

Particle Swarm Optimizationによるデータベース駆動型 システム同定法の改良

徳田 誠*, 宮本沙季**, 岩本光生*

Improvement of Data-Base Based System Identification by Particle Swarm Optimization

Makoto TOKUDA*, Saki MIYAMOTO** and Mitsuo IWAMOTO*

Abstract

In the process systems such as the chemical plants, designs of the control systems might be difficult because of their nonlinearities and/or time variations. In this paper, a system identification method in order to obtain local linear models has been proposed for the nonlinear time-variant systems. In particular, the traditional data-base based system identification method is improved by particle swarm optimization. Finally, effectiveness of the proposed method is numerically evaluated through an application of the typical benchmark model with nonlinearity.

1 緒言

化学プラントなどに代表されるプロセス系がもつ特性の1つに、非線形特性がある。システムが強い非線形特性を有する場合、大域を線形関数で近似すると、モデル化誤差が大きくなり過ぎるため、そのモデルを利用して制御系を設計しても、良好な制御性能が得られない場合がある。そこで、非線形関数や局所線形モデルでシステムを表現するための様々な手法が提案されている。

近年、非線形システムの振る舞いを、データベース（以下、DBと略記）に蓄積されたデータを用いて、局所線形モデルに逐次同定する手法（Just-In-Time, Model-On-Demand, Lazy Learningなど）が注目されている^{1)~6)}。また、制御系設計への応用として、システムパラメータやPIDパラメータを推定する手法が提案されている^{2), 4), 5)}。特に、文献2)の手法（以下、従来法と略記）は、DBから選択された複数のデータセットを用いてシステムパラメータを推定するだけでなく、システム変動に対応するために最急降下法によりシステムパラメータをオンライン修正するなど、優れた汎用性を有している。しかし、一般的なDB駆動型システム同定法は、モデリングの必要が生じるたびにDBからデータセットを選択し、その修正を行うことから多くの

処理時間を要する。したがって、サンプリング時間を考慮していない従来法のアルゴリズムには、実用性において問題がある。

ところで、近年、群をなす魚や鳥の行動パターンを最適化手法に取り入れたParticle Swarm Optimization（以下、PSOと略記）が注目されている^{7)~11)}。これは、群れを形成している各個体が、ある目的に適合するために、自己の経験と群れで共有する情報の両方に基づいて良好な解を探索する手法であり、設計パラメータが少ないなどの利点から、ニューラルネットワークにおける誤差逆伝播法の代わりに用いられるなど、様々な分野に応用されている¹¹⁾。

そこで、本論文では、従来法の問題点を改良した新しいDB駆動型システム同定法を提案する。PSOを導入することにより、モデル化誤差を絶対値で評価できるため、微小なモデル化誤差の修正に優れているなど、最急降下法を用いる従来法よりデータの修正時間を短縮できる。これにより、サンプリング時間に関する設計条件の緩和が期待できる。また、サンプリング時間内に、所望の同定性能を有するモデルを取得できなかった場合の処理方法など、サンプリング時間を考慮したアルゴリズムを設計することにより、実用性の向上を図っている。最後に、代表的な非線形モデルに適用することにより、提案手法の有効性を定量的に検証する。

*情報工学科

**専攻科生産システム工学専攻

2 PSOを用いたDB駆動型システム同定法

2.1 システムの記述

本手法で扱うシステム (同定対象) は, 実際には非線形特性および時変特性を有しているが, モデリングの必要 (要求点) が生じたとき, 局所的に次式の離散時間線形モデルで記述できると仮定する。

$$y(t) = q^T(t)x(t) \quad (1)$$

ここで, $y(t)$ は, 時刻 t におけるシステムの出力信号を表している。また $q(t)$ と $x(t)$ は, それぞれシステムの状態を示した要求点と, その係数からなるパラメータベクトルであり, 次式で表される。

$$q(t) = \begin{bmatrix} -y(t-1) & \cdots & -y(t-n_y) \\ u(t-1) & \cdots & u(t-n_u) \end{bmatrix}^T \\ = \begin{bmatrix} q_1(t) & q_2(t) & \cdots & q_{n_y+n_u}(t) \end{bmatrix}^T \quad (2)$$

$$x(t) = \begin{bmatrix} a_1(t) & \cdots & a_{n_y}(t) \\ b_0(t) & \cdots & b_{n_u}(t) \end{bmatrix}^T \quad (3)$$

