

適応性と汎化性を考慮した自己組織化適応制御器

湊原 哲也[†] 古川 徹生^{†a)}

The Self-Organizing Adaptive Controller for Adaptability and Generalization

Tetsuya MINATOHARA[†] and Tetsuo FURUKAWA^{†a)}

あらまし 人間は環境の変化に対して柔軟に対応できる能力をもち、環境に応じた適切な行動を起こすことができる。本論文は、そのような高い適応性と汎化性をもつ制御器の開発への足がかりとして自己組織化適応制御器 (Self-Organizing Adaptive Controller : SOAC) を提案するものである。提案手法は予測器と制御器を対としたモジュール構造をもち、環境の変化に対する速やかな適応性と未学習の状況にも対応できる汎化性をもつ。シミュレーションでは類似手法との比較によって提案手法の有効性が確認できた。また物理パラメータの変化する倒立振り子へ応用を試みたところ、安定した制御が可能であっただけでなく担当モジュールが物理パラメータの変化に合わせて位相情報を保ちながら切り換わり、パラメータ変化の可視化に成功した。

キーワード 自己組織化マップ, mnSOM, 適応制御, フィードバック誤差学習, 倒立振り子

1. ま え が き

人間は環境の変化に対して柔軟に対応できる能力をもち、環境に応じた適切な行動を起こすことができる。しかも、未経験の状況においても過去の経験を踏まえてそれなりの対応ができる。すなわち人間は高い適応能力と汎化能力をもつ。では人間のような高い適応能力と汎化能力をもつ制御器を構築するにはどうすればよいであろうか。これが我々の掲げる目標である。具体的には、(1) 制御対象の特性の突然の変化に対応し得る制御器の構築 (2) できるだけ少ない標本数からの汎化的な制御能力の獲得を目指している。

まず (1) に関して、このような制御器の構築には、モジュール構造をもつ学習システムを用いると有効である。学習によって獲得される性質の異なるモジュールを互いに切り換えあるいは組み合わせることによって複数の環境や制御対象に対して速やかな適応性を示すことが可能となる。

Narendra ら [1] や Wolpert ら [2] は予測器と制御器を対にしたモジュール構造をもつ手法を提案してい

る。両者の違いは Narendra らのモデルが予測誤差を最小にした予測器と対になっている制御器を用いて対象を制御する方法であるのに対して、Wolpert らのモデルでは予測誤差の soft-max 関数による“責任信号”をもとに責任信号の大きさに比例して各モジュールの出力と学習の重み付けを行う。本提案手法においても予測器と制御器を対にしたモジュール構造を採用しており、環境の変化に対して速やかな対応ができる。提案手法は Furukawa ら [3], [4] の modular network SOM (mnSOM) を制御系に応用したものであり、予測器と制御器が自己組織的に獲得される特徴をもつという点において従来手法と異なる。そのため、提案手法はシステムの類似度に基づいた自己組織化マップを生成することができる。すなわち似ているシステムはマップ空間上の近くに配置され、逆に似ていないシステムはマップ空間上の遠くに配置される。このことから提案手法はシステムの可視化、隠れパラメータ空間の発見、モデル解析といった利点をもつ。

次に (2) に関してであるが、一般に獲得したい環境や制御対象の数が増えるほど必要な学習量は増加する。したがって、少ない標本から汎化的な制御能力を獲得できることは学習時間を削減できるという意味で有効である。提案手法は mnSOM をもとにしていることで、少ない標本からそれらの内挿によって“中間的なシステム”を多数獲得できる。従来法では必要な環境

[†] 九州工業大学大学院生命体工学研究科・脳情報専攻, 北九州市
Department of Brain Science and System Engineering,
Kyushu Institute of Technology, 2-4 Hibikino, Wakamatsu-
ku, Kitakyushu-shi, 808-0196 Japan

a) E-mail: furukawa@brain.kyutech.ac.jp

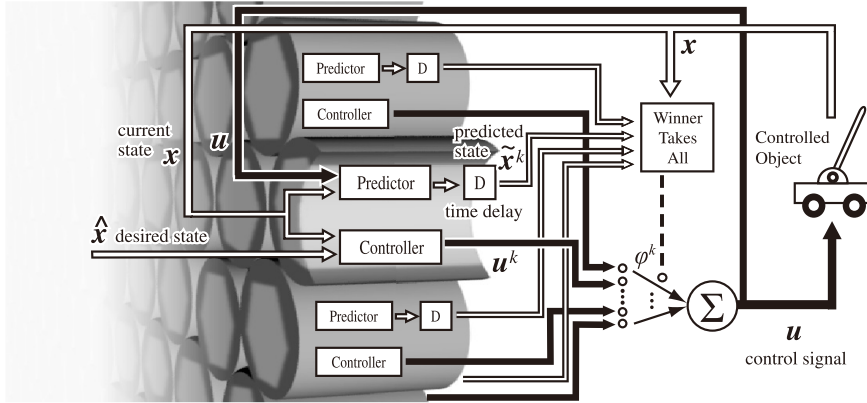


図 1 自己組織化適応制御器の基本構成図

Fig.1 Scheme of the self-organizing adaptive controller.

