



Tokyo Gakugei University Repository

東京学芸大学リポジトリ

<http://ir.u-gakugei.ac.jp/>

Title	偏差に応じてPIDチューニングを行うデータ駆動型制御系の一設計(fulltext)
Author(s)	佐野,桃香; 今井,慎一
Citation	東京学芸大学紀要. 自然科学系, 72: 117-123
Issue Date	2020-10-30
URL	http://hdl.handle.net/2309/159761
Publisher	東京学芸大学教育実践研究推進本部
Rights	

偏差に応じてPIDチューニングを行うデータ駆動型制御系の一設計

佐野 桃香^{*1}・今井 慎一^{*2}

技術科学分野

(2020年7月7日受理)

SANO, M. and IMAI, S.: Design of Data-Driven Control System for PID Tuning According to Error. Bull. Tokyo Gakugei Univ. Div. Nat. Sci., 72: 117-123. (2020) ISSN 2434-9380

Abstract

PID control with a simple structure is widely used in industry, especially in process systems. However, most real systems are non-linear systems, and it is difficult for fixed PID control to always obtain good control results. Therefore, a method called data-driven PID control in which the PID gain is changed sequentially has been proposed, and the adaptability to nonlinear systems has been improved. However, the data-driven control has a heavy computational load and has not been put to practical use. In this paper, a method of tuning the PID gain according to the deviation (control error) of the control result is proposed for non-linear systems. For the data driven PID control, create a program that performs PID tuning only when the error occurs with a certain size. A threshold is set for the error, and data-driven PID control is performed only when the threshold is exceeded. Simulate a Hammerstein model of a nonlinear system. Comparing the control results of the conventional method and the proposed method, it was found that there was almost no difference. However, the number of PID gain changes was significantly reduced 1/7 times compared to the conventional method. In the future, it will be continued to discuss ways to set more appropriate thresholds.

Keywords: Feedback Control, PID parameter, Data-Driven, Error

Department of Technology Sciences, Tokyo Gakugei University, 4-1-1 Nukuikita-machi, Koganei-shi, Tokyo 184-8501, Japan

要旨: 産業界, 特に化学プロセスや石油精製プロセスに代表されるプロセス系において, PID制御が広く使われている。その理由として, PID制御は構造が単純であることや, その物理的な意味が明確であることなどが挙げられる。ところが, 実在するシステムの多くは非線形システムであり, 固定のPID制御は常に良好な制御結果を得ることが難しい。そこで, システムの新しい入出力データをデータベースに保存し, 蓄えられたデータから新しいPIDゲインを算出するデータ駆動型PID制御という方法が提案された。PIDゲインが逐次変更されるため, 従来の固定PID制御より, 非線形なシステムへの適応能力が向上したことが報告されている。しかし, データ駆動型制御では, 1ステップごとにPIDチューニングを行っているため, 計算処理の負荷が大きく, 高精度のコンピュータでなければ処理ができないという問題がある。そこで本研究では, 制御結果を偏差(制御誤差)に応じてPIDゲインをチューニングする方法を提案し, 数値例を通して本手法の有効性を検証する。データ駆動型PID制御に対して, 偏差がある一定の大きさで生じた場合のみ, PIDチューニングを実施する制御系を設計する。偏差に対して閾値を設け, 閾値を超えた場合のみデータ駆動型PID制御を行う。閾値より偏差が小さい場合はPIDゲインの変更をせずに制御を行う。非線形システ

*1 東京学芸大学 2019年度卒業生

*2 東京学芸大学 技術・情報科学講座 技術科学分野 (184-8501 東京都小金井市貫井北町 4-1-1)

ムである Hammerstein モデルに対しシミュレーションを行う。従来法のデータ駆動型の制御結果と閾値が0.1のときの提案手法の制御結果を比較し、差が少ないことがわかった。しかし、PIDゲインの変更回数については、従来法は195回であったが、提案手法は27回であり、従来法に比べて1/7回という大幅なPIDゲインの変更回数の削減ができた。

