

安定余裕を考慮したデータベース駆動型制御系の一設計

Design of a Database-Driven PID Controller Considering Stability Margin

藤原 啓示[†] 木下 拓矢[†] 山本 透[†]

Keishi Fujiwara[†] Takuya Kinoshita[†] Toru Yamamoto[†]

[†] 広島大学 工学研究科

1 緒言

産業プロセスにおいては、制御構造が簡単で物理的意味が明確なPID制御[1]が、今なお幅広く用いられている。しかしながら、実システムの多くは非線形性を有しており、固定PIDパラメータで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法として、大量の操業データから直接制御パラメータを算出するデータベース駆動型PID制御[2]が提案されている。さらに、データベースに含まれるPIDパラメータをオフラインで学習可能なFictitious Reference Iterative Tuning (FRIT)[3]を用いたデータベース駆動型PID制御系[4]が提案されている。しかし、従来の方法では制御系の安定性は全く議論されていない。

本稿では、安定余裕を考慮したデータベース駆動型PID制御法を提案する。提案法では、算出されたPIDパラメータとシステムパラメータを用いることで、閉ループ伝達関数の安定余裕[5]を陽に取り扱うことができる。本研究は、従来の制御パラメータのみを格納していた方法に加え、システムパラメータも併せて格納し、これを用いて安定余裕を算出する。

本稿の構成を以下に示す。はじめに、安定余裕を考慮したデータベース駆動型PID制御系の制御系設計法について述べる。つぎに、FRITを用いたデータベース駆動型PID制御のオフライン学習を述べる。続いて、安定余裕と1-パラメータチューニングについて述べる。最後に、数値シミュレーションを通して本手法の有効性を検証する。

2 安定余裕を考慮したデータベース駆動型制御

2.1 制御系の概要

本稿で提案する安定余裕を考慮したデータベース駆動型PID制御の概要図を図1に示す。本制御系を簡単に説明する。まず、初期PIDコントローラにより制御することで得られた入出力と目標値を、データベース(Database)に格納する。また、FRITを用いたデータベース駆動型制御を用いてオフラインでPIDゲインを算出する。次に、データベース駆動型制御を行いPIDゲイン K^{old} とシステムパラメータ θ を算出する。算出された双方のパラメータを用いてシステムの安定余裕の算出を行う。安定余裕が小さい場合、簡単に調整可能な1-パラメータチューニング(1-Parameter

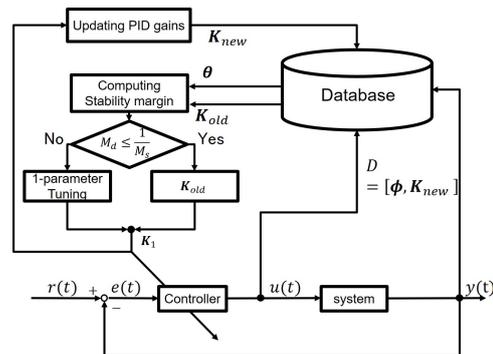


図 1: Block diagram of the proposed scheme.

Tuning) を用いて、PIDパラメータを調整する。安定余裕が大きい場合は、データベース駆動型制御で算出されたPIDパラメータを用いて制御する。

2.2 システムの記述

制御対象として、次式で表される離散時間非線形システムを考える。

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

ここで、 $y(t)$ システム出力、 $f(\cdot)$ は非線形関数、 $\phi(t-1)$ はシステムの時刻 t より前の状態を表しており、情報ベクトルと呼ぶ。また、情報ベクトル $\phi(t-1)$ は次式で定義される。

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

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。 $\phi(t-1)$ は現時刻 t よりも前の入出力データを含む情報ベクトルである。

2.3 安定余裕を考慮したデータベース駆動型制御

本稿では、制御則として以下のような速度型 I-PD 制御則を適用する。

$$\Delta u(t) = K_I(t)e(t) - K_P(t)\Delta y(t) - K_D(t)\Delta^2 y(t) \quad (3)$$

ただし、 Δ は差分演算子を表し、 $\Delta := 1 - z^{-1}$ で定義される。また、 $e(t)$ は制御誤差信号であり、 $r(t)$ を目標値とし、以下で定義される。

