モジュラーネットワーク型自己組織化マップを用いた水中ロボットのシステム同定

System Identification of an AUV using mnSOM

○西田　周平，石井　和男（九工大）

Shuhei NISHIDA and Kazuo ISHII, Kyushu Institute of Technology
nishida-shuhei@edu.brain.kyutech.ac.jp

Abstract: Autonomous Underwater Vehicles (AUVs) are attractive tools to survey earth science and oceanography, however, there exists 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, because of features caused by 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. 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.

Key words: AUV, Recurrent Neural Network, mnSOM

1. はじめに

自律型水中ロボット(AUV: Autonomous Underwater Vehicle)は，地球海洋科学調査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に関して研究を行い，非航行型の自律型水中ロボット Twin-Burger[2]を用いた実験を通じて有効性の確認を行ってきた[3],[4].

本報では，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)[5]を用いた水中ロボットにおける自己組織的行動決定システムを提案する．提案手法の有効性を検証するために，ダイナミクスマップの作成，及びそれらに対応したコントローラマップの作成シミュレーションについて報告する．

2. mnSOM を用いた自己組織的行動決定システム

2.1. mnSOM とは

mnSOM とは Kohonen によって提案された SOM[6]の各ベクトルユニットを MLP (Multi Layer Perceptron)や SOM などで置き換えて，関数空間などを取り扱えるように拡張したものである．入力データの内挿の補間を得ることができること，入力空間での距離関係が出力空間でも保存されるなどの SOM の性質を失っていないことから，SOM の一般化といえるアルゴリズムである．

mnSOM の構造を図 1 に示す．入力空間にある規則 f_i を持って広がっているデータクラス $\boldsymbol{D}_i$ の関係を出力空間へ写像することができる．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などにおいて有効性が報告されている．[7]-[9]

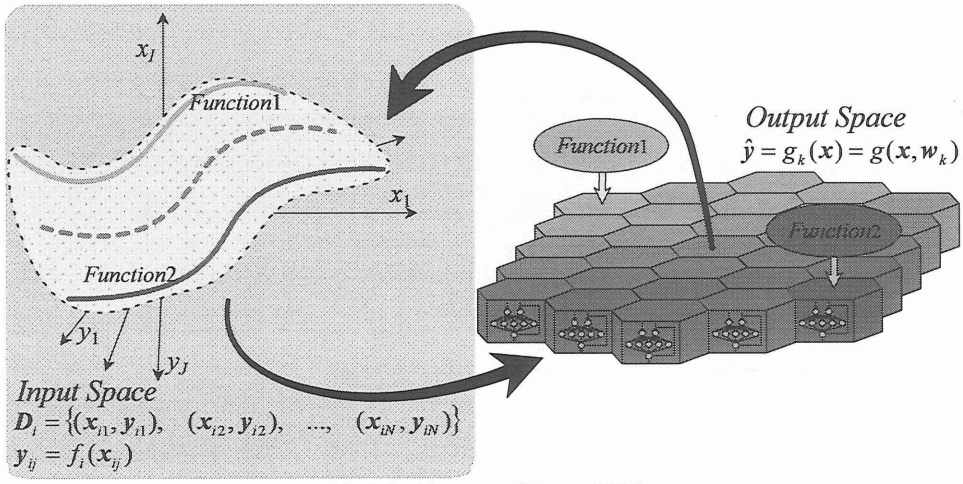


Fig.1 Architecture of the mnSOM

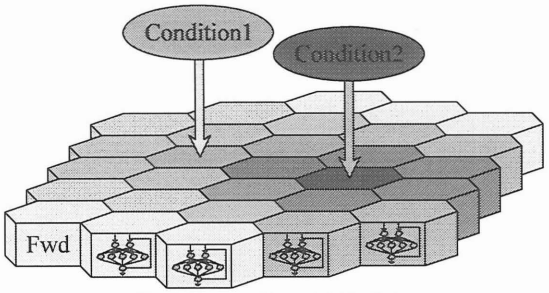


Fig.2 Forward Model Modules

2.2. フォワードモデルモジュール作成

RNN-mnSOM を用いて現在の速度及び制御入力と加速度の関係を得るフォワードモデルモジュール(FMMs: Froward Model Modules)を作成する．以降，状態変数 S は位置，ΔS は速度及び$\Delta^2 S$ は加速度，u は制御入力を示すものとする．図 2 に示すような FMMs 作成のアルゴリズムを以下に述べる．

