1 に示す. 制御仕様の評価項目は, 定常値 A_{inf} , オーバーシュート A_p , 整定時間 T_s とする.

4.2 データベース

Table 1 の各制御対象 $P_i(s)$, $i = 1, \dots, 8$ に対して手動で調整を行ない, その過程で得られるデータをデータベースに蓄える. まず, PID パラメータベクトルの初期値を

$$u_1 = [K_{P1}, K_{I1}, K_{D1}] = [1 \ 0 \ 0]$$

として, 閉ループ系のステップ応答試験を行ない, 計測さ

Table 2 The number of data for the plant

plant	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	Total
Num. of data	18	18	26	15	24	32	33	23	189

Table 3 Inputs to the proposed algorithm

	s_1						reference		
	u_1			r_1			r_q		
	K_P	K_I	K_D	A_{inf}	$A_P(\%)$	$T_s(s)$	A_{inf}	A_P	T_s
P_1	1	0	0	0.5	0.00	18.4	1	0	15
P_2	1	0	0	0.5	5.01	23.1	1	0	20
P_3	1	0	0	0.5	20.9	44.9	1	0	25
P_4	1	0	0	0.5	34.5	69.2	1	0	30
P_5	1	0	0	0.5	50.5	12.0	1	0	5
P_6	1	0	0	0.5	50.0	60.3	1	0	10
P_7	1	0	0	0.5	50.0	120.6	1	0	15
P_8	1	0	0	0.5	50.0	180.9	1	0	20

Table 4 Estimated PID parameters and the control results

	PID parameter $\hat{u}_q$			results		
	K_P	K_I	K_D	A_{inf}	$A_P(\%)$	$T_s(s)$
P_1	1.800	0.200	0.228	1.00	0.92*	15.15
P_2	1.433	0.120	1.500	1.00	0.00	17.25
P_3	0.969	0.070	1.710	1.00	0.69	26.10
P_4	0.965	0.056	2.713	1.00	0.26	29.80
P_5	2.150	2.015	2.064	1.00	0.73	4.75
P_6	1.500	0.500	12.284	1.00	0.11	7.50
P_7	1.579	0.283	28.332	1.00	0.61	14.75
P_8	1.826	0.192	42.450	1.00	0.18	18.70

れた定常値 A_{inf} , オーバーシュート A_p , 整定時間 T_s を用いて

$$r_1 = [A_{inf}, A_p, T_s]$$

とする. これらの値から $s_1 = [u_1, r_1]$ を初期値としてさらにパラメータ調整を手動で行ない, その過程における PID パラメータと制御結果 (A_{inf}, A_p, T_s) と共にデータベースに蓄える. 各制御対象ごとにこの調整作業を数回繰り返してデータベースを構築した. 構築されたデータベース中のデータ数を Table 2 に示す.

4.3 数値例と結果

4.2 で構築したデータベースを使い, 各制御対象に対して制御仕様を満足する PID パラメータを抽出する. 各制御対象に対し初期の状態を表わすベクトル s_0 を求め, 目標とする制御仕様 r_q を与えアルゴリズムに入力した. 具体的な入力値を Table 3 に示す. 提案手法により得られた PID パラメータとそれに対する制御結果を Table 4 に示す. また, 各制御対象のステップ応答を Fig. 1 に示す. Fig. 1 では, 開ループ時および提案手法により抽出された PID パラメータを使用したときのステップ応答を示した. どの対象に対しても, 適用前の PID パラメータと制御結果および目標値を与えるだけで目標値に近い制御結果となるような PID パラメータが抽出されている.

Fig. 1 Step responses of the open loop system (dotted line) and the closed-loop system by using the estimated PID parameters (solid line)

Fig. 2 Electro-hydraulic servo system

5. 電気油圧サーボ系

ここでは、非線形の制御対象として Fig. 2 で表わされる電気油圧サーボ系を扱う（詳細は参考文献⁸⁾を参照）。

5.1 制御対象のモデル

操作量は電流指令値 i であり、制御量はシリンダ位置 y

とする。この制御対象のモデルが以下のように与えられているとする。理想のサーボ弁のスプール変位 x_s と電流指令値 i の関係は、以下の伝達関数によって与えられる。

$$x_s = \frac{\omega_s^2}{s^2 + 2\zeta\omega_s s + \omega_s^2} i \quad (21)$$

油室 1, 油室 2 への流量 q_1, q_2 は, $x_s \geq 0$ の場合,

$$q_1 = \text{sgn}(P_s - P_1) q_r \bar{x}_s \sqrt{\frac{|P_s - P_1|}{P_R}} \quad (22)$$

$$q_2 = -\text{sgn}(P_2 - P_T) q_r \bar{x}_s \sqrt{\frac{|P_2 - P_T|}{P_R}} \quad (23)$$