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

また、 $K_P(t)$, $K_I(t)$, $K_D(t)$ は、それぞれ各ステップにおける比例ゲイン、微分ゲイン、積分ゲインを表している。

[STEP1] 初期データベースの作成

まず、入出力データ、PID ゲイン、およびシステムパラメータから構成される次式の情報ベクトル $\phi(j)$ を考える。この情報ベクトルは、ある操業区間の各時刻の入出力データとその際適用している PID ゲインを対にして構成される。

$$\phi(j) = [\phi(j), \mathbf{K}(j)] \quad j = 1, 2, \dots, N(0) \quad (5)$$

さらに、 $\bar{\phi}(j)$, $\mathbf{K}(j)$ は次式で与えられ、それぞれ入出力データ、PID ゲインを表している。

$$\begin{aligned} \phi(t) = & [-y(t-1), \dots, -y(t-m_y), \\ & u(t-k_r-1), \dots, u(t-k_r-m_u+1), \\ & r(t+1), r(t), y(t), \dots, y(t-n_y+1), \\ & u(t-1), \dots, u(t-n_u+1)] \end{aligned} \quad (6)$$

$$\mathbf{K}(t) := [K_P(t), K_I(t), K_D(t)] \quad (7)$$

また、 $N(0)$ は初期データ数（初期データにおける情報ベクトルの数）を表す。初期データベースにおける PID ゲインは固定であるので、 $\mathbf{K}(1) = \dots = \mathbf{K}(N(0))$ である。

[STEP2] オフラインで PID パラメータの算出

FRIT を用いたデータベース駆動型制御によるオフライン学習によって PID パラメータの算出し、データベースに格納する。オフライン学習については次の章において説明する。

[STEP3] 距離の選択、近傍の選択

時刻 t における目標値と入出力で構成された要求点 $\bar{\phi}(t)$ とデータベースに蓄えられている情報 $\bar{\phi}(j)$ との距離を、次式の重みつき L_1 ノルムから求める。

$$d(\bar{\phi}(t), \bar{\phi}(j)) = \sum_{l=1}^{n_y+n_u+1} \left| \frac{\bar{\phi}_l(t) - \bar{\phi}_l(j)}{\max \bar{\phi}_l(m) - \min \bar{\phi}_l(m)} \right| \quad (8)$$

(ただし $j = 1, 2, \dots, N(t)$)

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）を表している。同じく、 $\bar{\phi}(j)$ は時刻 t における要求点の第 l 番目の要素を表している。さらに $\max \bar{\phi}(m)$ はデータベースにあるすべての情報ベクトル ($\bar{\phi}(j), j = 1, 2, \dots, N(t)$) の第 l 番目の要素の中の最大値を、 $\min \bar{\phi}(m)$ は最小値を表している。いま、式 (8) 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、その選択されたデータ集合を近傍として定義する。

[STEP4] 局所コントローラの構成

次に、STEP2 において選択された近傍に対して、以下で示される、重みつき局所的線形平均法 (Linearly Weighted Average: LWA) により局所コントローラを

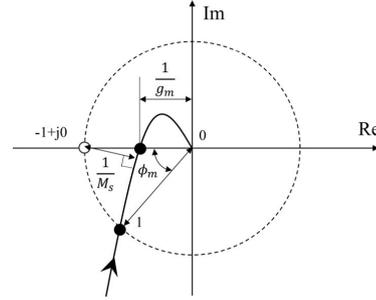


図 2: Stability margin in Nyquist diagram.

構成する。

$$\mathbf{K}^{old}(t) = \sum_{i=1}^k w_i \mathbf{K}(i), \quad \sum_{i=1}^k w_i = 1 \quad (9)$$

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 $\mathbf{K}(i)$ に対する重みであり、次式で与える。

$$w_i = \frac{e^{-d(i)}}{\sum_{t=1}^k e^{-d(t)}} \quad (10)$$

[STEP5] システムパラメータ推定

予測誤差 η がしきい値 $\gamma\sigma_\epsilon$ を超えた場合に、逐次最小二乗法によりシステムパラメータ推定を行う。 γ は統計的観点から 3.0~5.0 程度を目安とし、 σ_ϵ は推定誤差の標準偏差である。予測誤差がしきい値を超えない場合は、前のシステムパラメータの値をそのまま現在の値に代入する。さらに、推定されたシステムパラメータ \hat{a}_1 , \hat{b}_0 , \hat{b}_1 を用いて安定余裕を求めるために、非線形システムを一次遅れ系に近似する。システムゲインと時定数は以下の式を用いて算出する。 $\hat{L} = 4$ とする。