ただし, $u(t)$ はシステムの入力信号であり, n_y と n_u は, それぞれ y と u の次数で既知とする。

2.2 システムの同定法

2.2.1 初期DBの作成

まず, 同定するシステムの過去の入出力データから, 初期DBを作成する。DBに蓄えられた i 番目のデータセットは, 次式で表される。

$$D_i = \{\phi_i, \theta_i, \delta_i\} \quad (4)$$

ここで, $i = 1, 2, \dots, m_0$ であり, m_0 は初期DBに蓄積されているデータセット数である。また, 情報ベクトル ϕ_i は, 次式のような信号で構成される。

$$\phi_i = \begin{bmatrix} -y(t-1) & \cdots & -y(t-n_y) \\ u(t-1) & \cdots & u(t-n_u) \end{bmatrix}^T \\ = \begin{bmatrix} \phi_{i,1} & \phi_{i,2} & \cdots & \phi_{i,n_y+n_u} \end{bmatrix}^T \quad (5)$$

各 ϕ_i に対応したパラメータベクトル θ_i は, 以下の手順により作成される。はじめに, システムの入出力データに対して最小二乗法を施し, 大域モデルを作成する。次に, θ_i に最小二乗法で得られたシステ

ムパラメータベクトルを与える。すなわち, この段階では, $\theta_1 = \theta_2 = \cdots = \theta_{m_0}$ であり, これを後述する PSOにより, 適宜修正する。さらに, PSOでは, 同一のパラメータ (初期位置) で構成される集団に対しては, 最適値の探索が行えないため, θ_i の各要素に微小な乱数を付加する。 δ_i は, 過去の要求点と i 番目の情報ベクトルとの距離を積算したものであり, 初期値はすべて 0 とする。なお, 2.3において, δ_i を用いたデータの捨捨方法を記述する。

2.2.2 モデリングの実行条件

提案手法では, 次式が成立しないとき, 要求点を生成しモデリングを行う。

$$|y(t) - \hat{y}(t)| \leq \varepsilon |y(t)| \quad (6)$$

ここで, $\hat{y}(t)$ は, 本同定法によるシステム出力の予測値を示している。また, ε はモデル化誤差の許容範囲を示す設計パラメータであり, $0 < \varepsilon \leq 1$ とする。

2.2.3 近傍ベクトルの選択

要求点 $q(t)$ に対して, その近傍にある s 個の情報ベクトル ϕ_i をDBから抽出する。具体的には, まず, 次式で与えられる重み付き L_1 ノルムにより, $q(t)$ とDB内全ての ϕ_i との距離を求める。

$$d(q(t), \phi_i) = \sum_{j=1}^{n_y+n_u} \left| \frac{q_j(t) - \phi_{i,j}}{\max_{\alpha} \phi_{\alpha,j} - \min_{\beta} \phi_{\beta,j}} \right| \quad (7)$$

ここで, $i = 1, 2, \dots, m$ であり, m は $q(t)$ が与えられたときのDB内のデータセット数である。また,

$\max_{\alpha} \phi_{\alpha,j}$ と $\min_{\beta} \phi_{\beta,j}$ は, それぞれDB内の情報ベクトルのうち, j 番目の要素の最大値と最小値を表している。次に, 算出された距離のうち小さいものから s 個を取り出し, その情報ベクトルを要求点 $q(t)$ に対する近傍ベクトルとする。

