

水中ロボットにおける自己組織的行動獲得システム (第一報:自己組織化マップを用いた運動制御システムの提案)

学生員 西田周平* 正員 石井和男*
古川徹生*

A Self-Organizing Decision Making System for AUVs
(First report: A Control System using Self-Organizing Map)

by Shuhei Nishida, *Student Member* Kazuo Ishii, *Member*
Tetsuo Furukawa

Summary

Autonomous Underwater Vehicles (AUVs) are attractive tools for maintenance of underwater structures and oceanography, however, there are a lot of problems to be solved such as motion control, acquisition of sensor data, decision-making, navigation without collision, self-localization and so on. In order to realize useful and practical robots, underwater vehicles should take their action by judging the changing condition from their own sensors and actuators, and are desirable to make their behavior, to adapt to the working environment. We have been investigated the application of brain-inspired technologies such as Neural Networks (NNs), Self-Organizing Map (SOM), etc. into AUVs.

The motion of AUV is represented by complicated non-linear dynamics in six degrees of freedom with added-mass and hydrodynamic forces, and control systems should be adaptive and robust. In our previous adaptive control method using NNs, a time series of state variables and control signals should be fed into the control system in order to adapt the change of dynamic property and environment, therefore, the obtained information in the previous adaptation is getting less gradually. If the environment of the robot is rapidly changed, the previous control system takes time to adapt new environment and former environmental information does not remain correctly. Therefore, a new method, which keeps the information of initial state or previous environment and adapts to new environment, should be developed to improve the efficiency of the learning and reduce the learning cost with the use of the former environmental information which the robot had already learned. A new self-organizing decision making system for AUVs using modular network Self-Organizing Map (mnSOM) proposed by Tokunaga et. al. is discussed in this paper. The proposed decision making system is developed using recurrent NN type mnSOM. The efficiency of the system is investigated through the simulations.

1. 緒言

自律型水中ロボット(AUV: Autonomous Underwater Vehicle)は、地球海洋科学調査や水中作業のための次世代ツールとして期待されており、その実用化が望まれている。AUVの実現には、運動制御、センサ情報の取得、行動決定、衝突回避、自己位置推定など様々な問題がある¹⁾。しかし、水中という極限環境は人間には容易に想像しがたく、あらかじめロボットに対して様々な状況に対応できるようにプログラミングすることは困難である。したがって人間の負担を軽

減するため、AUVの自律性の向上が期待される。ロボットが自身のセンサで状態や環境を把握し、それらに基づいて状況を判断し行動を決定することが望ましい。著者らはニューラルネットワーク(NN: Neural Network)を始めとする生物の情報処理アルゴリズムの仕組みを参考にした学習能力に注目し、水中ロボットの運動制御や行動決定システムへ応用してきた。提案手法は、非航行型の自律型水中ロボット Twin-Burger²⁾ や航行型水中ロボット PW45 を用いた実験を通じてその有効性の確認を行っている³⁾⁻⁶⁾。

AUVの運動は水中での6自由度の運動であり、複雑かつ非線形な動特性を持っている。また、ミッションの違いによる装備品、搭載機器の変化によるロボット自身の動特性の変化や、潮流の変化によるロボット周囲の環境の変化に起因する運動特性の変化は、制御システムに大きな影響を与える。ロボットが自身や環境の変化に対して適応的に行動するこ

* 九州工業大学 大学院生命体工学研究科

とができれば望ましい。文献 3)~5)で提案されたニューラルネットワークによる適応制御手法では、水中ロボットの動特性及び制御器をオンラインで獲得することができるが、一対の動特性と制御器しか持っていないために適応する(動特性と制御器を調整する)過程で、過去における学習によって得た情報の影響は低下していく。つまり、過去において適応していた動特性及び環境に戻った場合においても再学習する必要があり、常時、ロボットにおいて計測される時系列情報を追加学習しなければならない。より高速に、かつ柔軟に動特性などの変化に適応するには、過去を含めた様々な適応状態を保持しながら、新しい動特性や環境へ適応する行動獲得システムが必要となる。

人は一度獲得した技能は、新たに技能を獲得したからといって失うことはない。獲得した複数の技能に対する複数の順モデルと逆モデルに関するモジュール構造の存在の可能性が指摘されている。そのような構造の一つとして川人らは多重順逆対モデル(MOSAIC: MODular Switching And Identification for Control)として提案しており、複数の動特性と制御器の対によって制御システムを構成している。同定器の予測誤差を用いて各制御器の出力に重み付けをし、制御出力を決定することでロバストな制御の実現を行っており⁷⁾、MOSAICと強化学習を組み合わせ、2リンクの振り子の振り上げ運動の獲得に成功している⁸⁾。

文献 6)において提案した自己組織化マップ(SOM: Self-Organizing Map)⁹⁾を用いた障害物回避手法では、ロボットの周囲の距離情報と回避方向に関する局所マップを獲得し、水中ロボットの障害物回避に適用している。これと同様に、ロボットの状態及び周囲の環境の変化とロボット動特性の関係を表現したダイナミクスマップを得ることができれば、ロボットを中心とした局所環境の状態遷移を観測できる。また、そのダイナミクスマップを基にして同時に複数の動特性に対応した制御器を獲得することが可能である。

しかしながら、一般的なSOMは入力としてベクトル空間しか取り扱うことができない。さらに学習過程が教師無し学

習であるため、ロボットの時系列情報から入出力関係を学習してロボットの動特性や制御器を表現するには、競合層における(入力, 出力)の組としてベクトル表現する等の工夫が必要となる。入出力の写像関係を得るには、教師有り学習アルゴリズムを持ったニューラルネットワークが適している。ロボットの行動決定システムには、ロボットがおかれた環境を自動的に認識し把握する“教師無し”アルゴリズム、及び動特性や制御器獲得のための“教師有り”アルゴリズムの両者の考え方が必要であり、両者の長所を取り入れた情報処理技術が望まれる。