1. はじめに

産業界、特に化学プロセスや石油精製プロセスに代表されるプロセス系において、制御構造の簡単なPID制御⁽¹⁾が広く使われている。その理由として、PID制御は構造が単純であることや、その物理的な意味が明確であることなどが挙げられる。ところが、実在するシステムの多くはその特性が環境条件や動作条件に応じて変動したり、数学モデルで記述できなかつたりするような要素を含む。そのような非線形システムに対して、固定のPID制御は良好な制御結果を得ることが難しい。そこで非線形システムに対して有効な制御手法として、データ駆動型制御⁽²⁾⁽³⁾⁽⁴⁾等が提案されている。データ駆動型制御は、新しいシステムの入出力データが得られるたびにデータベースに保存し、必要に応じて、過去に大量に蓄えられているデータベースから要求点に類似したものを近傍として取り出し、局所モデルを作る手法である。これは、コンピュータ技術の発展により、大量のデータの保存及び読み出し、書き込み、高速な検索などが可能となったためである。データ駆動型制御によって、PIDゲインが逐次変更されるため、従来の固定PID制御より、非線形なシステムへの適応能力が向上したことが報告されている。しかし、データ駆動型制御では、1ステップごとにPIDチューニングを行っているため、計算処理の負荷が大きく、高精度のコンピュータでなければ処理ができないという問題がある。

そこで本研究では、制御結果が目標値に追従しなかった場合のみPIDゲインをチューニングする方法を提案する。その方法として、偏差に対して閾値を設け、閾値を超えた場合のみデータ駆動型PID制御を行う。閾値より偏差が小さい場合はPIDゲインの変更をせずに制御を行う。非線形システムである Hammerstein モデルに対しシミュレーションを行い、計算負荷の削減と制御結果に着目した時の本手法の有効性を検証する。

2. 偏差に応じてPIDチューニングを行うデータ駆動型制御系の一設計

提案する制御系のブロック線図を図1に示す。提案

する制御法では、データ駆動型PID制御に対して、偏差を評価する。ある一定の大きさで偏差が生じた場合のみ、PIDチューニングを実施することにより、非線形システムを制御する。

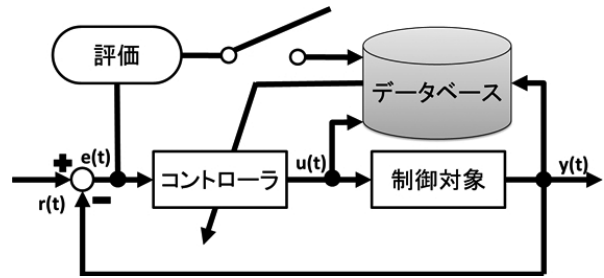


図1 提案手法のブロック線図

2. 1 システムの記述

制御対象として、次式で表わされる離散時間非線形システムを考える。

$$y(t) = f(\varphi(t-1)) \quad (1)$$

ここで、 $y(t)$ はシステム出力、 $f(\cdot)$ は非線形関数、 $\varphi(t-1)$ はシステムの時刻 t より前の状態 (ヒストリカルデータ) を表す。 $\varphi(t-1)$ を情報ベクトルと呼ぶことにし、以下で定義する。

$$\varphi(t-1) := [y(t-1), \dots, y(t-n_y), u(t-1), \dots, u(t-n_u)] \quad (2)$$

ここで、 $u(t)$ は制御入力、 n_y 、 n_u はそれぞれ出力と入力の次数である。データ駆動型モデリングでは (2) 式の形式でデータベースへの蓄積が行われる。また、出力 $y(t+1)$ の予測値を得るために必要な $\varphi(t)$ を要求点と呼ぶ。要求点に対する出力の予測値は、事前に大量に蓄えられたデータベースから要求点に類似した情報ベクトル $\varphi(t-1)$ を近傍として抽出し、局所モデルを構成することによって得られる。