や制御対象の数だけモジュールを用意しておきそれらについてののみしか学習ができない点、提案手法の方が汎化性の高いシステム記述ができる。

本論文では (1), (2) を実現する新しい制御手法 (自己組織化適応制御器 (Self-Organizing Adaptive Controller : SOAC)) を提案し, Wolpert らの多重順逆対モデル (Multiple Paired Forward-Inverse Models : MPFIM) との比較を行った。その結果, SOAC は未学習データに対して従来法よりも高い汎化性をもつことが分かった。また SOAC を物理パラメータの変化する倒立振り子系へ適用したところ, 制御対象のパラメータ変化に対して位相情報を保ったまま担当モジュールが適応的に切り換わり, パラメータ変化の可視化に成功した。

2. Self-Organizing Adaptive Controller (SOAC)

2.1 基本構成

SOAC は予測器 (predictor) と制御器 (controller) を対とするモジュールを格子状に配置した構造をもつ (図 1)。 k -th モジュールの予測器は, 制御対象 (controlled object) の現在の状態 (current state) $x(t)$ と制御信号 (control signal) $u(t)$ を入力とし, Δt 秒後の制御対象の状態の予測値 (predicted state) $\hat{x}^k(t + \Delta t)^{(\text{注1})}$ を出力する。すなわち,

$$\hat{x}^k(t + \Delta t) = p_f^k(x(t), u(t)) \quad (1)$$

と表されるとする。予測器はどのモジュールを使用するかの指標を与える。一方, k -th モジュールの制御器

は, 制御対象の現在の状態 $x(t)$ と目標の状態 (desired state) $\hat{x}(t)$ を入力とし, 制御信号 $u^k(t)$ を出力する。

$$u^k(t) = c_f^k(x(t), \hat{x}(t)) \quad (2)$$

SOAC は, 学習モードと実行モードの二つのモードをもつ。学習モードでは, すべての予測器と制御器^(注2)を学習アルゴリズムに従って学習する。実行モードでは, 学習の完了したモジュールを用いて実際に対象を制御する。

2.2 学習モード (予測器)

本節では, 予測器の学習について説明し, 制御器の学習は次節で説明する。今, 事前に I 個の既知な制御対象があり, これらを学習に使用するものとする。したがって, これらを制御する制御器も I 個用意する。よって I 個の時系列データ $\{x_i(t), u_i(t)\} (i = 1, \dots, I)$ が得られる。

予測器の学習アルゴリズムは mnSOM のアルゴリズムと等しい [3], [4]。したがって, 予測器のアルゴリズムは mnSOM と同様に, 評価過程, 競合過程, 協調過程, 適応過程の四つの過程からなる。ここで, 予測器は MLP (Multi-Layer Perceptron) であると仮定し, 重みベクトルを ${}^p w^k$ とする。

[評価過程] まず各予測器の出力と訓練データの誤差を I 個すべてに対して求める。

(注1): 本論文では, 上付き文字は SOAC に関するインデックス (モジュール番号), 下付き文字はデータに関するインデックス (クラス番号) を表す。

(注2): すべての予測器を予測器マップ (predictor-map), すべての制御器を制御器マップ (controller-map) と称する。

$$pE_i^k = \frac{1}{T} \int_0^T \|x_i(t) - \tilde{x}_i^k(t)\|^2 dt \quad (3)$$

ここで、 $\tilde{x}_i^k(t)$ と pE_i^k はそれぞれ i -th 訓練データに対する k -th 予測器の出力と平均予測誤差である．また T は時系列の長さを表す．

〔競合過程〕 予測誤差を求めた後、すべての訓練データについて最適モジュール BMM (Best Matching Module : BMM) を決める．BMM は平均予測誤差を最小とするモジュールにより決定される．

$$*_i = \arg \min_k pE_i^k \quad (4)$$

〔協調過程〕 近傍関数によって学習分配率 ψ_i^k を決める．

$$\psi_i^k = \frac{\exp[-\|\xi_i^k - \xi_i^*\|^2 / 2\sigma^2]}{\sum_{i'=1}^I \exp[-\|\xi_i^k - \xi_{i'}^*\|^2 / 2\sigma^2]} \quad (5)$$

ここで、 ξ_i^k, ξ_i^* はそれぞれ k -th モジュールと i -th 訓練データに対する BMM のマップ空間における座標を表す．また σ は近傍関数の半径を決めるパラメータで、学習回数とともに単調減少する．