徳永らによって提案されたモジュラーネットワーク自己組織化マップ(mnSOM: modular network Self-Organizing Map)¹⁰⁾は、関数をモジュールの要素として扱えるため、自己組織化マップの教師無しアルゴリズムを継承しつつ、モジュールの要素として教師有り学習アルゴリズムを有する情報処理システムを導入することにより、上記の両者のアルゴリズムを同時に実現することが可能である。

本研究では、MLP型mnSOMを用いた水中ロボットの自己組織的行動獲得システムを提案する。mnSOMにはSOMの競合層の各ベクトルユニットを階層型ニューラルネットワークの一つである多層パーセプトロン(MLP: Multi Layer Perceptron)に置き換えたMLP型mnSOMや、SOMで置き換えたSOM²等があり、関数空間などを取り扱えるように拡張したものである。SOMの特徴である、(1)入力データの内挿の補間を得ることができること、(2)入力空間での距離関係が競合層においても保持される、等のSOMの性質を有しており、SOMの一般化といえるアルゴリズムである。提案手法の有効性を検証するため、ロボットの運動特性を競合層のマップとして表現したフォワードモデルマップ、フォワードモデルマップに対応したコントローラマップ、及びオンラインでの運動特性適応シミュレーションの結果について述べる。

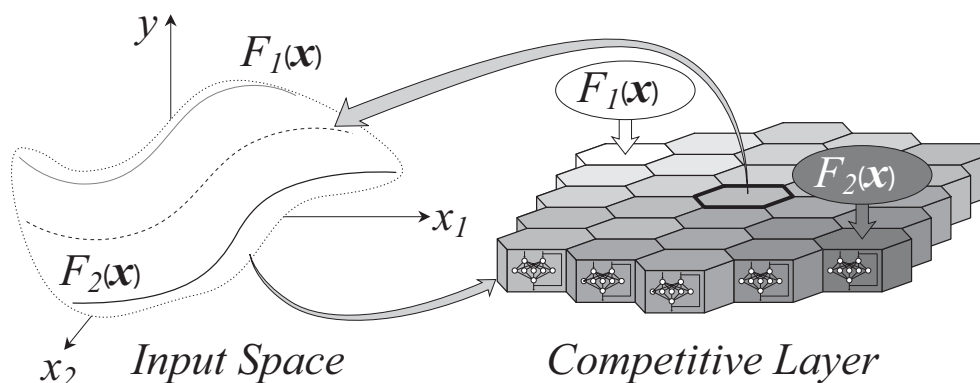


Fig.1 Concept of modular network SOM (mnSOM)

2. mnSOM (modular network SOM)

本論文において情報処理システムとして導入する mnSOM の概念図を Fig.1 に示す. mnSOM は, 入力空間において複数のデータクラスが有する入出力の写像関係(以下の(2)式)を競合層のモジュールとして表現することが可能な情報処理システムである. 競合層には通常の SOM と同様に, 例えば六角格子状や正方格子状にモジュールを配置する. 学習の過程で入力空間のデータクラスが表現している関係を教師有り学習によって獲得しながら, 並行してクラス間の評価関数から算出された類似度をもとに教師無し学習によって類似した特徴を有するクラスは近接し, 特徴が異なるクラスは遠距離に配置された競合層マップを得ることができる.

Fig.1 の左図に示される $F_1(\mathbf{x})$ と $F_2(\mathbf{x})$ は, 入力空間におけるある入出力関係を表現している. 学習後, Fig.1 の右図に示されるような競合層が得られたとする. 競合層において, $F_1(\mathbf{x})$ と $F_2(\mathbf{x})$ の中間に位置するモジュールは, 入力空間においても $F_1(\mathbf{x})$ と $F_2(\mathbf{x})$ の中間に位置する関数を表現しており, 入力空間での幾何学的関係が競合層にも保存されたマップが得られる. mnSOM の適用例として関数のパラメータ推定, 非線形主成分解析, 動的システムのマッピング, 自己組織化適応制御器などの有効性が報告されている¹¹⁾⁻¹⁴⁾.

mnSOM の学習アルゴリズムは, モジュールとして何を採用したかによって異なるが, ここでは競合層のモジュールとして多層パーセプトロン (MLP) を用いた場合の MLP 型 mnSOM の学習のアルゴリズムを以下に述べる. 次章で述べるフォワードモデルマップ及びコントローラマップの学習には以下のアルゴリズムを用いている.

M 個のデータクラス $\{D_1, D_2, \dots, D_M\}$ があり, それぞれ N 個の入出力対 $(\mathbf{x}_{ij}, \mathbf{y}_{ij})$ を持っており, データ D_i に関して写像関係は関数 $f_i()$ を用いて, 式(1), (2)で表現される.

$$D_i = \{(\mathbf{x}_{i1}, \mathbf{y}_{i1}), (\mathbf{x}_{i2}, \mathbf{y}_{i2}), \dots, (\mathbf{x}_{iN}, \mathbf{y}_{iN})\} \quad (1)$$

$$\mathbf{y}_{ij} = f_i(\mathbf{x}_{ij}) \quad (i = 1, \dots, M; j = 1, \dots, N) \quad (2)$$

mnSOM のモジュールは K 個あり, k 番目のモジュールが持つ MLP の結合加重を \mathbf{w}_k , 出力を $\hat{\mathbf{y}}_{ij}^k$ とすると, 各モジュールが表現している関数 $g_k()$ は入力 \mathbf{x} に対して式(3)として表現される. また, MLP を関数 $g()$ としてみなすと, 入力 \mathbf{x} 及び結合加重 \mathbf{w}_k の関数として表現できる.

$$\hat{\mathbf{y}}_{ij}^k = g_k(\mathbf{x}_{ij}) = g(\mathbf{x}_{ij}, \mathbf{w}_k) \quad (3)$$

このとき, 学習アルゴリズムは以下の 5 つの学習過程から構成される.