2. 2 データ駆動型PID制御器の設計⁽²⁾

制御則は次式のようなI-PD制御則を適用する。

$$\begin{aligned}\Delta u(t) &= \frac{k_c T_s}{T_I} e(t) - k_c \left(\Delta + \frac{T_D}{T_s} \Delta^2 \right) y(t) \\ &= K_I e(t) - K_P \Delta y(t) - K_D \Delta^2 y(t)\end{aligned}\quad (3)$$

ここで、 $e(t)$ は制御誤差信号であり、 $r(t)$ を目標値とすると以下で定義される。

$$e(t) := r(t) - y(t)\quad (4)$$

また、 k_c 、 T_I 、および T_D はそれぞれ比例ゲイン、積分時間、微分時間を、 T_s はサンプリング時間を表す。 $K_P = k_c$ 、 $K_I = k_c T_s / T_I$ 、 $K_D = k_c T_D / T_s$ と置き換えており、 K_I 、 K_D はそれぞれ積分ゲイン、微分ゲインと呼ぶ。 $\Delta (= 1 - z^{-1})$ は差分演算子を表している。プロセス系の多くは非線形性を有しているため、PIDゲイン(K_P 、 K_I 、 K_D)が固定のままであると、良好な制御結果を得ることが困難である。したがって、データ駆動型モデリング法の考えに基づき、システムの特長、状態に応じてPIDゲインが自己調整される制御手法を考える。ここで(3)式を以下のような可変ゲイン($K_P(t)$ 、 $K_I(t)$ 、 $K_D(t)$)を持つ制御則に置き換えることを考える。

$$\Delta u(t) = K_I(t)e(t) - K_P(t)\Delta y(t) - K_D(t)\Delta^2 y(t)\quad (5)$$

(5)式は以下のような関係式で書くことができる。

$$u(t) = g(\varphi'(t))\quad (6)$$

$$\varphi'(t) := [K(t), r(t), y(t), y(t-1), y(t-2), u(t-1)]\quad (7)$$

$$K(t) = [K_P(t), K_I(t), K_D(t)]\quad (8)$$

ただし、 $g(\cdot)$ は線形関数を表している。ここで(6)式と(7)式を(1)式と(2)式に代入すると、

$$y(t+1) = h(\tilde{\varphi}(t))\quad (8)$$

$$\begin{aligned}\tilde{\varphi}(t) &:= [y(t), \dots, y(t-n_y+1), K(t), \\ &\quad r(t), u(t-1), \dots, u(t-n_u+1)]\end{aligned}\quad (9)$$

ただし、 $n_y \geq 3$ 、 $n_u \geq 2$ であり、 $h(\cdot)$ は非線形関数を表している。よって $K(t)$ は以下のように与えられる。

$$K(t) = F(\tilde{\varphi}(t))\quad (10)$$

$$\begin{aligned}\tilde{\varphi}(t) &:= [y(t+1), y(t), \dots, y(t-n_y+1), \\ &\quad r(t), u(t-1), \dots, u(t-n_u+1)]\end{aligned}\quad (11)$$

ただし、 $F(\cdot)$ は非線形関数を表している。(11)式には、未来の出力値 $y(t+1)$ が含まれているが、 $y(t+1) \rightarrow r(t+1)$ となるような制御系を設計することを考えており、ここでは $y(t+1)$ を $r(t+1)$ に置き換えて考える。また $r(t+1)$ は時刻 t において与えられるものとする。したがって、(11)式の $\tilde{\varphi}(t)$ を次式として改めて定義する。

$$\begin{aligned}\tilde{\varphi}(t) &:= [r(t+1), r(t), y(t), \dots, y(t-n_y+1), \\ &\quad u(t-1), \dots, u(t-n_u+1)]\end{aligned}\quad (12)$$