〔適応過程〕 協調過程で得られた学習分配率 ψ_i^k を用いて予測器の結合加重を次式によって更新する．

$$\Delta p w^k = -\eta \sum_{i=1}^I \psi_i^k \frac{\partial pE_i^k}{\partial p w^k} \quad (6)$$

ただし η は学習率を表す正の定数である．

これら 4 過程をネットワークが定常状態になるまで繰り返す．その結果、似た性質をもつモジュールがマップ空間上の近い位置に配置される．すなわち制御対象のダイナミクスの違いを反映した自己組織化マップが生成される．

2.3 学習モード（制御器）

SOAC の制御器として利用可能なものの一つとして Kawato によって提案されたフィードバック誤差学習 (feedback-error-learning) がある [5]．フィードバック誤差学習を用いることの利点は、従来型の線形フィードバック制御器 (Conventional Feedback Controller : CFC) を用いて訓練することができ、事前に最適な制御器を決定する必要がないこと、したがって追加学習も可能になることである．ここではフィードバック誤差学習を用いた場合についてのみ説明する．

SOAC のモジュールとしてフィードバック誤差学習を導入したモデルを図 2 に示す．制御器は CFC

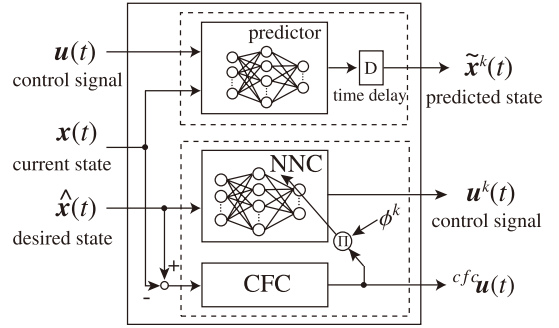


図 2 モジュールのブロック線図
Fig. 2 Block diagram of a module.

とニューラルネットワーク制御器 (Neural Network Controller : NNC) から構成され NNC は CFC と学習分配率の重み付き出力を誤差信号として教師あり学習を行う．制御器の学習は制御と同時にされるため、学習則は次節で併せて説明する．

2.4 実行モード

実行モードでは、まず制御対象の挙動 $x(t)$ と各予測器が予測した挙動 $\tilde{x}^k(t)$ との誤差を次式で定義する．

$$p_e^k(t) = (1-\varepsilon)p_e^k(t-\Delta t) + \varepsilon \|x(t) - \tilde{x}^k(t)\|^2 \quad (7)$$

ここで $p_e^k(t)$ は指数減衰平均誤差である．すなわち実行モードでは、ごく近い過去から現在までの予測誤差の時間平均をとる．時間平均をとる区間は $0 < \varepsilon \leq 1$ で決まり、 ε の値が小さいほど時間平均をとる区間は長くなる． ε の大きさは制御対象に加わる外乱やノイズの大きさによって決まる．一般に外乱やノイズが大きいかほど ε は小さくする方がよい．

次に BMM を求める．BMM は $p_e^k(t)$ を最小とするモジュールであり添字を*とすれば、

$$*(t) = \arg \min_k p_e^k(t) \quad (8)$$

と表せる．BMM が決まったら各モジュールのいわゆる“責任信号”を次式で求める．

$$\phi^k = \frac{\exp[-\|\xi_i^k - \xi^*\|^2 / 2\sigma_\infty^2]}{\sum_{k'=1}^K \exp[-\|\xi_{k'} - \xi^*\|^2 / 2\sigma_\infty^2]} \quad (9)$$

ここで、 σ_∞ は近傍半径の最終値 (学習終了時の値) を表す定数である．この式は、BMM が最も責任信号が大きく、マップ空間において BMM からの距離が遠いモジュールほど責任信号が小さくなることを意味する．

最終的に、制御対象へ与えられる制御信号は、各

NNC の出力と責任信号の積和に CFC の出力 ${}^{cf}c_u(t)$ を加えることで得られる^(注3) .

$$u(t) = \sum_{k=1}^K \phi^k u^k(t) + {}^{cf}c_u(t) \quad (10)$$

$$u^k(t) = {}^cf f^k(\hat{x}(t)) \quad (11)$$

$${}^{cf}c_u(t) = {}^{cf}cW(\hat{x}(t) - x(t)). \quad (12)$$

ただし, ${}^{cf}cW$ はフィードバック係数行列である. このとき, NNC の学習則は重みベクトルを ${}^cw^k$ として, ${}^{cf}c_u$, ϕ^k を用いて次のように表される.

$$\Delta {}^cw^k = \eta \cdot \phi^k \frac{\partial {}^{cf}f^k}{\partial {}^cw^k} \cdot {}^{cf}c_u \quad (13)$$

2.5 パラメータマップ

提案手法は先に述べたとおり予測器と制御器を対にしたモジュールを基本要素としており, 予測器と制御器が自己組織的に獲得される特徴をもつ. もちろん, このままでも機能するが, 制御対象に関する情報 (物理パラメータ) をモジュールの要素に追加することで, 後で議論する事前情報を用いた BMM のフィードフォワード選択や物理パラメータの推定などのシステム解析に利用できる.