(i) 初期化: 各モジュールの \mathbf{w}_k をランダムに初期化する.

(ii) 評価過程: 評価関数(4)式を用いて, i 番目のクラス D_i の入力データ \mathbf{x}_{ij} に対する k 番目のモジュールの評価値 $E_i(k)$ を求める. $E_i(k)$ は入力クラス D_i の写像関係 $f_i()$ と $g_k()$ の関数間の距離であり, 入力クラス D_i と k 番目の競合層モジュールの類似度を意味する. 各モジュールにおいて, 全てのクラス D_i に対する評価値 (類似度) $E_i(k)$ を計算する.

$$E_i(k) = \frac{1}{N} \sum_{j=1}^N \|f_i(\mathbf{x}_{ij}) - g_k(\mathbf{x}_{ij})\|^2 \quad (4)$$

(iii) 競合過程: 式(5)に従い, 入力クラス D_i に対する評価値 (類似度) $E_i(k)$ が最小となる競合層モジュールを勝者モジュール (BMM: Best Match Module) とし, 入力クラス D_i の BMM のインデックス (モジュール番号) を c_i とする.

$$c_i = \arg \min_k E_i(k) \quad (i = 1, \dots, M) \quad (5)$$

(iv) 協調過程: 式(6)に従い, BMM からの距離および学習回数によって, 学習分配率 $\Psi_i(k)$ を決定する. $\Psi_i(k)$ は k 番目のモジュールが D_i の入出力関係を学習する量を示す. 関数 $h()$ は近傍関数であり, 競合層における BMM と各モジュールとの配置から計算される幾何学的な距離 $L(k, c_j)$ 及び学習回数 t の増加と共に単調減少する関数を選択する. 本研究ではガウス関数を用いる.

$$\Psi_i(k) = h(L(k, c_i), t) / \sum_{m=1}^M h(L(k, c_m), t) \quad (6)$$

(v) 適応過程: 各モジュールの MLP において, 誤差逆伝播法 (BP: Back Propagation) を用いて結合加重を更新する. 式(7)に示すように学習率 η と学習分配率 $\Psi_i(k)$ を用いて \mathbf{w}_k を更新する. ここで, $\eta \Psi \rightarrow \eta$ と置き換えると更新式は一般的な BP 法と同様となる.

$$\Delta \mathbf{w}_k = -\eta \sum_i (\Psi_i(k) \partial E_i(k) / \partial \mathbf{w}_k) \quad (7)$$

(ii)~(v) の過程を学習が収束するまで繰り返す. (iv) の協調課程の概念図を Fig.2 に示す. 横軸をモジュール番号, 縦軸を近傍関数及び学習分配率とする. 近傍関数にはガウス関数を用い, データクラス D_1 の近傍関数を h_1 , データクラス D_2 の近傍関数を h_2 とする. データクラス D_1 に対する学習分配率 $\Psi_1(k)$ は, 点線で示されるように, 最も適合した勝者モジュールではなく左端に位置するモジュールの分配率が最も高い値となる. これは学習が進行するに伴って, 勝者モジュールが競合層の周辺に移動することを意味している.

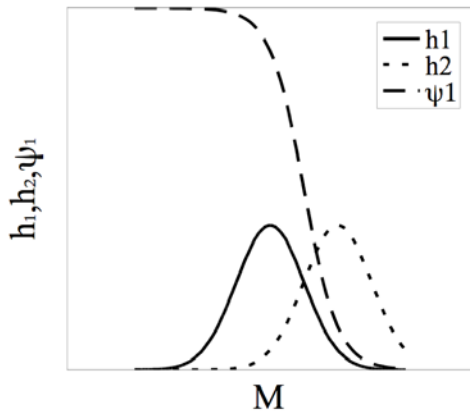


Fig. 2 Adaptation Process

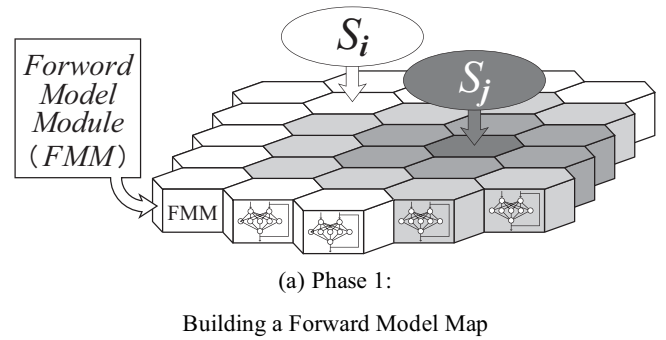
3. mnSOM を用いた自己組織的行動獲得システム

本研究では、環境やロボットの動特性の変化に応じて自己組織的にロボットの行動を獲得するシステムの開発を目標としている。本論文ではその第一段階として、動特性の変化に適応する制御システムの開発を行う。良好な制御性能を得るには、動特性の変化に素早く追従し、制御則を適応的に調整していく必要がある。ロボットの制御において扱う情報は、時系列データであるのでモジュールに MLP の一つであるリカレント型のニューラルネットワーク (RNN: Recurrent Neural Network) を用いる。学習アルゴリズムは、先に述べた MLP 型 mnSOM のアルゴリズムと同じである。RNN 型 mnSOM を用いて、状態量と操作量に関する様々な時系列データを学習させることにより、ロボットの動特性を取得しながら、異なった動特性を有するモジュール群を競合層に配置していく。本研究で提案する制御システムは、Fig.3 に示すように、ロボットの動特性を表現するフォワードモデルマップ及びこれに対応したコントローラマップの2層のRNNから構成されている。

RNN 型 mnSOM を用いた適応型制御システムは、フォワードモデルマップの作成、コントローラマップの作成、及びロボット制御への適用の3つのフェーズ構成されている。各フェーズに関して以下に述べる。