以上の準備の後、データ駆動型PIDコントローラを設計する。その具体的なアルゴリズムを以下に示す。

[STEP1] 初期データベースの作成

データ駆動型モデリング法では、過去に蓄積したデータが存在しない場合、原理的に同定を行うことができない。したがって、本手法では、ある平衡点周りで得られた入出力データから、Ziegler & Nichols (ZN)法⁽⁵⁾や、Chien, Hrones & Rewisk (CHR)法⁽⁶⁾などを用いて得られるPIDゲインを使用し、それによる入出力データとPIDゲインからなる次式を初期データベースとして作成する。

$$\Phi(j) := [\tilde{\varphi}(j), K(j)], \quad j = 1, 2, \dots, N(0)\quad (13)$$

ただし、 $\tilde{\varphi}(j)$ と $K(j)$ は(12)式および(8)式により与えられる。また、 $N(0)$ は初期データ数(初期データベースにおける情報ベクトルの数)を表している。

ここで、制御対象の操業範囲やその条件は事前に把握できることから、初期データベースには、データ駆動型モデリングにおいて局所線形モデルを構築するために必要となる十分な操業データを格納する。

[STEP2] 距離の計算、近傍の選択

要求点 $\tilde{\varphi}(t)$ とデータベースに蓄えられている情報ベクトル $\tilde{\varphi}(j)$ との距離を次式の重みつき \mathcal{L}_1 ノルムにより求める。

$$d_{i,j}(\tilde{\varphi}(t), \tilde{\varphi}(j)) = \sum_{l=1}^{n_y+n_u+1} \left| \frac{\tilde{\varphi}_l(t) - \tilde{\varphi}_l(j)}{\max \tilde{\varphi}_l(m) - \min \tilde{\varphi}_l(m)} \right| \quad j = 1, 2, \dots, N(t)\quad (14)$$

ここで、 $N(t)$ は時刻 t においてデータベースに蓄えられているデータ数 (情報ベクトル数) を示している。また、 $\bar{\varphi}_l(j)$ は、第 j 番目の情報ベクトルの第 l 番目の要素を示している。同じく、 $\bar{\varphi}_l(t)$ は、時刻 t における要求点の第 l 番目の要素を示している。さらに $\max \bar{\varphi}_l(m)$ は、データベースにあるすべての情報ベクトル ($\bar{\varphi}_l(j), j=1, 2, \dots, N(t)$) の第 l 番目の要素の中で、最も大きな要素を表しており、 $\min \bar{\varphi}_l(m)$ は、その最小値を表している。いま、(14) 式により求められた距離 d が小さいものから k 個の情報ベクトルを選択し、以下、その選択されたデータ集合を近傍として定義することとする。

[STEP3] 局所モデルの構成

つぎに、STEP2において選択された近傍に対して、以下で示される重みつき局所線形平均法 (Linearly Weighted Average: LWA)⁽⁷⁾ により局所モデルを構成する。

$$K^{old}(t) = \sum_{i=1}^k w_i K(i), \quad \sum_{i=1}^k w_i = 1 \quad (15)$$

ここで、 w_i は選択された第 i 番目の情報ベクトルに含まれる $K(i)$ に対する重みであり、次式で与える。

$$w_i = \sum_{l=1}^{n_y+n_{u+1}} \left(1 - \frac{[\bar{\varphi}_l(t) - \bar{\varphi}_l(j)]^2}{[\max \bar{\varphi}_l(m) - \min \bar{\varphi}_l(m)]^2} \right) \quad (16)$$