$x_s < 0$ の場合,

$$q_1 = -\text{sgn}(P_1 - P_T) q_r \bar{x}_s \sqrt{\frac{|P_1 - P_T|}{P_R}} \quad (24)$$

$$q_2 = \text{sgn}(P_s - P_2) q_r \bar{x}_s \sqrt{\frac{|P_s - P_2|}{P_R}} \quad (25)$$

である。 P_1, P_2 は、それぞれ油室 1, 油室 2 の圧力、 q_r はサーボ弁の定格流量、 P_s, P_r はそれぞれ供給圧、帰還圧とする。 P_R は定格流量降下とする。また、 $\bar{x}_s$ は、実際のサーボ弁のバルブ開度であり、ラップ Δ を用いて

$$\bar{x}_s = \begin{cases} 0, & |x_s| < \Delta \\ |x_s| - \Delta, & \Delta \leq |x_s| < 1 + \Delta \\ 1, & 1 + \Delta \leq |x_s| \end{cases} \quad (26)$$

のように与えられる。

シリンダーの圧力特性は

$$\frac{dP_1}{dt} = \frac{K_\phi}{V_{10} + A_1 y} \left(q_1 - A_1 \frac{dP_1}{dt} - C_i(P_1 - P_2) - C_o P_1 \right) \quad (27)$$

$$\frac{dP_2}{dt} = \frac{K_\phi}{V_{20} + A_2 y} \left(q_2 - A_2 \frac{dP_2}{dt} - C_i(P_1 - P_2) - C_o P_2 \right) \quad (28)$$

ここで、 K_ϕ は油の体積弾性係数、 C_i, C_o はそれぞれ内部漏れ係数、外部漏れ係数である。また、 V_{10}, V_{20} は油室 1, 油室 2 の体積、 A_1, A_2 は、油室 1, 油室 2 の断面積とする。

シリンダと負荷の質量の合計を M_t とし、摩擦特性を F_{ric} としたときのシリンダーの運動方程式を

$$M_t \frac{d^2 y}{dt^2} = A_1 P_1 - A_2 P_2 - F_{ric} \left(y, \frac{dy}{dt} \right) \quad (29)$$

とする。

制御対象のパラメータは、Table 5 のように定められている。また、油室 1, 油室 2 の体積、油室 1, 油室 2 の断面積は 1 に正規化されており、シリンダの負荷の質量 M_t も標準値は 1 である。

5.2 制御目的と制御仕様

Fig. 2 の油圧サーボ系において、シリンダ変位 y を目標値 $r(t)$

$$r(t) = \begin{cases} 0.6 & 5(i-1) \leq t < 5i, (i=1, 2, 3, \dots) \\ 0.0 & 5i \leq t < 5(i+1) \end{cases} \quad (30)$$

Table 5 Parameters

Symbol	Value
Δ	0.0027
q_r	9.8754
P_s	22.73
P_T	0.5413
P_R	9.7431
K_ϕ	866.05
C_i	0.0261
C_o	0.0106
ω_s	$2\pi \times 58$
ζ	0.65

に追従させる制御系を構成することを目的とする。なお、負荷とシリンダの質量 M_t は、

S1) $M_t = 0.2 \sim 10$

で変動するとする。また、制御仕様は、

S2) 定常値: $0.59 \leq A_{inf} \leq 0.61$

S3) 最大行き過ぎ量: $A_p \leq 10[\%]$

S4) 整定時間: $T_s \leq 1.0[s]$

S5) 立ち上がり時間: $R_s \leq 0.7[s]$

を満足しなければならないとする。

5.3 PID 制御の適用

Fig. 2 の制御対象に対して、PID 制御を適用する。ただし、制御誤差に対して PID 制御補償した際、制御結果が悪くなることを避けるため参照信号フィルタを設けた。