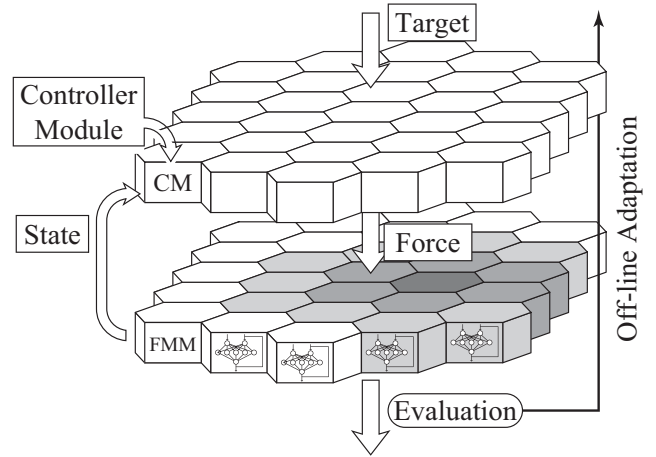
3.1 フォワードモデルマップの作成

動作環境の変化や動特性の変動、アクチュエータの特性変化や故障を想定、或は、実機により計測し、状態量と操作量の組からなるデータクラスを準備し、学習する。Fig.3(a)において、ある動特性を表現する時系列データ $S_i(x)$ は白色で



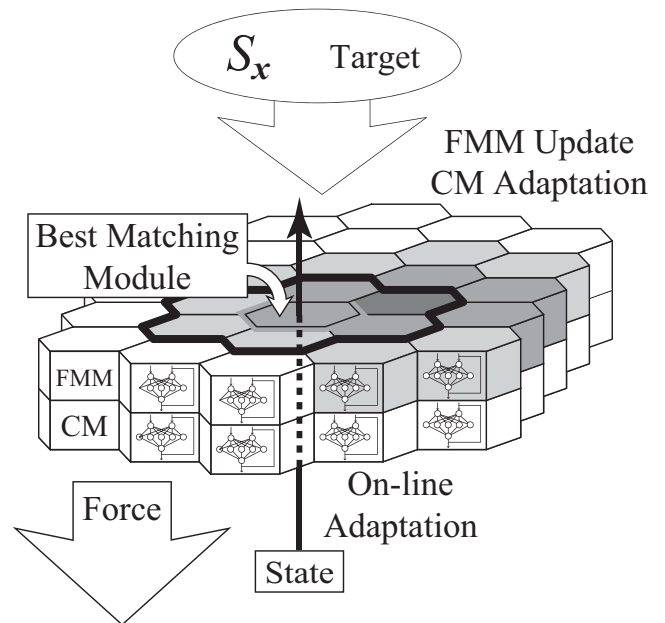
(a) Phase 1:

Building a Forward Model Map



(b) Phase 2:

Adaptation of a Controller Map using the Forward Model Map.



(c) Phase 3:

Implementation of the Control Map to Robot Control and On-line Adaptation using the Forward Model Map and the Controller Map.

Fig.3 Learning Processes of an Adaptive Controller System using RNN-mnSOM

示された競合層モジュールが勝者モジュールとして対応し、別の動特性を表現する時系列データ $S_j(x)$ は黒色の競合層モジュールが勝者モジュールとして対応している状況を示している。モジュールの色は学習した動特性の影響を意味しており、勝者モジュールの間には $S_i(x)$ と $S_j(x)$ から得られた動特性によって補完された動特性を有するモジュール (Forward Model Module: FMM) が得られる。各モジュールの学習係数及び学習回数は同じ値とし、各データクラスに対する勝者モジュールが選択され、学習が収束したと判断された段階で学習を終了し、次の学習段階に移行する。得られた学習結果は、以降、フォワードモデルマップと呼ぶ。

3.2 コントローラマップの作成

第2段階として、作成したフォワードモデルマップを用い、各モジュール (FMM) に適応した制御器を作成する。Fig.3(b)の上部に示されたコントローラマップの作成においては、競合層における幾何学的な配置は考慮せず、対応する FMM に対してコントローラを適応させるものとする。コントローラマップにおけるモジュールをコントローラネットワークモジュール (Controller Module: CM) とすると、各 CM に対して制御目標値が与えられる。対応する FMM から得られるロボットの状態量と目標値との偏差量をもとに操作量が算出され、フォワードモデルマップに送信される。フォワードモデルマップでは、操作量から計算される次の時間ステップの状態量と目標値との誤差をコントローラマップに逆伝播することにより、BP 法を用いて制御則の調整を行う。制御目標値、学習係数及び学習回数は各モジュールにおいて同じ値とし、ある程度学習が収束するまで行う。

3.3 ロボットの制御及びオンライン調整

第1段階及び第2段階の学習は、準備されたデータのオフライン学習及び調整であった。第3段階では制御対象である

Table 1 Coefficient M and C for Limit Cycle Motion

(M, C)		
$D_0 : (80, 25)$	$D_3 : (90, 25)$	$D_6 : (100, 25)$
$D_1 : (80, 50)$	$D_4 : (90, 50)$	$D_7 : (100, 50)$
$D_2 : (80, 100)$	$D_5 : (90, 100)$	$D_8 : (100, 100)$

ロボットの制御と並行して、ロボットの動作環境や制御特性の変化に対してオンライン学習による適応を行う。

ロボットのミッション遂行中において、ある一定期間サンプリングされた時系列データ (Fig. 3(c)における $S_x(x)$) をフォワードモデルマップへ入力する。フォワードモデルマップは動特性推定器として用いられ、最も動特性を表現する勝者モジュール (FMM) が選択される。ロボットの制御には、勝者となった FMM に対応するコントローラモジュールから得られる操作量が採用される。これと並行して、FMM の追加学習による動特性変化への適応、及び FMM の更新に伴うコントローラモジュールの調整が行われる。