[STEP4] データ修正

STEP3において、局所モデルとして得られるPIDゲイン K^{old} は、STEP2, STEP3を繰り返し行い、PIDゲインを逐次抽出したとしても、システムの状態に応じてPIDゲインが適切に調整されない恐れがある。そこで、STEP3で得られたPIDゲイン K^{old} に応じた修正を行ない、その修正されたデータ K^{new} をデータベースに蓄えるものとする。その修正方法は以下のような最急降下法を用いる。

$$K^{new}(t) = K^{old}(t) - \eta \frac{\partial J(t+1)}{\partial K(t)} \quad (17)$$

$$\eta := [\eta_p, \eta_I, \eta_D] \quad (18)$$

ここで、 η は学習係数、 $J(t+1)$ は以下で定義される誤差の評価規範を表している。

$$J(t+1) := \frac{1}{2} \varepsilon(t+1)^2 \quad (19)$$

$$\varepsilon(t) := y_r(t) - y(t) \quad (20)$$

ただし、 $y_r(t)$ は参照モデルの出力を示しており、次式のように設計される。

$$y_r(t) = \frac{z^{-1}T(1)}{T(z^{-1})} r(t) \quad (21)$$

$T(z^{-1})$ は参照モデルの特性多項式である。この $T(z^{-1})$ を特性多項式として、次式により設計する。

$$T(z^{-1}) := 1 + t_1 z^{-1} + t_2 z^{-2} \quad (22)$$

$$\left. \begin{aligned} t_1 &= -2 \exp\left(-\frac{\rho}{2\mu}\right) \cos\left(\frac{\sqrt{4\mu-1}}{2\mu} \mu\right) \\ t_2 &= \exp\left(-\frac{\rho}{\mu}\right) \\ \rho &:= \frac{T_s}{\sigma} \\ \mu &:= 0.25(1-\delta) + 0.51\delta \end{aligned} \right\} \quad (23)$$

ここで、 T_s はサンプリング時間を示す。 σ は立ち上がり時間に相当するパラメータであり、実用的にはシステムの時定数を目安とし、 $1/2 \sim 1/3$ 程度に設計するのが望ましいとされる。また、 μ は応答の減衰特性に関係するパラメータを示しており、 δ によって調整される。 δ は実用的には、 $0 \leq \delta \leq 2.0$ として設計することが望ましいとされ、 δ を大きくすると応答形状は振動的になる。(23) 式に関しては、参考文献⁽⁸⁾に基づいて設計する。

(20) 式において、目標値 $r(t)$ ではなく参照モデルの出力 $y_r(t)$ を用いることにより、モデル追従特性を有したPID制御系が設計できる。また、(17) 式、右辺第2項のそれぞれの偏微分は以下のように展開される。

$$\left. \begin{aligned} \frac{\partial J(t+1)}{\partial K_p(t)} &= \frac{\partial J(t+1)}{\partial \varepsilon(t+1)} \frac{\partial \varepsilon(t+1)}{\partial y(t+1)} \frac{\partial y(t+1)}{\partial u(t)} \frac{\partial u(t)}{\partial K_p(t)} \\ &= -\varepsilon(t+1) \theta_1(t) \frac{\partial y(t+1)}{\partial u(t)} \\ \frac{\partial J(t+1)}{\partial K_I(t)} &= \frac{\partial J(t+1)}{\partial \varepsilon(t+1)} \frac{\partial \varepsilon(t+1)}{\partial y(t+1)} \frac{\partial y(t+1)}{\partial u(t)} \frac{\partial u(t)}{\partial K_I(t)} \\ &= -\varepsilon(t+1) \theta_2(t) \frac{\partial y(t+1)}{\partial u(t)} \\ \frac{\partial J(t+1)}{\partial K_D(t)} &= \frac{\partial J(t+1)}{\partial \varepsilon(t+1)} \frac{\partial \varepsilon(t+1)}{\partial y(t+1)} \frac{\partial y(t+1)}{\partial u(t)} \frac{\partial u(t)}{\partial K_D(t)} \\ &= -\varepsilon(t+1) \theta_3(t) \frac{\partial y(t+1)}{\partial u(t)} \end{aligned} \right\} \quad (24)$$

ただし、

$$\Theta_k(t) = \begin{cases} -\Delta y(t) & : k = 1 \\ e(t) & : k = 2 \\ -\Delta^2 y(t) & : k = 3 \end{cases} \quad (25)$$