2.2.4 PSOによるモデリング

本手法では, PSOを用いて要求点 $q(t)$ に対応する適切なパラメータベクトル $x(t)$ を求め, 非線形システムの局所線形モデルを作成する。まず, PSOにおける各個体の初期位置を $x_l^{(0)}$ と表記し, 近傍ベクトルに対応したパラメータベクトルを用いる。例えば, DB内の a 番目の情報ベクトル ϕ_a が近傍ベクトルとして選ばれた場合, a 番目のパラメータベクトル θ_a が初期位置の 1 つとなる。したがって, 個体番号 $l = 1, 2, \dots, s$ である。また, 各個体の位置 $x_l^{(k)}$ と速度 $v_l^{(k)}$ は次式により更新される。

$$x_l^{(k)} = x_l^{(k-1)} + v_l^{(k)} \quad (8)$$

$$v_l^{(k)} = \gamma^{(k)} \cdot v_l^{(k-1)} + c_p \{x_{p,l}^{(k-1)} - x_l^{(k-1)}\} + c_g \{x_g^{(k-1)} - x_l^{(k-1)}\} \quad (9)$$

ここで、 k は探索回数であり、 $x_{p,l}^{(k-1)}$ と $x_g^{(k-1)}$ は、それぞれ l 番目の個体の $k-1$ 回目までの探索における最良解と、 $k-1$ 回目の探索における集団の最良解を示している。さらに、重み c_p と c_g は、これまでの研究結果から、それぞれ $[0, 2]$ と $[0, 4 - c_p]$ の乱数で与えられる。また、 $\gamma^{(k)}$ は慣性項係数であり、一般的には次式で与えられることが多い。

$$\gamma^{(k)} = \gamma_{\max} - \frac{\gamma_{\max} - \gamma_{\min}}{k_{\max}} k \quad (10)$$

γ_{\max} 、 γ_{\min} 、および k_{\max} は、それぞれ慣性項係数の最大値、最小値、最大探索回数を示しており、多くの先行研究から、 $\gamma_{\max} = 0.9$ と $\gamma_{\min} = 0.4$ を用いることが推奨されている。しかし、探索できる回数が未知であるため、サンプリング時間を考慮したシステムに、(10)式を適用することはできない。そこで、本手法では、慣性項係数を次式で与える。

$$\gamma^{(k)} = \gamma_{\max} - \frac{\gamma_{\max} - \gamma_{\min}}{(T_s - t_1)} t_2 \quad (11)$$

ここで、 T_s はサンプリング時間を表し、 t_1 と t_2 は、それぞれサンプリング処理からデータ探索までに要した時間と、サンプリング処理から現在までに経過した時間とする。

本手法では、各個体の位置を、次式の適合度で評価する。

$$J_l^{(k)}(t) = \frac{|y(t) - \hat{y}_l^{(k)}(t)|}{|y(t)|} \times 100 \quad (12)$$

$$\hat{y}_l^{(k)}(t) = q^T(t) x_l^{(k)} \quad (13)$$

PSOは、 $J_l^{(k)}(t)$ の最小化に基づいて最適化を行うが、実際にはサンプリング時間を考慮する必要があるため、次のサンプリング時間に到達するまでに、(6)式を満足するパラメータベクトル $x(t)$ が取得できなかった場合は、適合度が最小の $x_{p,l}^{(k-1)}$ を、要求点 $q(t)$ に対する $x(t)$ とする。ただし、この場合の $q(t)$ と $x(t)$ は、データベースに追加されない。

以上をまとめると、PSOは次の手順に従って処理される。

- [1] 各個体の初期位置 $x_l^{(0)}$ を決定
- [2] 各個体の適合度 $J_l^{(k)}(t)$ を算出
- [3] $x_{p,l}^{(k-1)}$ と $x_g^{(k-1)}$ を取得
- [4] 各個体の位置 $x_l^{(k)}$ と速度 $v_l^{(k)}$ を更新
- [5] 次のサンプリング時間になるか、(6)式を満足するまで、[2]~[4]を反復
- [6] パラメータベクトル $x(t)$ を取得

2.3 データの追加と削除

前節で得られたパラメータベクトル $x(t)$ は、そのときの要求点と共に、 $m+1$ 番目のデータセットとしてDBに登録される。ただし、データセット数が m_{\max} に到達した場合には、まず、初期DBに蓄えられていたデータセットを優先的に破棄し、続いて対応する δ_i の大きい順にデータセットを削除する。これにより、過去の全要求点から最も距離が離れた(類似していない)データセットが順次消去される。