4. 適応制御システムの構築

前章において述べた学習アルゴリズムを用いて、水中ロボットの適応制御システムを構築する。

4.1 フォワードモデルマップ

提案する適応制御システムを構築するため、第1段階としてフォワードモデルマップを作成する。フォワードモデルマップの作成のため、次式で示される水中ロボットの運動方程式を用いてリミットサイクル運動に関する時系列情報を取得した。

$$F = M\ddot{x} + C\dot{x} \quad (8)$$

ここで、 F は推進力、 \dot{x} は速度、 \ddot{x} は加速度であり、定数 M, C はそれぞれ付加質量を含んだ質量及び抗力係数である。

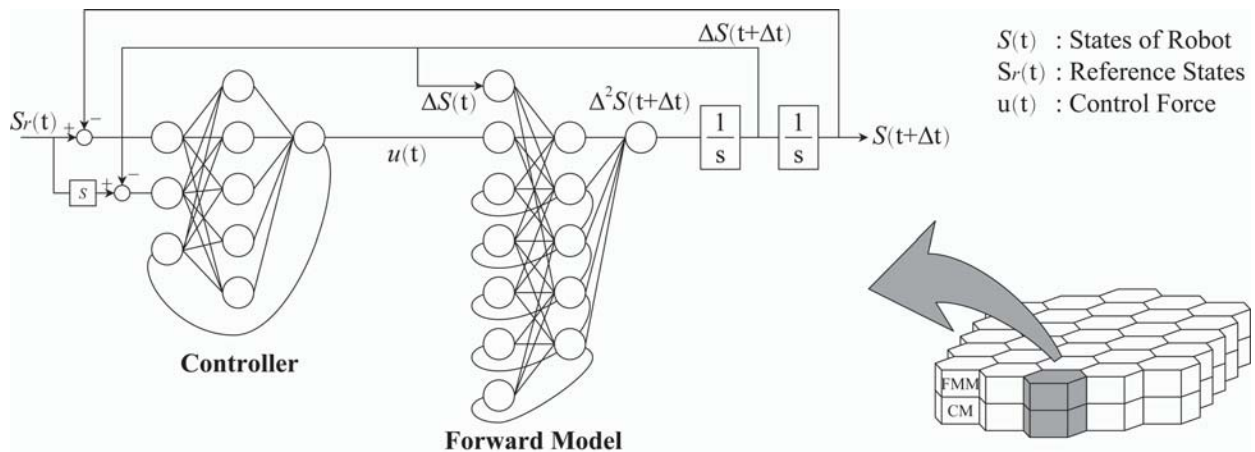


Fig.4 The Network Structure of a Forward Model Module and a Controller Module

提案する手法の有効性を検証するため、ここでは取り扱う運動は一自由度とし、他の運動モードとの連成及び干渉は動特性の変化とみなすものとした。フォワードモデルマップの教示データとして、Table 1 に示した9種類の組み合わせ ($M=80, 90, 100, C=25, 50, 100$) から時系列データ D_i ($i=0\sim 8$) を用意した。(8)式を用い、速度の絶対値が $0.2[m/s]$ を超えた時点で制御入力 $5[N]$ を反転させ、リミットサイクル運動を行わせた。一つの時系列データは、50 秒間、 $10[Hz]$ でサンプリングした 500 個の状態量と操作量の組み合わせである。一つのフォワードモデルモジュール(FMM)の構造は、Fig.4 の右半分において Forward Model として示した構造のリカレント型 MLP とし、フォワードモデルマップは 6×6 の正方

Table 2 Best Matching Module for each Data Class

D_i (Module Num. in x-axis , Module Num in y-axis)		
$D_0 : (0, 0)$	$D_3 : (4, 0)$	$D_6 : (6, 1)$
$D_1 : (0, 0)$	$D_4 : (4, 1)$	$D_7 : (6, 3)$
$D_2 : (2, 6)$	$D_5 : (4, 6)$	$D_8 : (6, 6)$

格子状に配置された競合層とした。

100,000 回の学習によって得られたフォワードモデルマップを Fig.5 に示す。入力した各データクラス D_i に対する勝者モジュールの関係を Table 2 に示す。勝者モジュールに関しては、リミットサイクル運動の同定結果を Fig. 5 に合わせて表示している。細線は速度、太線は加速度であり、横軸は時間を表している。また、フォワードモデルマップの各モジュールの色は、あるモジュールとその近接するモジュールと距離を表しており、濃い色ほど距離が大きく白色に近いほど近接するモジュールと類似していることを意味する。各入力データに対する勝者モジュールはフォワードモデルマップの周辺に配置されており、第二章において述べた学習アルゴリズムが機能していることがわかる。

フォワードモデルマップを検証するため、各モジュールに対してリミットサイクル試験を行った。各モジュールから得られた時系列データから、最小二乗法によって運動方程式のパラメータ M, C を算出して数値的に評価する。運動方程式の係数 M, C の推定を行った結果を Fig.6、時系列の速度を横軸、加速度を縦軸にとった状態推移の様子を Fig.7 に示す。Fig.6 における ■ は入力データ作成時に用いた係数、× はフォワードモデルマップから推定された各フォワードモデルモジュールのパラメータを表している。パラメータ $M-C$ 空間において、フォワードモデルマップは格子状に広がっており、入力データのパラメータに完全には一致していないものの、中間に存在するモジュールは入力データによって補完さ

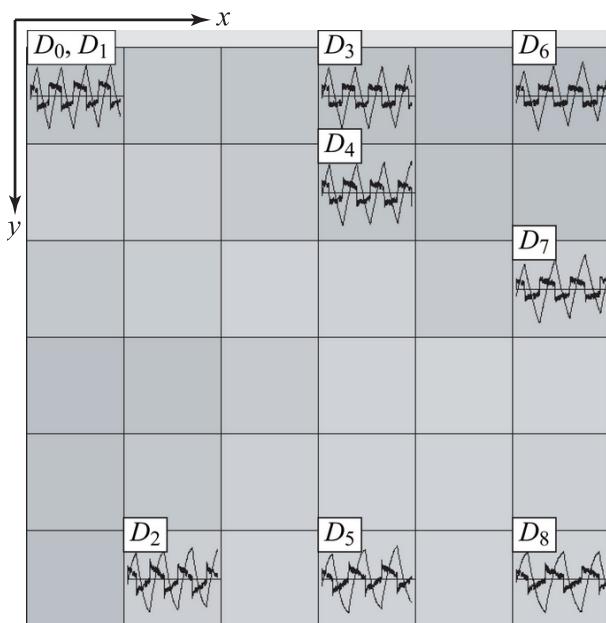


Fig. 5 A Forward Model Map Obtained from the Time Series of Limit Cycle Simulation Data.