ここで、(24) 式を計算するためには、システムヤコビアン $\partial y(t+1)/\partial u(t)$ を求めなければならない。いま、 $x = |x| \text{sign}(x)$ という関係を用いて、そのシステムヤコビアンを以下のように考える。

$$\frac{\partial y(t+1)}{\partial u(t)} = \left| \frac{\partial y(t+1)}{\partial u(t)} \right| \text{sign} \left(\frac{\partial y(t+1)}{\partial u(t)} \right) \quad (26)$$

ただし、 $\text{sign}(x) = 1(x > 0), -1(x < 0)$ 。いま、 $|\partial y(t+1)/\partial u(t)|$ を学習率 η に含ませて考えると、システムヤコビアンの計算に対してはシステムの入出力間の勾配の符号さえわかればよいことになる。したがって、本論文においてその符号は一定、もしくは既知であるとする。

[STEP5] 冗長データ削除

実システムへの適用を考えた場合、本手法はサンプリング時間内にSTEP2からSTEP4までの操作を行わなければならないという時間的制約がある。その手順は以下の2段階からなっている。

まず、情報ベクトルの距離に関する条件 ((27) 式) を満たすデータ $\Phi(\bar{i}) (\bar{i} = 1, 2, \dots)$ を、近傍データを除いたデータベースより選択する。

[条件1]

$$d(\bar{\varphi}(t), \bar{\varphi}(i)) \leq \alpha_1, \quad i = 1, 2, \dots, N(t) - k \quad (27)$$

ただし、 $\Phi(\bar{i})$ は次式で定義される。

$$\Phi(\bar{i}) := [\bar{\varphi}(\bar{i}), K(\bar{i})], \quad \bar{i} = 1, 2, \dots \quad (28)$$

さらに、PIDパラメータに関する条件 ((29) 式) を満たすデータ $\Phi(i) (i = 1, 2, \dots)$ を $\Phi(\bar{i})$ の中から選択する。

[条件2]

$$\sum_{i=1}^3 \left\{ \frac{K_i(\bar{i}) - K_i^{\text{new}}(t)}{K_i^{\text{new}}(t)} \right\}^2 \leq \alpha_2 \quad (29)$$

ただし、 $\Phi(i)$ は次式で定義される。

$$\Phi(i) := [\bar{\varphi}(i), K(i)], \quad i = 1, 2, \dots \quad (30)$$

$\Phi(i)$ が複数存在する場合は、(29) 式の値が最も小さい情報ベクトルをデータベースから削除する。

以上のアルゴリズムにより、冗長なデータが削除され、データの過剰な増加を抑制することができる。

[STEP6]

STEP2に戻る。

以上により、データ駆動型モデリング法に基づいたPIDコントローラの設計を行うことができる。

2.3 偏差を応じたPIDチューニング法

2.2節のデータ駆動型PID制御に対して、偏差がある一定の大きさで生じた場合のみ、PIDチューニングを実施する方法について説明する。まず、閾値を τ と定義する。 τ に関する条件を次式とする。

$$\tau > |e(t)| \quad (31)$$

この条件を満たす場合のみ2.2節のデータ駆動型PID制御を行い、満たさない場合はPIDゲインを変更せずに制御を行う。ここで、(31) 式に含まれているパラメータ τ は、ある正定数で与えられる設計パラメータを表している。このパラメータの設定には、若干の試行錯誤を必要とする。

3. 数値例

制御対象として次式で与えられるHammersteinモデル⁽⁹⁾を用いる。Hammersteinモデルは、アクチュエータに非線形性を表現する場合などに用いられる。いま、Hammersteinモデルを次式で与える。

$$\left. \begin{aligned} y(t) &= 0.6y(t-1) - 0.1y(t-2) \\ &\quad + 1.2x(t-1) - 0.1x(t-2) + \xi(t) \\ x(t) &= 1.5u(t) - 1.5u(t)^2 + 0.5u(t)^3 \end{aligned} \right\} \quad (32)$$