今, 制御対象に関する物理パラメータ (例えば長さや重さ) をベクトル p とし, i -th 訓練データに対するパラメータベクトルを p_i と表す. このとき各モジュールが表現するパラメータベクトル $\tilde{p}^k$ は学習モードで最終的に得られた学習分配率 ψ_i^k と p_i を用いて次式で定義される.

$$\tilde{p}^k = \sum_{i=1}^I \psi_i^k p_i \quad (14)$$

この $\tilde{p}^k$ の集合 $\{\tilde{p}^1, \dots, \tilde{p}^K\}$ をパラメータマップ (parameter-map) と呼ぶことにする. パラメータマップは制御器マップと同様に予測誤差をもとに計算された学習分配率を用いるために特別な学習は必要なく, 予測器マップの生成に伴って自動的に生成される.

3. シミュレーション

3.1 MPFIM との比較

類似手法である MPFIM との性能比較を行うために以下に示すばね・マス・ダンパ系を用いてシミュレーションを行った.

$$M\ddot{x} + B\dot{x} + Kx = u \quad (15)$$

シミュレーションでは 9 種類の制御対象を用いて MPFIM と SOAC の学習を行い, 6 種類の対象を用いてテストを行った. 使用したパラメータは質量 $M = 1.0$ [kg] (一定) とし減衰係数 B [kg/s] とばね定数 K [kg/s²] についてのみ可変とした (図 3 (a) 参照). 目標軌道 (desired trajectory) は Ornstein-Uhlenbeck 過程を用いて作成した (図 3 (a)). サンプル間隔は 1,000 [Hz] で 30 [s] のデータを作成した. CFC のフィードバック係数は ${}^{cf}cW = [k_x, k_{\dot{x}}, k_{\ddot{x}}] = [5, 10, 0.5]$ とした. CFC のみを用いたときの応答を図 3 (b) に示す. 予測器は入力を位置, 速度, 制御信号とし出力を加速度とした. 制御器は位置, 速度, 加速度を入力とし制御信号を出力するものとした. 予測器, 制御器ともに 2 層の線形ネットワークを用いた. 予測器と制御器は MPFIM と SOAC で同じ構造のものを用いた. MPFIM のモジュール数は訓練データ数と同じ 9 とし, SOAC では SOM の次元を 2 として $9 \times 9 = 81$ とした. MPFIM におけるスケーリングパラメータ σ は学習中には手動で設定し, 実行中は固定した ($\sigma = 0.1$). 一方 SOAC におけるスケーリングパラメータ (近傍半径) は学習中は指数関数的に単調減少させ, 実行モードでは固定した ($\sigma_\infty = 1.8$). 学習は 9,000 [s] (30 [s] $\times$ 300) とした. シミュレーションは四次の runge-kutta 法によって実現した. 刻み幅は $h = 0.001$ [s] とした.

学習後ネットワークを固定して MPFIM と SOAC による制御をそれぞれ行った. その結果 SOAC の軌道誤差は MPFIM よりも小さくなった (図 3 (d)). なお, モジュール数を変えた場合についても同様の検証を行ったが, いずれの場合においても SOAC の方が制御性能が優れていた (図 3 (c)).

3.2 倒立振り子への応用

3.2.1 物理パラメータの変化の可視化

SOAC の特徴としてシステムの類似度に基づく自己組織化マップの生成が挙げられる. したがって SOAC は SOM のような可視化機能をもつ. すなわち似ているシステムはマップ空間上の近くに集まり, 逆に似てないシステムは遠い位置に配置される. ここでは SOAC がもつシステムの可視化機能についての検証を行うために倒立振り子を用いてシミュレーションを行った. シミュレーションで用いた倒立振り子の運動方程式を付録 1. に, パラメータを図 4 に示す.

(注3): 式 (10) の代わりに BMM の制御器の出力を制御入力とする方法も考えられる.