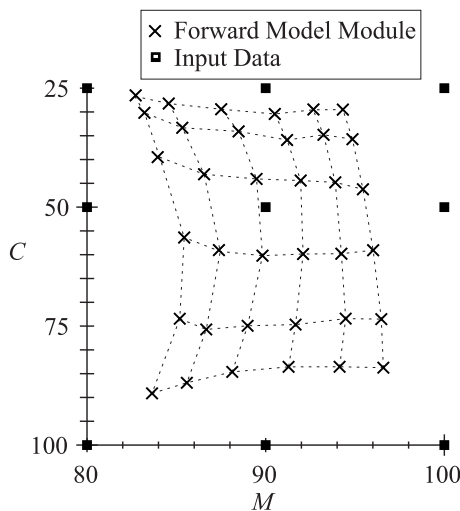


Fig. 6 Forward Model Map Evaluation in $M-C$ Space by the Least Square Method.

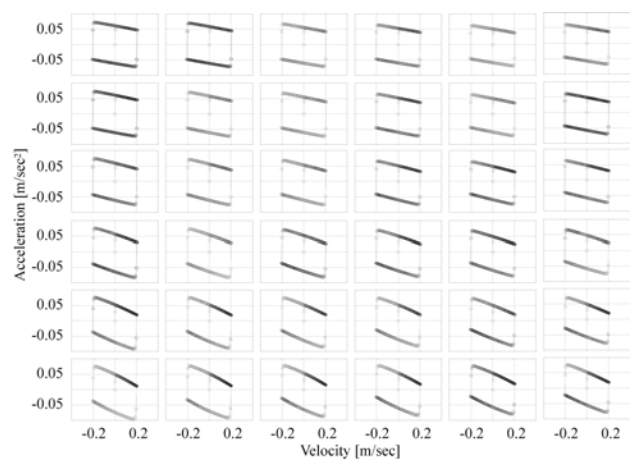


Fig.7 Acceleration-Velocity Relationship Obtained from Limit Cycle Simulation with FMMs

れた中間的な振る舞いを表現していることがわかる。

Fig.7 における状態推移の様子は、各モジュールの位置に対応している。リミットサイクルの平行四辺形は左側から右側に向かって徐々に面積が狭くなっており、 M の増加に伴って加速度が減少している。また、上段から下段に向かうにつれて傾きが急峻になっている。上から下に向かって C の値が増加し、減衰が大きくなっている。

4.2 コントローラマップの作成

前節において得られたフォワードモデルマップを用いてコントローラマップを作成する。コントローラモジュール (CM) として、Fig.4 左側の Controller と示した構造のリカレント型 MLP を用いた。各フォワードモデルモジュールとコントローラモジュールを一つの対とし、コントローラモジュールとフォワードモデルモジュールから構成される一つの大きなリカレント型 MLP としてコントローラモジュールを調整する。コントローラモジュールの結合加重は、乱数を用いて初期化し、予め設定した目標軌道にフォワードモジュールの出力が追従するようにコントローラの結合加重を調整する。両者を一つの MLP としてみなし、フォワードモデルモジュールから出力される状態量をコントローラモジュールの評価値として用いる。評価値を BP 法を通じてコントローラネットワークへ逆伝播することにより、教師無し学習による結合加重の更新が可能となる。

結合加重の更新のための評価値 E_c は、次式で定義した。

$$E_c = \frac{1}{2} \sum_7 \left[p_1 \{S_r(t) - S(t)\}^2 + p_2 \{\Delta S_r(t) - \Delta S(t)\}^2 + q_1 u(t)^2 \right] \quad (9)$$

ここで、 $S(t)$ 、 $\Delta S(t)$ は、それぞれロボットの位置及び速度、 $S_r(t)$ 、 $\Delta S_r(t)$ はそれぞれ目標位置及び目標速度を、 $u(t)$ は制御器から出力される操作量である。 p_1, p_2, q_1 はそれぞれ位置、速度、操作量に関する項の重み定数である。

コントローラ調整のための目標軌道は、時間 0~25[sec] において 0.5[m]、25~50[sec] で -0.5[m]、目標速度を 0.0[m/s] とし、サンプリングレートは 10[Hz] とした。15,000 回のコントローラモジュールの調整を行った結果を Fig.8 に示す。

コントローラモジュールの配置は、Fig.5 のフォワードモデルモジュールの幾何学的配置に対応している。Fig.8 の各要素に表示された時系列データは、目標軌道に対する制御シミュレーションの結果を示しており、横軸は時間[sec]、太線は目標値[m]、点線はロボットの位置[m]、細線は操作量 [Nx10] を表している。各コントローラモジュールは全て目標値を追従しており、フォワードモデルモジュールが有する動特性に対応した制御器を生成できている。

4.3 コントローラのオンライン調整

RNN 型 mnSOM を用いた適応制御システムについて検証するため、参考文献 5) において提案したオンラインコントローラ適応システムを比較対象とし、学習していない時系列データに対する適応能力の評価を行った。比較対象システムにおいて、初期のロボットの動特性として $(M, C) = (25, 25)$ を与えた場合の時系列データを学習することによって、フォワードモデル及びコントローラを構築した。RNN 型 mnSOM は、前節において得られた制御システムである。