ここで、 $\xi(t)$ は平均0、分散0.001のガウス性白色雑音である。このHammersteinモデルの静特性を図2示す。図から、このシステムが非線形を有していることがわかる。

また、目標値 $r(t)$ を以下のように与える。

$$r(t) = \begin{cases} 0.5(0 \leq t < 50) \\ 1.0(50 \leq t < 100) \\ 2.0(100 \leq t < 150) \\ 1.5(150 \leq t \leq 200) \end{cases} \quad (33)$$

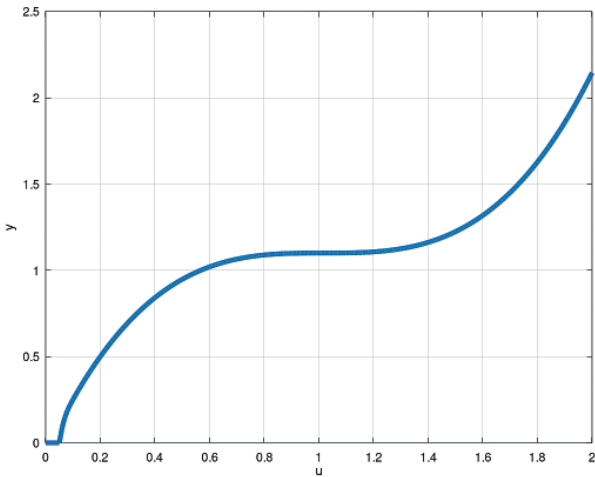


図2 Hammersteinモデルの静特性

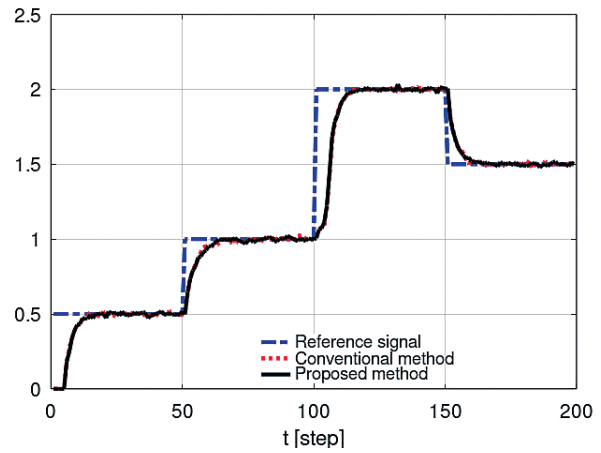


図3 従来法と提案手法の制御結果の比較

表1 各種設計パラメータ

出力と入力の次数	$n_y = 3$ $n_u = 2$
近傍データの数	$k = 6$
学習係数	$\eta_p = 0.8$ $\eta_l = 0.8$ $\eta_D = 0.2$
初期データ数	$N(0) = 200$
データベース更新のための設計パラメータ	$\alpha_1 = 0.5$ $\alpha_2 = 0.1$
閾値	$\tau = 0.1$