M 個の時系列データクラス $\boldsymbol{D}_i = (\boldsymbol{D}_1, \boldsymbol{D}_2, \ldots, \boldsymbol{D}_M)$があり，それぞれ N 時間の入出力対$(\boldsymbol{x}_{ij}, y_{ij}) = \{(x_{i1}, y_{i1}), (x_{i2}, y_{i2}), \ldots, (x_{iN}, y_{iN})\}$を持っており，関数 $f_i(.)$を構成しているとすると，式(1)の関係が成り立つ．

$$\begin{aligned} \boldsymbol{D}_i &= (\boldsymbol{x}_{ij}, y_{ij}) = \{(\Delta S_{ij}, u_{ij}), \ \Delta^2 S_{ij}\} \quad (i = 1 \sim M, j = 1 \sim N) \\ \Delta^2 S_{ij} &= f_i(\Delta S_{ij}, u_{ij}) \end{aligned} \tag{1}$$

一方, mnSOM の RNN モジュールが K 個あり，k 番目の FMM の持つ RNN の結合加重は $\boldsymbol{w}_k$ で，入力$(\Delta S_{ij}, u_{ij})$と出力 $\Delta^2 \hat{S}_{ij}(k)$ は，式(2)に示す関数 $F_k(.)$を構成している．

$$\Delta^2 \hat{S}_{ij}(k) = F(\Delta S_{ij}, u_{ij}; \boldsymbol{w}_k) = F_k(\Delta S_{ij}, u_{ij}) \tag{2}$$

このとき，学習は以下のように行われる．

(i) 初期化: FMMs の結合加重 $\boldsymbol{w}_k$ を乱数で初期化する

(ii) 評価過程: 式(3)を用いて，$\Delta^2 \hat{S}_{ij}(k)$ と$\Delta^2 S_{ij}$ の平均二乗誤差 $E_{Fwd_i}(k)$のクラス $\boldsymbol{D}_i$ に対して計算する．式(4)は $f_i(.)$と $F_k(.)$の距離を求めていることになる．

$$E_{Fwd_i}(k) = \frac{1}{2N} \sum_{j=1}^{N} \left\{ \Delta \hat{S}_{ij}(k) - \Delta S_{ij} \right\}^2 \tag{3}$$

(iii) 競合過程: 式(4)に従って，誤差が最小となる BMM(Best Match Module)のインデックスを k_i^*を決定する．

$$k_i^* = \arg\min_k E_{Fwd_i}(k) \tag{4}$$

(iv) 協調過程: 式(5)に従って 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 i 番目のデータクラスの入出力関係を学習する量を示す．$h(.)$は近傍関数で，距離 L 及び学習回数 T が増加するにつれて単調減少する関数を選択する．

$$\Psi_i(k) = h\{L(k, k_i^*), T\} \Big/ \sum_{i'=1}^{M} h\{L(k, k_{i'}^*), T\} \tag{5}$$

(v) 適応過程: 誤差逆伝播(BP: Back Propagation)法によって学習する．式(6)に示すように学習率ηと学習分配率$\Psi_i(k)$を用いて $\boldsymbol{w}_k$ 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 法と同じになる．

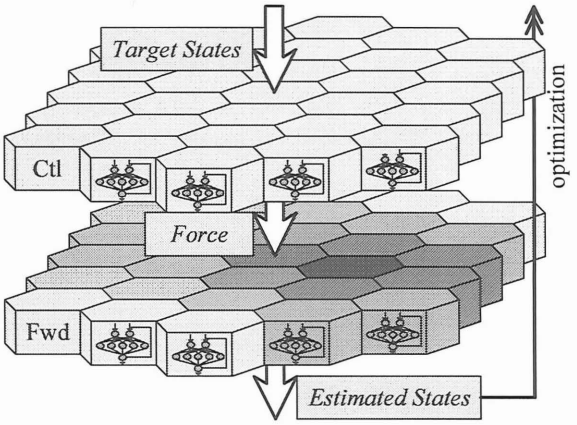


Fig.3 Controller Model Modules

$$\Delta \boldsymbol{w}_k = -\eta \sum_{i=1}^{M} \left\{ \Psi_i(k) \frac{\partial E_i(k)}{\partial \boldsymbol{w}_k} \right\} \tag{6}$$

(ii)~(v)の過程を学習が収束するまで繰り返す．

2.3. コントローラモデルモジュール作成

RNN-mnSOM を用いて，目標値と現在の状態の誤差から制御入力を決定するコントローラモデルモジュール(CMMs: Controller Model Modules)を作成する．図 3 に示すように，CMMs と FMMs を結合し，ひとつの RNN-mnSOM として構成し，FMMs の各ウェイトを固定して，CMMs のウェイトを調整する．CMMs 作成における入力データクラス $\boldsymbol{D}_i$ は目標状態 $\boldsymbol{r} = (r, \Delta r)$を用いて以下の式(8)のように表現する．

$$\begin{aligned} \boldsymbol{D}_i &= (\boldsymbol{x}_{ij}, y_{ij}) = \{(r_{ij} - S_{ij}, \Delta r_{ij} - \Delta S_{ij}, u_{ij-1}), u_{ij}\} \\ u_{ij} &= g_i(r_{ij} - S_{ij}, \Delta r_{ij} - \Delta S_{ij}, u_{ij-1}) \end{aligned} \tag{8}$$