教示データとして与えていない $(M, C) = (75, 75)$ 及び $(120,$

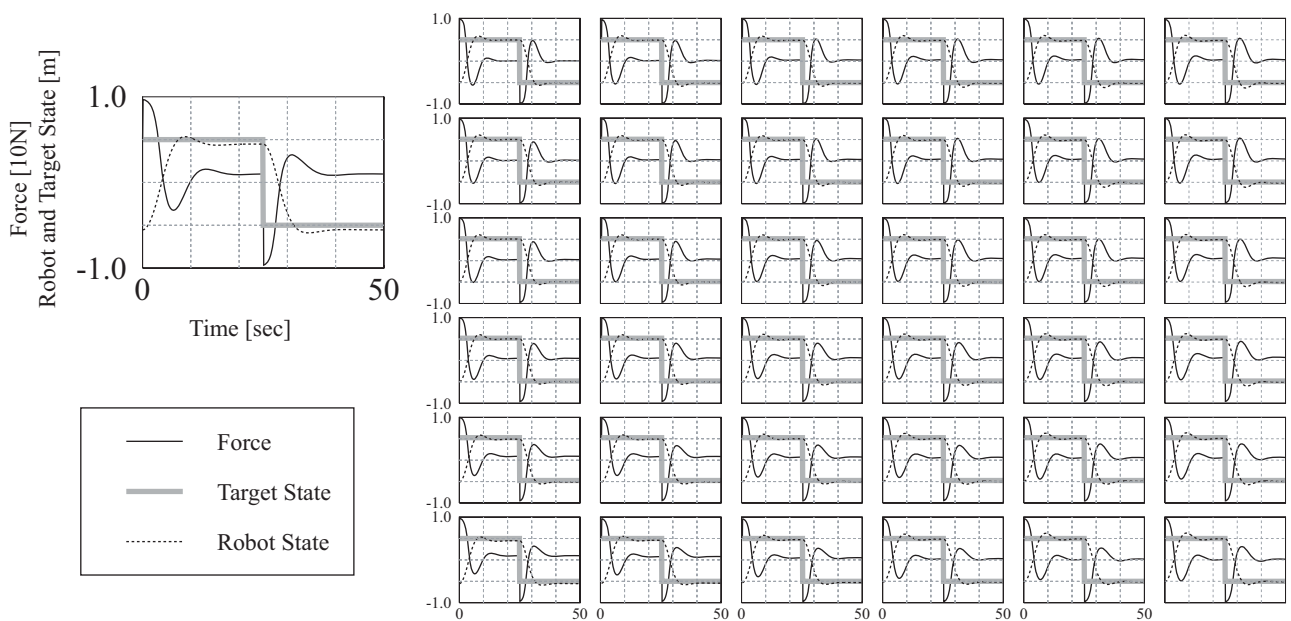


Fig.8 Controller Map: Controller Network Modules are adjusted using corresponding Forward Model Modules.

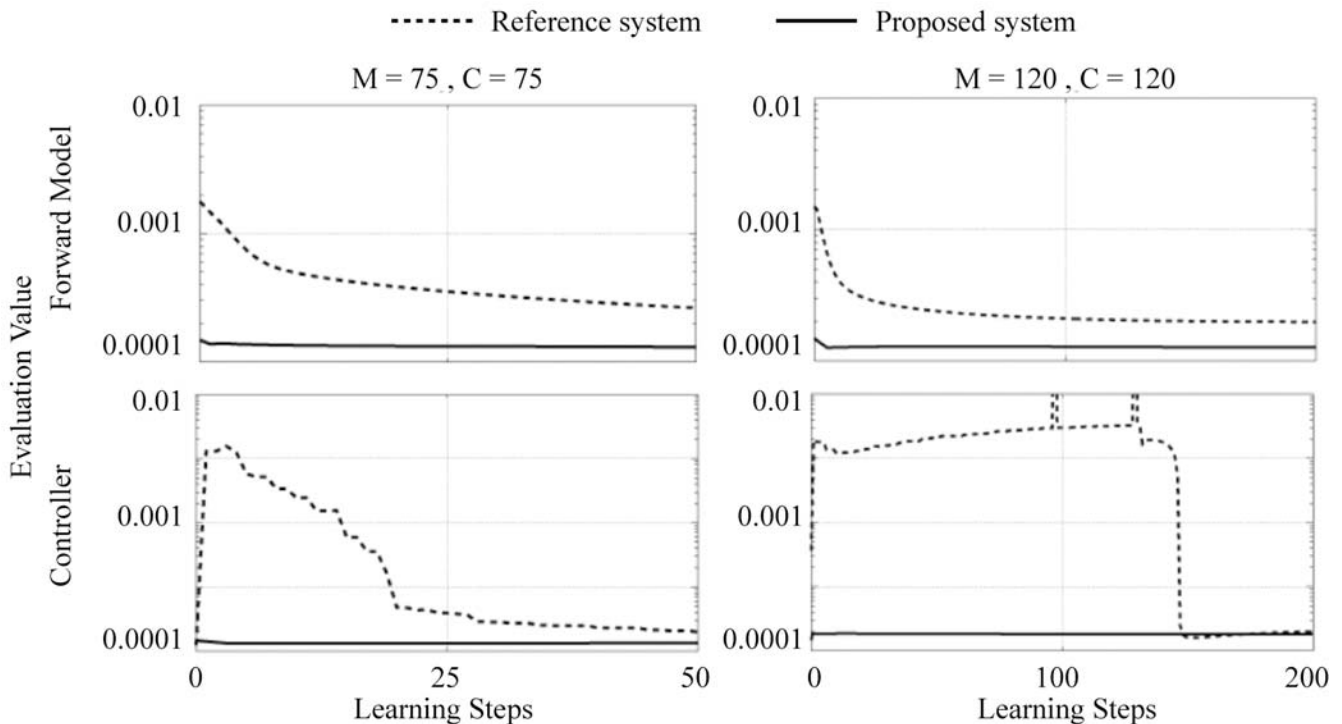


Fig.9 Transition of Evaluation Values. Solid lines are obtained from the proposed system and dot-lines are from the reference system.