$$u(t) = \left(K_P + K_I \frac{1}{s} + K_D \frac{s}{0.1s + 1} \right) e(t) \quad (31)$$

$$e(t) = y(t) - \frac{1}{0.05s + 1} r(t) \quad (32)$$

5.4 入力ベクトルの選定

制御結果の特徴量からなるベクトル r_i の要素を決める。与えられた制御仕様は 5 つの指標のうち、立ち上がり時間 R_s は整定時間 T_s と相関があるので、 r_i は、つぎの 3 つ指標からなるベクトルとした。

$$r_i = [A_{inf}, A_p, T_s] \quad (33)$$

5.5 データベースの作成

各負荷 $M_t = 0.2, 0.6, 1, 3, 5, 7, 9$ に対して手動でパラメータ調整を行ない、その過程で得られるデータをデータベースに蓄える。まず、初期 PID パラメータを

$$u_1 = [K_{P1}, K_{I1}, K_{D1}] = [1 \ 0 \ 0]$$

として、閉ループ系のステップ応答試験を行ない、計測された定常値 A_{inf} 、オーバーシュート A_p 、整定時間 T_s を用いて

$$r_1 = [A_{inf}, A_p, T_s]$$

とする。これらの値から $s_1 = [u_1, r_1]$ を初期値としてさらにパラメータ調整を手動で行ない、その過程における PID パラメータと制御結果 (A_{inf}, A_p, T_s) と共にデータベースに蓄える。各負荷ごとにこの調整作業を数回繰り返しデータ

Table 6 The number of data for M_t

M_t	0.2	0.6	1	3	5	7	9	Total
The number of data	23	17	26	16	23	13	16	134

Table 7 Inputs to the proposed algorithm

M_t	s_1						reference		
	u_1			r_1			r_q		
	K_P	K_I	K_D	A_{inf}	$A_p(\%)$	$T_s(s)$	A_{inf}	A_p	T_s
2	1	0	0	0.597	10.79	0.66	0.6	0	0.8
3	1	0	0	0.598	18.39	0.88	0.6	0	0.8
4	1	0	0	0.601	23.32	1.05	0.6	0	0.8
5	1	0	0	0.602	26.78	1.56	0.6	0	0.8
6	1	0	0	0.601	30.25	1.79	0.6	0	0.8
7	1	0	0	0.599	33.54	1.99	0.6	0	0.8
8	1	0	0	0.598	36.15	2.17	0.6	0	0.8
9	1	0	0	0.598	38.04	2.35	0.6	0	0.8

ベースを構築した。構築されたデータベース中のデータ数を Table 6 に示す。

5.6 数値例と結果

第 5.5 項で構築したデータベースを使い、制御仕様を満足するような PID パラメータを抽出する。ここでは、どの負荷においても、 $t \leq 10$ では標準的な負荷 $M_t = 1$ に対して有効な PID パラメータ

$$u_1 = [K_{P1}, K_{I1}, K_{D1}] = [1 \ 0 \ 0] \quad (34)$$

を用いて追従を行ない、 $t = 10$ で PID パラメータの更新を行なった。 u_1 から得られる制御結果の特徴量 r_1 は、始めの 5 秒間のステップ応答から得られる制御結果を使用した。

各負荷での具体的な入力値を Table 7 に示し、提案手法により得られた PID パラメータとそれに対する制御結果を Table 8 に示す。また、各負荷に対するシリンダ位置の変位を Fig. 3 に示す。まず、定常値、行き過ぎ量、整定時間に関しては、負荷 $M_t = 1, 3, 5, 7$ において、目標値に近い制御結果となるような PID パラメータが抽出されている。負荷 $M_t = 2, 4, 6, 8$ は、データベースにない、つまり過去に経験していない負荷であるため、目標値との誤差が $M_t = 1, 3, 5, 7$ に比べて大きくなっているが、ある程度良好な制御結果となるパラメータが抽出されている。これは、データベースから現在の状況とよく似た過去の状況を探すという手順により補完機能が働いたためと考えられる。 $M_t = 8, 9$ に関しては、さらに誤差が大きくなっているが、これは、 $M_t = 8, 9$ という大きな負荷に対して PID 制御器が有効に機能せず、データベースに仕様を満たすデータがなかったためと考えられる。また、立ち上がり時間 R_s に関しては、入力ベクトルに加えなかったものの整定時間との相関からどの負荷においても制御仕様を満たす結果が得られている。