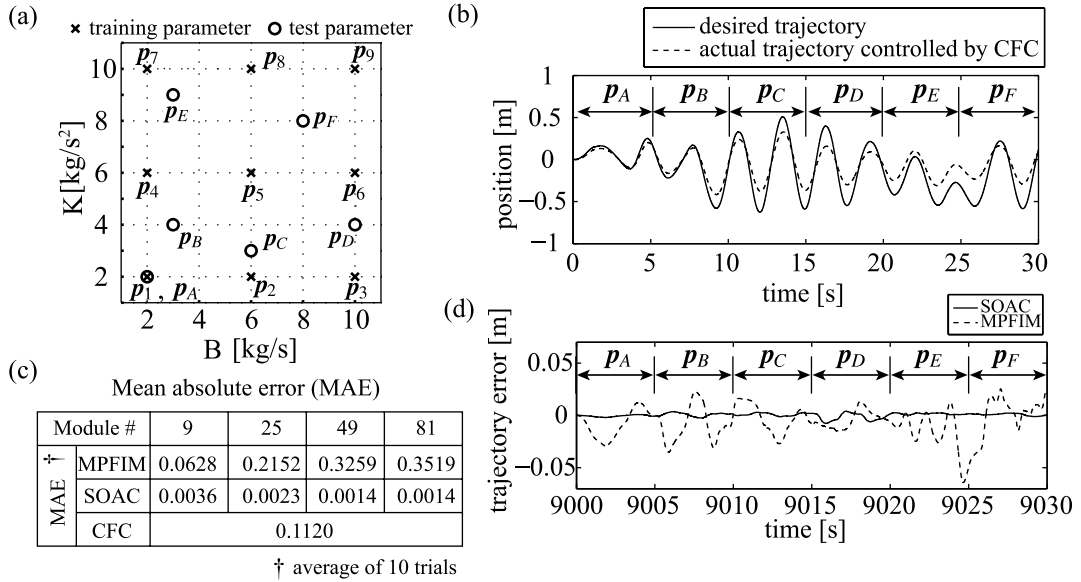


図 3 SOAC と MPFIM を用いたばね・マス・ダンパ系の応答
Fig. 3 Responses of the spring-mass-damper system using SOAC and MPFIM.

SOAC の予測器は $x(t)$ と $u(t)$ を入力 (入力層素子数 5) とし, 0.01 秒後の状態の予測値を出力とするものとした (出力層素子数 4). ただし予測器は 2 層の線形ネットワークとした. 制御器については, モジュール選択によるシステムの可視化に焦点を絞るために, あらかじめ用意された解析的なものを使用することにした. 具体的には線形二次レギュレータを用いて予測器が同定したシステムについて次式を最小化するようなフィードバック係数を定めた (付録 2.).

$$\int_0^\infty (x^T Q x + u^T R u) dt \quad (16)$$

モジュール数は $9 \times 9 = 81$ とした. 学習モードでは予測器についてのみ学習を行い, その後はネットワークを固定した. 制御入力 BMM の制御器の出力を用いた. 実行モードでは図 4 に示すパラメータを 10 秒ごとに変化させたときの応答を調べた. ただし台車の目標位置を 10 秒ごとに変化させ, 合計 90 秒間の応答を調べた.

学習モードで得られた結果を図 5 に示す. 図中の各格子内の波形は SOAC が同定したシステムの閉ループ系に対する振り角度 θ のインパルス応答を表している. 倒立振り子は不安定系であるから SOAC が同定したシステムそのもののインパルス応答はとれないために, CFC を含んだフィードバック系のインパルス応

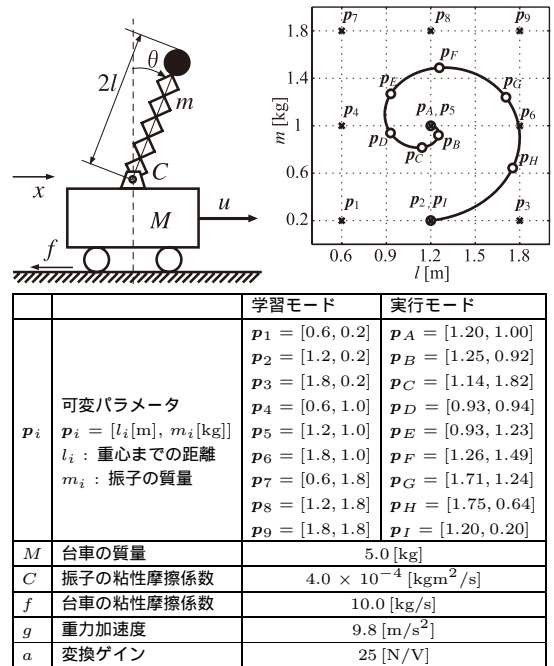


図 4 振り子の長さ及び重さが変化する倒立振り子. シミュレーションで用いたパラメータを表に示す.

Fig. 4 The inverted pendulum with variable mass and arm length. The relevant parameters are shown in the table.