3 数値計算例

提案手法の同定精度を検証するために、本手法を代表的な非線形システムであるHammersteinモデルに適用する。Hammersteinモデル⁴⁾は次式で表される。

$$\begin{cases} y(t) = 0.6y(t-1) - 0.1y(t-2) \\ \quad + 1.2w(t-1) - 0.2w(t-2) \\ w(t) = 1.5u(t) - 1.5u^2(t) + 0.5u^3(t) \end{cases} \quad (14)$$

数値計算では、CPUとメモリがそれぞれ1.7[GHz]と1[GB]の計算機、ならびにMath Works社のMATLABを使用した。

(14)式で与えられるHammersteinモデルの静特性を図1に示す。

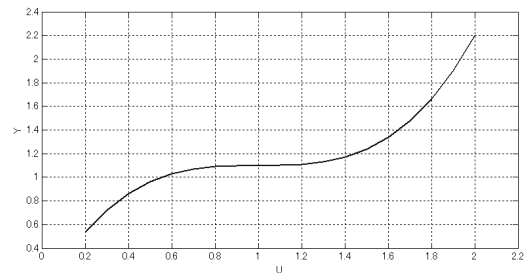


図1 (14)式の静特性

図1における横軸 U は、同定対象に与えたステップ入力の大きさで、縦軸 Y はステップ応答の最終値を示している。図1より、(14)式は、0.5から1.5の周辺において強い非線形性をもつことが分かる。

提案手法によって得られた局所線形モデルの静特性を図2に示す。初期DBを作成する際には、平均0, 分散9.0の擬似乱数を入力信号として与え、設計パラメータは、それぞれ $n_y = 2$, $n_u = 3$, $m_o = 300$ とした。また、サンプリング時間はプロセス系では比較的短い1 [sec]とした。さらに、モデリングの際に必要な設計パラメータは、それぞれ $\varepsilon = 0.05$, $s = 10$, $m_{\max} = 800$ とした。図2より、大域において良好な同定精度が得られていることが分かる。

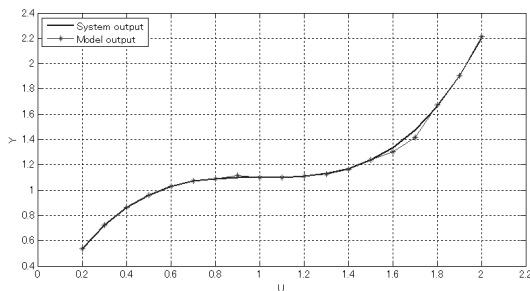


図2 提案手法による局所線形モデルの静特性

次に、非線形時変システムへ適用する。具体的には、(14)式で与えられるHammersteinモデルに対して時刻 $t = 250$ [step] にシステム変動を発生させる。システム変動後の同定対象は次式で表される。

$$\begin{cases} y(t) = 0.2y(t-1) - 0.2y(t-2) \\ \quad + 1.2w(t-1) - 0.6w(t-2) \\ w(t) = -1.5u(t) + 0.5u^2(t) + u^3(t) \end{cases} \quad (15)$$

システム変動後の同定対象の静特性を図3に示す。

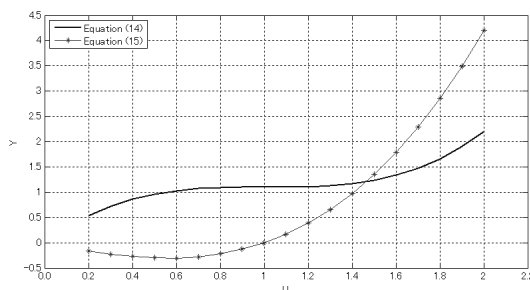


図3 システム変動前後の同定対象の静特性

(15)式で与えられる同定対象は、(14)式の同定対象とは異なり、 -0.5 から1の周辺に強い非線形性をもつことが分かる。システム変動が発生した直後のDB内には(14)式のシステムに対する過去の入出力データが蓄積されているため、PSOによるデータ修正機能がない場合、システム変動後の同定精度は著