$$\hat{T} = \frac{-T_s}{\log(-\hat{a}_1)} \quad (11)$$

$$\hat{K} = \frac{\hat{b}_0 + \hat{b}_1}{1 + \hat{a}_1} \quad (12)$$

[STEP6] 安定余裕

算出された PID パラメータとシステムパラメータを用いて、図 2 に示すナイキスト線図の安定余裕に関するパラメータ M_s から、システムの安定余裕を調べる。 M_s は感度関数の最大値であり、式 (14) に示す感度関数を用いる。

$$M_s = \max_{\omega} |S(j\omega)| \quad (13)$$

$$S(j\omega) = \frac{1}{1 + G(j\omega)C(j\omega)} \quad (14)$$

図 2 に示すように、 M_s が大きい場合は安定余裕が小さくなり、 M_s が小さい場合は安定余裕が大きくなる。

したがって、その安定余裕の大きさを判別するために、次式の M_d を導入する。

$$M_s \geq M_d \quad (15)$$

本稿では、 M_d よりも M_s が小さい場合すなわち、安定余裕が大きい場合はデータベース駆動型制御を用いた PID パラメータを適用する。反対に、 M_d よりも M_s が大きく安定余裕が小さい場合は、次節に示す 1-パラメータチューニングにより PID パラメータを調節する。

[STEP7] 1-パラメータチューニング

安定余裕を確保するように 1-パラメータを用いて PID パラメータをチューニングする。具体的には、 k_c の値を小さくし安定余裕を十分に確保する。

3 FRIT を用いたデータベース駆動型 PID 制御のオフライン学習

3.1 FRIT

まず、FRIT について説明する。FRIT は、一回の実験によって得られた入出力データ $u_0(t)$, $y_0(t)$ および、これらのデータから生成される擬似参照入力 $\tilde{r}(t)$ を用いて、制御器の制御パラメータを直接的に算出する方法である。ここで、 $C(z^{-1})$ は制御器であり次式で表される。

$$C(z^{-1}) = c_0 + c_1 z^{-1} + \dots + c_n z^{-n} \dots \dots \dots (16)$$

ここで、 n は制御則の次数を示し、PID 制御則の場合 $n = 2$ となる。また、 $C(z^{-1})$ の入出力関係を次式に示す。

$$u_0(t) = \frac{C(z^{-1})}{\Delta} (\tilde{r}(t) - y_0(t)) \quad (17)$$

式 (8) より擬似参照入力 $\tilde{r}(t)$ は、制御器と実験データから以下のように算出される。

$$\tilde{r}(t) = C^{-1}(z^{-1}) \Delta u_0(t) + y_0(t) \quad (18)$$

また、設計者はあらかじめ所望の特性を有する参照モデル $G_m(z^{-1})$ を設計する。FRIT では、擬似参照入力 $\tilde{r}(t)$ に対する参照モデルの出力を $\tilde{y}_m(t)$ とし、 $\tilde{y}_m(t)$ と $y_0(t)$ の誤差が小さくなるような制御パラメータを決定する。

3.2 FRIT を用いたデータ駆動型 PID 制御

次に、FRIT を用いたデータ駆動型 PID 制御について説明する。まず、閉ループデータでの要求点 $\bar{\phi}_0(t)$ における PID パラメータを算出するために、式 (8) を用いて要求点とデータベース内の情報ベクトルの距離を計算し、 k 個の近傍データを選択する。次に、式 (24) を用いて以下に示す PID ゲイン $\mathbf{K}^{old}(t)$ を算出する。以下のような最急降下法を用いて、PID ゲイン $\mathbf{K}^{old}(t)$ を学習し、新たに $\mathbf{K}^{new}(t)$ を導出する。

$$\mathbf{K}^{new}(t) = \mathbf{K}^{old}(t) - \eta \frac{\partial J(t+1)}{\partial \mathbf{K}(t)} \quad (19)$$