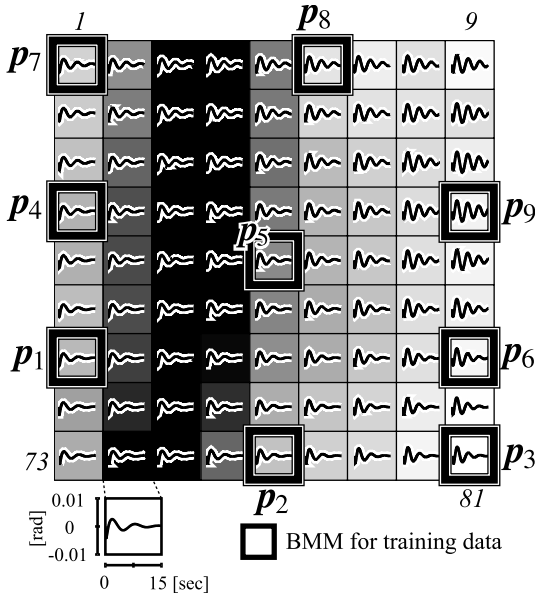


図 5 学習により得られた予測器マップ。図中の波形は CFC を含んだ閉ループ系に対する各モジュールのインパルス応答を表す。ここで CFC のフィードバック係数は ${}^{cfc}W = [k_x, k_\theta, k_{\dot{x}}, k_{\dot{\theta}}] = [-0.25, -5.64, -0.67, -2.03]$ とした。また図中の濃淡は訓練データに対する各モジュールの出力とその 8 近傍との平均誤差を表しており、淡い色ほど近傍との特性が似ていることを意味する。

Fig. 5 A predictor-map generated by SOAC. Each waveform represents an impulse response to a closed-loop system involving CFC, where CFC is ${}^{cfc}W = [k_x, k_\theta, k_{\dot{x}}, k_{\dot{\theta}}] = [-0.25, -5.64, -0.67, -2.03]$. The grayscale indicates the mean error for training data between a module and the 8 neighborhoods, i.e. the lighter the color is, the more similar the property is.

答を示している。波形は同じ CFC に対する応答を表しており、波形の違いがシステムの違いを表現している。波形を見比べると振子振動の減衰の遅い特性をもつモジュールは右上の方に集まり、逆に振子振動の減衰の早いシステムがマップ空間上の左下に集まっておりシステムの違いを反映したマップが形成されているのが分かる。

次に実行モードによって得られた振子の応答波形を図 6 に、そのときの BMM の遷移の様子を図 7 に示す。SOAC の制御器は解析的に用意されたものであるが、パラメータの変化する倒立振子を制御するためには適切なモジュール選択が行われなければならない。しかし図 7 を見ると担当モジュールが長さ及び重さのパラメータとの位相を保ったまま選択された。すな

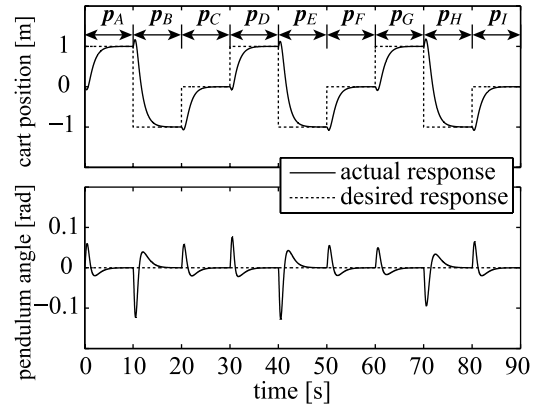


図 6 倒立振子の応答波形。振子の長さや重さのパラメータが 10 秒ごとに变化する。

Fig. 6 Sample response waveforms (actual, desired) of the inverted pendulum system. The mass and arm length are changed every ten seconds.

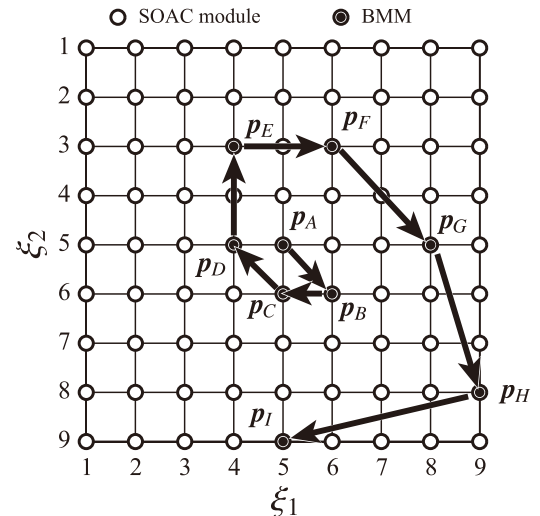


図 7 倒立振子の物理パラメータの変化の可視化

Fig. 7 Visualization of the changing physical parameters of the inverted pendulum.