しく低下すると考えられる。上記の非線形時変システムに本手法を適用した結果を図4に示す。

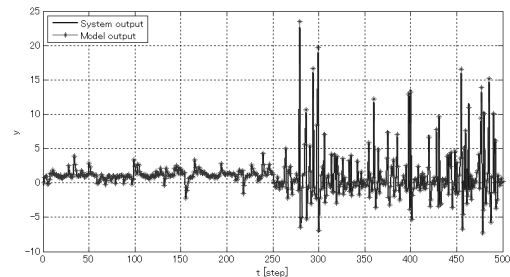


図4 提案手法の動特性

図4は、平均1.0, 分散0.5の擬似乱数を同定対象に500 [step]印加した結果得られる動特性を示している。この結果より、同定対象の出力値と出力予測値がほぼ一致していることがわかる。ここで、図4で示された本手法のモデル化誤差率を測定するために次式を導入し、図5にその様子を示す。

$$e(t) = \frac{|y(t) - \hat{y}(t)|}{|y(t)|} \times 100 \quad (16)$$

図5から、システム変動の前後共に、設定したモデル化誤差率の許容範囲である $\pm 5\%$ 以内にあり、時変系に対してもPSOの働きにより良好にシステム同定が行われていることが分かる。

以上により、本手法の有効性が定量的に示された。

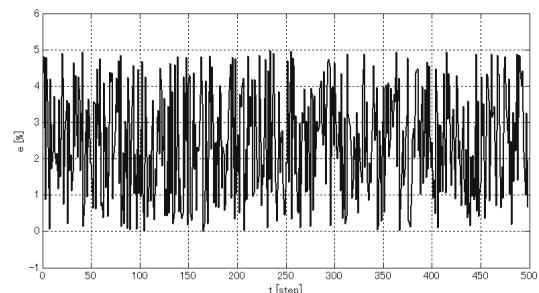


図5 モデル化誤差率

4 結言

本論文では、非線形時変システムにおける局所線形モデルを取得するための新しいDB駆動型システム同定法を提案した。数値計算を通して、サンプリング時間が比較的短い動作環境においても、良好な同定精度が得られることが示された。

今後は、本手法によって得られた局所線形モデルのシステムパラメータを用いて、制御系を設計することを検討している。

参考文献

- 1) A.Stenman, F.Gustafsson, and L.Ljung, "Just in time models for dynamical systems", 35th IEEE Conference on Decision and Control, pp.1115-1120, 1996
- 2) 高尾, 山本, 雛元, "Memory-Based型システム同定法による一般予測制御の一設計", 電学論C, Vol.125, No.3, pp.422-449, 2005
- 3) 井上, 山本, "近傍の分割抽出に基づくメモリベース推定の改良", システム制御情報学会論文誌, Vol.18, No.10, pp.368-372, 2005
- 4) 大田, 山本, "PID制御器のデータベース駆動型チューニング", 計測自動制御学会論文集, Vol.40, No.6, pp.664-669, 2004
- 5) 牛田, 木村, "Just-In-Timeモデリング技術を用いた非線形システムの同定と制御", 計測と制御, Vol.44, No.2, pp.102-106, 2005
- 6) 内田, 大貝, 伊藤, "大規模データベースオンラインモデリング-高炉への適用-" 計測と制御, Vol.44, No.2, pp.107-111, 2005
- 7) 伊藤, 田中, "Particle Swarm Optimizationの高次元問題における探索性能の改善", 計測自動制御学会論文集, Vol.42, No.5, pp.557-579, 2006
- 8) 越野, 村田, 木村, "Particle Swarm Optimizationの改良とポートフォリオ選択への応用", 電子情報通信学会論文誌, Vol.J89-A, No.1, pp.48-60, 2006
- 9) 緒方, 小野, 中山, "Particle Swarm Optimizationによる複数回探索法の基礎検討", システム制御情報学会論文誌, Vol.19, No.8, pp.330-332, 2006
- 10) 井出, 安田, "適応型Particle Swarm Optimizationに関する基礎的検討", 電学論C, Vol.124, No.2, pp.550-557, 2004
- 11) 伊藤, 田中, "Particle Swarm Optimizationによるニューラルネットワークの学習に関する研究", 平成17年電気学会電子・情報・システム部門大会論文集, pp.1087-1089, 2005