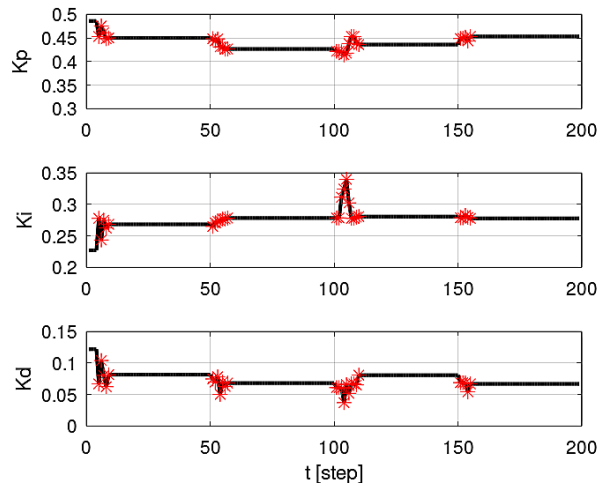


図4 $\tau = 0.1$ のときのPIDゲインの時間的变化

さらに、情報ベクトル $\bar{\varphi}(t)$ を以下のように与える。

$$\bar{\varphi}(t) := [r(t+1), r(t), y(t), y(t-1), y(t-2), u(t-1)] \quad (34)$$

また、参照モデルに含まれている設計多項式 $T(z^{-1})$ を以下のように設計する。

$$T(z^{-1}) = 1 - 0.271z^{-1} + 0.0183z^{-2} \quad (35)$$

ここで、 $T(z^{-1})$ は先にも述べたように、文献⁽⁸⁾を参考にして設計した。さらに、提案手法に含まれている各種設計パラメータを表1にまとめる。

データ駆動型制御における初期データベースに蓄えておくPIDゲインは、CHR法に基づいて算出する。そのPIDゲインは以下のとおりである。

$$K_p = 0.486, K_I = 0.226, K_D = 0.122 \quad (36)$$

図3は従来法のデータ駆動型の制御結果と提案手法の制御結果を示す。図4にPIDゲインの時間的变化を示す。また、図4に示すポイントは、PIDゲインをチューニングしている所を示す。図3および図4の結果から、従来のデータ駆動型の制御結果と提案法の制御結果を比較し、差が少ないことが確認される。しかし、PIDゲインの変更回数については、従来法は逐次PIDゲインをチューニングしているため195回の変更が必要であるのに対し、提案手法では27回で収まっており、従来法に比べて約1/7回という大幅なPIDゲインの変更回数の削減ができた。

4. まとめ

本論文では、非線形システムに対し、偏差に応じてPIDチューニングを行うデータ駆動型制御系について

考察した。具体的には、閾値を定義し、閾値に関する条件を満たした場合のみPIDゲインを変更し制御を行う手法を提案し、非線形システムであるHammersteinモデルに対しシミュレーションを行い、本手法の有効性を検証した。今後は、偏差の閾値の設定について、項目反応理論を用いて制御性能を評価する予定である。項目反応理論は、測定精度をきめ細かく確認できる点がある。この点を制御に活かすと、制御の精度を適切に評価できると考えられる。

引用文献

- (1) 須田信英 編著：PID制御，朝倉書店（1992）
- (2) T.Yamamoto, K.Takao and T.Yamada: Design of a Data-Driven PID Controller, IEEE Trans. on Control Systems Technology, Vol.17, No.1, pp.29-39 (2009)
- (3) 今井慎一・山本透：データベースの切り替え機構を有するデータ駆動型制御系の一設計，電気学会論文誌. C, 電子・情報・システム部門誌, Vol.133, No.6, pp.1096-1102 (2013)
- (4) 木下拓矢・山本透：類似度に基づくデータベース駆動型制御系の一設計，電気学会論文誌. C, 電子・情報・システム部門誌, Vol.140, No.3, pp.312-319 (2020)
- (5) J.G.Ziegler and N.B.Nichos: Optimun settings for automatic controllers, Trans.ASME, Vol.64, No.8, pp.759-768 (1942)
- (6) K.L.Chien, J.A.Hrones and J.B.Reswick: On the Automatic Control of Generalized Passive Systems, Trans.ASME, Vol.74, pp.175-185 (1972)
- (7) C.G.Atkeson, A.W.Moore and S.Shaal: Locally weighted learning for control, Artificial Intelligence Review, Vol.11, pp.75-114 (1997)
- (8) 重政隆, 高木康夫, 市川義則, 北森俊行：制御系設計のための実用的な汎用参照モデル，計測自動制御学会論文集, Vol.19, No.7, pp.592-594 (1983)
- (9) P.Stoica and Söderström: Instrumental-variable methods for identification of Hammerstein system, Int. J. Contr., Vol.35, pp.459-476 (1982)