わち振子のパラメータがゆっくりと変化する場合には担当モジュールの変化もゆっくり変化（たかだか 1 モジュールの移動）し、徐々にパラメータの変化が大きくなると担当モジュールも遠くのモジュールへと切り換わった。その結果安定した制御を実現できた（図 6）。

3.2.2 SOAC を用いたシステム解析例

ここでは SOAC によるシステム解析の例としてパラメータ推定を行う。具体的には 3.2.1 で扱った倒立振子を用いて、与えられた倒立振子からその長さ及び重さの推定を試みた。

今、学習では用いていない長さと重さをもつ振子 ($p = [1.60, 1.50]$) を用意し、SOAC (3.2.1の結果と同じもの) を用いて制御することを考える。このときに振子の長さや重さの情報は与えずに、位置、速度などのセンサ情報から予測誤差に基づき BMM を決定する。

実際に選ばれた BMM がもつパラメータベクトル $\tilde{p}^*$ を調べたところ、振子のパラメータ推定値は $\tilde{p}^* = [1.65, 1.54]$ となり、このときの実際のパラメータ p とのユークリッド距離 (誤差) はわずかに $\|p - \tilde{p}^*\| = 0.064$ であった。また同様の推定を 50 種類のパラメータ (訓練データで用いたパラメータの中で最大の長さや重さ、及び最小の長さや重さを頂点にもつ長方形内からランダムに選んだ) に対して行ったところ平均推定誤差は 0.1052 であった。このことからパラメータ推定の有効性が確認できた。

4. 討 論

4.1 SOAC と MPFIM の比較

4.1.1 近傍関数について

SOAC と MPFIM の本質的な違いは近傍関数の定義の仕方であり、SOAC が SOM の近傍関数に従っているのに対し、MPFIM はエントロピー最大化の考えに基づいている [6]。そのために SOAC ではデータを補間するような“中間的なシステム”を得ることができるが、MPFIM ではそのような機能は備わっていない。これはエントロピー最大化に基づく手法がデータの中心 (クラスタの中心) を表現するようなアルゴリズムになっているためであり、データが存在していない領域を表現することは基本的にできないためである。

4.1.2 汎化性と適応性について

文献 [7] で MPFIM の未学習データに対する汎化性が議論されているが、これは学習の完了したモジュールの出力の線形内分によって未学習データに対応できるという意味であり、未学習データを表現したモジュールが形成されるわけではない。上でも述べたようにエントロピー最大化による方法より SOM の近傍関数の方が内挿の機能は優れているために SOAC の方が MPFIM よりも高い汎化性をもつと考えられる。このことはシミュレーション結果からもうかがうことができる。

また SOAC では追加学習が可能であり学習モードの内挿では不十分だった制御対象に対しても制御しながら追加学習を行うことで環境に適応することが可能

であり、この点 MPFIM よりも高い適応性をもつと考えられる。

4.2 パラメータマップの有効性

式 (17) で得られるパラメータマップは BMM のフィードフォワード選択とパラメータ推定の二つの用途で用いることができる。一つ目は、制御対象に関する見た目のパラメータ (長さや重さ) が事前情報として与えられる場合に、初期の BMM を次のように決定する、一種のフィードフォワードの選択法としてパラメータマップを用いる方法である。

$$*(0) = \arg \min_k \|p - \tilde{p}^k\|^2 \quad (17)$$

式 (7) のような予測が正しかったかどうかの事後評価によって BMM を決定するフィードバックの選択法では一般に時間遅れが生じるが、パラメータマップを用いることでより素早い BMM の切換が可能となる。ただし式 (4) と式 (17) では距離尺度が異なるために一般に同じマップとにならないことに注意する必要がある。したがって、初期段階での BMM はパラメータマップを用いて決定し、その後は予測誤差が最小となった真の BMM を用いて制御することが望ましい。

二つ目のパラメータ推定は制御対象に関するパラメータが未知のときに観測可能なセンサ情報から観測できないパラメータを推定する問題に適用できる。シミュレーションでは倒立振子の位置、速度の観測情報を用いて倒立振子の長さや重さの推定を行い、その有効性が確認できた。複数のモデルを用意してそれらの単純な切換で制御を行う従来の手法に比べて、提案手法を用いることで今制御対象がどういった状況や環境にあるかを把握しながら制御できる。このことは危険予測などにおいて重要であることから、SOAC を用いたパラメータ推定は有効であると考えられる。

5. む す び

本論文では、自己組織化適応制御器 (SOAC) を提案し、従来法との比較及び倒立振子系への適用を試みた。シミュレーションの結果、SOAC は従来法よりも高い汎化性をもつことが分かった。また物理パラメータの変化する倒立振子に対して長さ、重さの位相情報を保ったまま担当モジュールが適応的に切り換わり、安定した制御が可能であったのみならずパラメータ変化の可視化に成功した。