120)として得られた時系列データに対する応答を Fig.9 に示す. $(M, C) = (75, 75)$ は提案する RNN 型 mnSOM において学習した領域内のパラメータ設定であり, 内挿できることが予想される. $(M, C) = (120, 120)$ は学習領域外の時系列データである. 実線が提案する RNN 型 mnSOM を用いた適応制御システム, 点線が比較対象システムの応答を表している. 上図がフォワードモデルの推定誤差, 下図がコントローラの制御誤差の推移である.

比較対象システムでは, 学習初期においてフォワードモデルの誤差が大きく, フォワードモデルが急速に変更される. フォワードモデルの変更に伴ってコントローラの制御誤差が増加している. コントローラがある程度収束するまでの学習回数は, $(M, C) = (75, 75)$ へと動特性を変化させた場合は, 約 20[step], $(M, C) = (120, 120)$ のときは約 150[step]が必要であった. これは参考文献 5)における結果と同様の結果を示している. 一方, 提案システムにおいては, 与えられた時系列を最もよく表現しているフォワードモデルモジュールが存在するため, 学習初期に若干の調整がみられるものの, 推定誤差及び制御誤差とも比較対象システムに比べて良好な適応性能を示している.

4. 結論

本論文では, RNN 型 mnSOM を用いた適応制御システムを提案した. 第一段階として, ロボットの動特性を表現するフォワードモデルマップの生成, 第二段階として作成したフ

ォワードモデルマップを用いたコントローラマップの作成, 第三段階において, 学習していない時系列データに関するオンライン調整法について評価した. フォワードモデルマップにおいて, 学習した時系列の特徴が表現されており, 学習していない入力時系列に対しても内挿が期待できる. コントローラマップにおいて, 対応するフォワードモデルモジュールの動特性に適応した制御器を自動的に生成することが可能であることを示した. 適応シミュレーションにおけるオンライン学習では, 動特性の変化に対して高速な適応が可能であることを示した. 本論文で提案する適応制御システムは, 教師有り学習と教師無し学習を組み合わせた情報処理システムであり, 入力データの入出力関係を直接表現しながら, 自己組織的に動特性の変化に対応することが可能である.

今後は, 実機を用いたフォワードモデルマップ及びコントローラマップの作成を行い, 実機の制御及びオンラインコントローラ調整実験を行っていく予定である.

謝辞

本研究の成果の一部は, 文部科学省 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」(拠点番号 J19) の支援を受けて実施した. 東京大学生産技術研究所浦環教授には貴重なご意見を賜った. ここに謝意を表します.

参考文献

- 1) 浦環, 高川真一, “海中ロボット総覧”, (1994), 成山

堂書店

- 2) T. Fujii, T. Ura, H. Chiba, Y. Nose and K. Aramaki, (1993), "Development of a versatile test-bed "Twin-Burger" toward realization of intelligent behaviors of autonomous underwater vehicles", OCEANS'93, vol.1, pp.1186-1191
- 3) K. Ishii, T. Fujii and T. Ura, (1995), "An on-line adaptation method in a neural network based control system for AUVs", IEEE Journal of Oceanic Engineering, Vol.20, No.3, pp.221-228
- 4) 石井和男, 浦環, 藤井輝夫, (1995), "ニューラルネットワークによる潜水艇の運動の同定 (その2: 学習過程の改良とコントローラ調整への適用)", 日本造船学会論文集, Vol. 177, pp. 429-435
- 5) 石井和男, 藤井輝夫, 浦環, 能勢義昭, (1997), "ニューラルネットワークによる潜水艇の運動の同定 (その3: 学習による外乱への適応)", 日本造船学会論文集, Vol. 182, pp. 469-479, 1997
- 6) S. Nishida, K. Ishii and T. Ura, (2004), "A Self-Organizing Map Based Navigation System for an Underwater Robot", IEEE International Conference on Robotics and Automation, pp.4466-4471
- 7) 川人光男, 銅谷賢治, 春野雅彦, (2000), "ヒト知性の計算神経科学<第4回> 多重順逆対モデル(モザイク) - その情報処理と可能性", 科学, Vol.70, No.11, pp.1099-1017
- 8) 鮫島和行, 銅谷賢治, 川人光男, (2001), "強化学習 MOSAIC: 予測性によるシンボル化と見まね学習", 日本ロボット学会誌 19 卷 5 号, pp.551-556
- 9) T. Kohonen, (1982), "Self-organized formation of topologically correct feature maps", Biological Cybernetics, vol.43, pp.59-69
- 10) K. Tokunaga, T. Furukawa and S. Yasui, (2003), "Modular Network SOM: Extension of SOM to the realm of function space", 3rd Workshop on Self-Organizing Maps, pp.173-178
- 11) T. Furukawa, K. Tokunaga, K. Moroshita and S. Yasui, (2005), "Modular Network SOM (mnSOM): From Vector Space to Function Space", International Joint Conference on Neural Networks
- 12) K. Tokunaga, T. Furukawa, (2005), "Nonlinear ASSOM Constituted of Auto associative Neural Modules", 5th Workshop on Self-Organizing Maps, pp.637-644
- 13) T. Furukawa, T. Tokunaga, S. Kaneko, K. Kimotsuki and S. Yasui, (2004), "Generalized Self-Organizing Maps (mnSOM) for Dealing with Dynamical Systems", International Symposium on Nonlinear Theory and its Applications, pp.231-234
- 14) T. Minatohara, T. Furukawa, (2005), "Self-Organizing Adaptive Controllers: Application to the Inverted Pendulum" 5th Workshop on Self-Organizing Maps, pp.44-48