$$\boldsymbol{\eta} := [\eta_P, \eta_I, \eta_D] \quad (20)$$

ここで、 $\boldsymbol{\eta}$ は学習係数、 $J(t+1)$ は以下で定義される評価規範を表している。

$$J(t+1) = \frac{1}{2} \varepsilon(t+1)^2 \quad (21)$$

$$\varepsilon(t) := y_0(t) - \tilde{y}_m(t) \quad (22)$$

ただし、 $\tilde{y}_m(t)$ は次のように設計される。

$$\tilde{y}_m(t) = \frac{z^{-1}T(1)}{T(z^{-1})} \tilde{r}(t) \quad (23)$$

なお、 $T(z^{-1})$ は参照モデルの特性多項式である。

$$\left. \begin{aligned} \frac{\partial J(t+1)}{\partial K_P(t)} &= \frac{\partial J(t+1)}{\partial \tilde{y}_m(t+1)} \frac{\partial \tilde{y}_m(t+1)}{\partial \tilde{r}(t)} \frac{\partial \tilde{r}(t)}{\partial K_P(t)} \\ &= -\frac{\varepsilon(t+1)G_m(1)\Delta y_0(t)}{K_I^{old}(t)} \\ \frac{\partial J(t+1)}{\partial K_I(t)} &= \frac{\partial J(t+1)}{\partial \tilde{y}_m(t+1)} \frac{\partial \tilde{y}_m(t+1)}{\partial \tilde{r}(t)} \frac{\partial \tilde{r}(t)}{\partial K_I(t)} \\ &= -\frac{\varepsilon(t+1)G_m(1)\Gamma(t)}{K_I^{old}(t)^2} \\ \frac{\partial J(t+1)}{\partial K_D(t)} &= \frac{\partial J(t+1)}{\partial \tilde{y}_m(t+1)} \frac{\partial \tilde{y}_m(t+1)}{\partial \tilde{r}(t)} \frac{\partial \tilde{r}(t)}{\partial K_D(t)} \\ &= -\frac{\varepsilon(t+1)G_m(1)\Delta^2 y_0(t)}{K_I^{old}(t)} \end{aligned} \right\} (24)$$

ここで、 $\Gamma(t)$ は次式として与えられる。

$$\begin{aligned} \Gamma(t) &= -\Delta u_0(t) - K_P^{old}(t) + K_D^{old}(t)y_0(t) \\ &\quad + \{K_P^{old} + 2K_D^{old}(t)\}y_0(t-1) \\ &\quad - K_D^{old}(t)y_0(t-2) \end{aligned} \quad (25)$$

4 数値例

4.1 制御対象と設定パラメータ

提案法の有効性を数値シミュレーションにより検証する。次式に示す Hammerstein model を対象とする。

$$\left. \begin{aligned} y(t) &= 0.6y(t-1) - 0.1y(t-2) \\ &\quad + 1.2x(t-k_r-1) - 0.1x(t-k_r-2) \\ x(t) &= 1.5u(t) - 1.5u^2(t) + 0.5u^3(t) \end{aligned} \right\} (26)$$

また、各時刻における目標値を以下のように設定した。

$$r(t) = \begin{cases} 1.0 & (0 < t \leq 100) \\ 0.5 & (100 < t \leq 200) \\ 4.0 & (200 < t \leq 300) \\ 1.5 & (300 < t \leq 400) \end{cases} \quad (27)$$

一方、初期 PID ゲインは以下の値を適用した。

$$K_P = 0.09, K_I = 0.015, K_D = 0.015 \quad (28)$$

表 1 に、提案法で用いた各種パラメータの値を示す。本稿における安定余裕のしきい値は、安定余裕を十分に確保するために、 $M_d = 0.65$ に設定した。

表 1: User-specified parameters included in the proposed scheme.