今後の課題としては、SOAC の安定性に関する考察と非線形課題への応用、そしてオンライン学習へ向け

たアルゴリズムの修正などが挙げられる．

謝辞 本研究は九州工業大学 21 世紀 COE プログラム「生物とロボットが織りなす脳情報工学の世界」及び科研費基盤 (C) (課題番号 17500193) の支援を受けて行われた．

文 献

- [1] K.S. Narendra, J. Balakrishnan, and M.K. Ciliz, "Adaptation and learning using multiple models, switching, and tuning," IEEE Control Syst. Mag., vol.15, no.3, pp.37–51, 1995.
- [2] D.M. Wolpert and M. Kawato, "Multiple paired forward and inverse models for motor control," Neural Netw., vol.11, pp.1317–1329, 1998.
- [3] T. Furukawa, K. Tokunaga, S. Kaneko, K. Kimotsuki, and S. Yasui, "Generalized self-organizing maps (mnSOM) for dealing with dynamical systems," Proc. International Symposium on Nonlinear Theory and its Applications, pp.231–234, Fukuoka, Japan, Nov.-Dec. 2004.
- [4] K. Tokunaga, T. Furukawa, and S. Yasui, "Modular network SOM : Self-organizing maps in function space," Neural Information Processing - Letters and Reviews, vol.9, no.1, pp.15–22, 2005.
- [5] M. Kawato, "Feedback-error-learning neural network for supervised motor learning," in Advanced Neural Computers, ed. R. Eckmiller, pp.365–372, Elsevier, North-Holland, 1990.
- [6] K. Rose, F. Gurewitz, and G. Fox, "Statistical mechanics and phase transitions in clustering," Phys. Rev. Lett., vol.65, no.8, pp.945–948, 1990.
- [7] M. Haruno, D.M. Wolpert, and M. Kawato, "MO-SAIC model for sensorimotor learning and control," Neural Comput., vol.13, no.10, pp.2201–2220, 2001.

付 録

1. 倒立振子の運動方程式

シミュレーションで扱った倒立振子の運動方程式は式 (A.1), (A.2) に従う．

$$(M + m)\ddot{x} + ml \cos \theta \cdot \ddot{\theta} - ml\dot{\theta}^2 \sin \theta + f\dot{x} = a \cdot u \quad (\text{A.1})$$

$$ml \cos \theta \cdot \ddot{x} + (I + ml^2)\ddot{\theta} - mlg \sin \theta + C\dot{\theta} = 0 \quad (\text{A.2})$$

ここで, x を台車の位置 [m], $\dot{x}$ を速度 [m/s], $\ddot{x}$ を加速度 [m/s²], θ を振子の角度 [rad], $\dot{\theta}$ を角速度 [rad/s], $\ddot{\theta}$ を角加速度 [rad/s²] とする．状態変数は $x = [x, \theta, \dot{x}, \dot{\theta}]^T$ とし, 制御変数は u とした．なお, M は台車の質量 [kg], m は振子の質量 [kg], l は振子の重心までの距離 [m], f は台車の粘性摩擦係数

[kg/s], C は振子の粘性摩擦係数 [kgm²/s], g は重力加速度 [m/s²], a は信号を力に変換するゲイン [N/V], I は慣性モーメントで, $I = ml^2/3$ [kgm²] である．

なお, シミュレーションは四次の runge-kutta 法 (刻み幅 $h = 0.01$) によって実現した．出力トルクには疑似乱数によるガウス分布のノイズ $N(0, dt)$ が各シミュレーション時刻 $dt = 0.01$ ごとに入るとした．

2. フィードバック係数の求め方

線形二次レギュレータによるフィードバック係数は次式で与えられる．

$${}^{cf}W^k = R^{-1}B^kP^k \quad (\text{A.3})$$

ここで P^k は (4×4) の対称行列で次の Riccati 型行列方程式の唯一な正定値の解である．

$$(A^k)^T P^k + P^k A^k + Q - P^k B^k R^{-1} (B^k)^T P^k = 0 \quad (\text{A.4})$$

A^k, B^k はそれぞれ k -th 予測器の結合荷重から推定したシステム行列と制御ベクトルを表す．また, $Q \in \Re^{4 \times 4}$ を単位行列とし, $R = 1$ とした．

(平成 19 年 7 月 26 日受付, 10 月 17 日再受付)



湊原 哲也

平 15 九州工業大学大学院生命体工学研究科・脳情報専攻博士前期課程了．現在, 同専攻博士後期課程在学中．



古川 徹生 (正員)

平元大阪大学大学院博士前期課程了．同年九州工業大学・情報工学部助手, 平 16 同大学院・生命体工学研究科助教授を経て, 平 18 同大学院教授, 現在に至る．博士 (工学)．日本神経回路学会会員．