6. おわりに

本稿では、データベース駆動型の PID パラメータの自動抽出法を提案し、シミュレーションによりその有効性を確

Table 8 Estimated PID parameters and the control results

M_t	PID parameter $\hat{u}_q$			results			
	K_P	K_I	K_D	A_{inf}	A_P	$T_s(s)$	R_s
2	1.0250	0.0162	0.100	0.60	0.00	0.72	0.46
3	1.0275	0.0163	0.099	0.60	0.00	0.80	0.51
4	1.0251	0.0162	0.100	0.60	0.00	0.68	0.40
5	0.9802	0.0110	0.117	0.60	0.00	0.79	0.45
6	0.9600	0.0113	0.124	0.60	0.00	0.76	0.48
7	0.9608	0.0113	0.124	0.60	0.35	0.80	0.50
8	0.9600	0.0113	0.124	0.60	3.07	0.84	0.52
9	0.9801	0.0103	0.140	0.60	3.65	0.88	0.55

Fig. 3 Cylinder displacements by using the estimated PID parameters (solid line) and the reference input (dotted line), where $[K_P, K_I, K_D] = [1, 0, 0]$ is used before $t = 10$.

認した。本報告では、PID パラメータの調整は 1 回のみ行なったが、得られた制御結果をデータベースに蓄え、繰り返し PID パラメータの調整を行なうような学習機能を追加することで制御結果が向上することが予想される。今後の課題は、学習機能の追加と繰り返し調整による収束性の確認、ノイズを含めた事例への適用、また、データ数と制御性能の解析などが挙げられる。

謝辞 本研究は平成 14 年度科学研究費補助金若手研究 (B)No. 13750418 および平成 15 年度科学研究費補助金萌芽研究 No. 15656102 の補助のもとに行なわれた。ここに感謝の意を表する。また、第 5 章で用いたモデルを記述した Matlab/Simulink ファイルを SICE 制御適応学習制御調査研究会、特に東京都立科学技術助教授増田士朗先生に提供して頂いた。ここに感謝の意を表する。

参考文献

- 1) 重政, 芳谷, 大松: セルフチューニング PID 方式: 動向と事例, 計測と制御, **73-6**, 423/431 (1998)
- 2) A. Stenman: Model on demand: Algorithms, analysis and applications, PhD thesis, Department of Electrical Engineering Linköping University (1999)
- 3) A. Stenman, F. Gustafsson and H. Ljung: Just in time models for dynamical systems, 35th IEEE Conference on Decision and Control, 1115/1120 (1996)
- 4) G. Bontempi, M. Birattari and H. Bersini: Lazy Learning for local modeling and design, Int. J. Control, **72-7**, 8 (1999)
- 5) Q. Zheng and H. Kimura: Just-In-Time Modeling for Function Prediction and its Applications, Asian. J. Control, **3-1**, 35/44 (1999)
- 6) 江田, 大森, 佐野: Model on Demand 型局所線形モデリングを用いた制御系設計の一方法, 第 1 回適応学習制御シンポジウム, 65/68 (2001)
- 7) 牛田, 河村, 鄭, 木村: Just-In-Time 手法に基づく 2 自由度制御系の設計法, 第 2 回適応学習制御シンポジウム, 23/26 (2002)
- 8) 増田, 水野, 山本, 水本, 深尾: 電気油圧シリンダに対する適応制御用ベンチマーク問題の作成-SICE 制御部門適応学習制御調査研究会活動報告-, 計測と制御, **40-10**, 758/763 (2001)
- 9) 川路茂保: フレキシブル・インテリジェンスの概念と産業応用; 第 3 回適応学習制御シンポジウム特別講演資料 (2003)

[著者紹介]

太田 純

2002 年, 大阪大学基礎工学部システム科学科卒業。同年, 大阪大学大学院基礎工学研究科システム人間系専攻博士前期過程に入学し現在に至る。

山本 茂 (正会員)

1989 年大阪大学大学院工学研究科博士前期課程修了。現在, 大阪大学大学院基礎工学研究科助教授。制御理論とその応用, 特に通信ネットワークに関連した制御やカオス制御などの研究に従事。大阪大学博士 (工学), IEEE, システム制御情報学会などの会員