Rise time	$\sigma = 4$
Damping coefficient	$\delta = 0.1$
Orders of the information vector	$n_y=3, n_u=2$
Orders of the information vector	$m_y=2, m_u=2$
The real time-delays	$k_r=4$
Number of neighbors	$k = 6$
Learning rates offline	$n_P=2.5 \times 10^{-4}$ $n_I=1.3 \times 10^{-4}$ $n_D=1.0 \times 10^{-4}$
Learning rates online	$n_P=1.0 \times 10^{-3}$ $n_I=1.0 \times 10^{-3}$ $n_D=1.0 \times 10^{-3}$
Criteria of the stability margin	$M_d = 0.65$
Number of neighbors	$N(0) = 400$

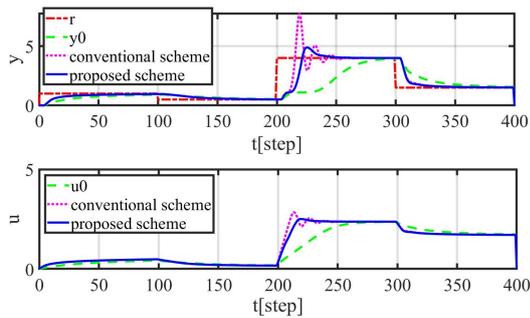


図 3: Control result by using the conventional and proposed scheme.

4.2 シミュレーション結果

図 3 は従来法と提案法の制御結果を示す。本稿の従来法とは、ロバスト性を考慮していないデータベース駆動型制御である。図 4 は閉ループ伝達関数の安定余裕を示す。図 5 は PID ゲインの推移である。図 3 を見ると、従来法では 210 [STEP] 付近で振動し始めた。そのときの $1/M_s$ は M_d よりも小さくなり安定余裕が劣化した。一方、提案法では 1-パラメータチューニングにより、 $1/M_s$ よりも M_d が大きくなるように調整し、さらにオンラインで PID ゲインを調整した。グラフからわかるように、 $1/M_s$ は M_d よりも大きくなり、制御結果は振動が抑えられた。以上より、シミュレーションにより提案法の有効性が確認された。

5 結言

本稿では、1つのデータベースから PID パラメータとシステムパラメータの双方を算出し、閉ループ伝達関数の安定余裕を考慮したデータベース駆動型 PID 制御系を提案した。提案法では、1-パラメータチューニ

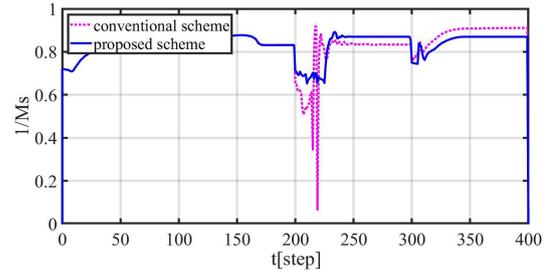


図 4: Stability margin of the conventional and proposed scheme.

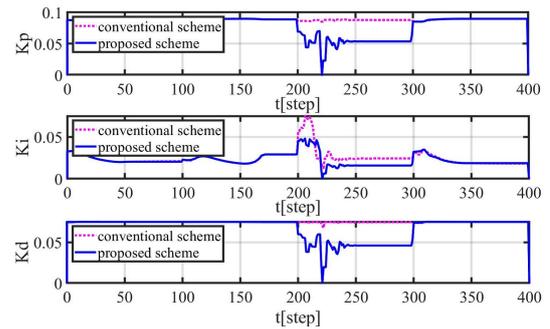


図 5: Trajectories of PID gains of the conventional and proposed scheme.

ングを用いることで安定余裕を確保でき、最適な PID パラメータの算出が可能となる。本稿では、非線形システムを線形システムに近似して安定余裕を算出しているが、本質的に非線形システムに対する安定性を議論していないため、今後検討する必要がある。また、安定余裕のしきい値についても今後検討する必要がある。なお、本研究は JSPS 科研費 16H02921 の助成を受けたものである。

参考文献

- [1] 須田：PID 制御，朝倉出版 (1992)
- [2] T. Yamamoto, K. Takao and T. Yamada: Design of a Data-Driven PID Controller, IEEE Transaction on Control Systems Technology, Vol. 17, pp.29-39(2009)
- [3] 相馬，金子，藤井：一回の実験データに基づく制御器パラメータチューニングの新しいアプローチ-Fictitious Reference Iterative Tuning の提案：システム制御情報学会論文誌, Vol. 17, No.12, pp.528-536(2004)
- [4] 西田，脇谷，中本，山本：FRIT を用いたデータ駆動型 PID 制御系の一設計：電気学会研究会資料, pp.31-36(2013)
- [5] 黒川，佐藤，荒木，小西：FRIT を利用した感度関数に基づく PID 制御系設計：電気学会研究会資料, pp.105-109(2018)