ここで，r は目標位置，Δr は目標速度である．

一方， k 番目の CMM の持つ RNN の結合加重は v_k で，入力$(r_{ij} - S_{ij}, \Delta r_{ij} - \Delta S_{ij}, u_{ij-1})$と出力 u_{ij} は，式(9)に示す関数 $G_k(.)$を構成している．

$$\begin{aligned} \hat{u}_{ij}(k) &= G(r_{ij} - S_{ij}, \Delta r_{ij} - \Delta S_{ij}, u_{ij-1}; v_k) \\ &= G_k(r_{ij} - S_{ij}, \Delta r_{ij} - \Delta S_{ij}, u_{ij-1}) \end{aligned} \tag{9}$$

2.3 節で述べたように，学習は(i)-(v)のプロセスで行われる．CMMs 作成において異なるのは，(ii)評価過程における評価関数 $E_i(k)$と，それに伴って変化する(v)適応過程における結合加重の更新式のみであるが，更新式そのものは式(7)と同じである．式(9)に CMMs に対する評価関数を示す．式(9)は，目標状態に現在の状態が近ければ，また，制御入力の絶対値が少ないほど，値が小さくなり，良い評価であることを示している．

$$E_{Cul_i}(k) = \frac{1}{2N} \sum_{j=1}^{N} \left\{ p_1 (r_{ij} - S_{ij})^2 + p_2 (\Delta r_{ij} - \Delta S_{ij})^2 + p_3 {u_{ij}}^2 \right\} \tag{9}$$

ここで，p_1，p_2，p_3は，それぞれ位置，速度及び制御入力に対する評価値への重み係数である．

2.4. 自己組織的行動決定システム

FMMs と CMMs を用いてどのようにロボットを制御するかを述べる．まず，実際のロボットの状態量を用いて，FMMs から BMM を式(5)より決定する．得られた FMM のインデックスと同じ CMM へ目標状態を入力しその出力をロボットへの制御入力とする．実際に観測される状態を S_{Exp} とすると，上記のプロセスは，式(10)-(12)のように表現できる．

N'時間観測したデータを用いて，式(10)を用いて FMMs との誤差を計算する．

$$E_{Exp}(k)=\frac{1}{2N'}\sum_{j=1}^{N'}\left\{\Delta\hat{S}_{Exp\,j}(k)-\Delta S_{Exp\,j}\right\}^2 \tag{10}$$

次に，式(11)に従って，観測された状態を最も良く表現しているフォワードモデルのインデックス k^*_{Exp} を求める．

$$k^*_{Exp}=\arg\min_k E_{Exp}(k) \tag{11}$$

インデックス k^*_{Exp} のコントローラモデルの出力をロボットへの制御入力として採用する．

$$u_{Exp}=G_{k^*_{Exp}}(r_j-S_{Exp\,j},\Delta r_j-\Delta S_{Exp\,j},u_{Exp\,j-1}) \tag{12}$$

3. シミュレーション

3.1. ダイナミクスマップの作成

提案手法の Phase1 として，FMMs を作成するシミュレーションを行った．まずは，RNN-mnSOM を用いて入力したデータの中間値が補間されたマップが得られるかを検証するために，ひとつの運動モードに関してパラメータを変えてシミュレーションを行った．

式(13)に単純化した水中ロボットの運動方程式を示す．

$$F=M\ddot{x}+C\dot{x}|\dot{x}| \tag{13}$$

Fは力，$\dot{x}$は速度，$\ddot{x}$加速度を示しており，M，Cはそれぞれ質量及び付加質量，非線形流体力に関するパラメータである．式(8)を用いてパラメータ M，C を table1 に示すように変化させ，速度の絶対値が 0.2[m/s]を超えたら制御入力 5[N]の方向を反転させるリミットサイクル運動の時系列データを 50 秒間，10[Hz]でサンプリングしたものを用意し，6x6 格子状モジュール構造の RNN-mnSOM に入力した．

Fig.4-(a)は，Table1 に示している 9 個のパラメータから生成した時系列を mnSOM へ入力し，100,000 回学習を行った結果のマップである．マップの各正方形は FMM を表しており，それぞれダイナミクスを表現した RNN を持っている．入力したデータクラスに対する BMM にはその時系列の速度(細線)，加速度(太線)を，時間を横軸にとってプロットしている．入力したデータクラス D_i ($i=0$〜8)は，それぞれマップ上の(0, 0)，(0, 0)，(2, 6)，(4, 0)，(4, 1)，(4, 6)，(6, 1)，(6, 3)，(6, 6)に配置されている．また，得られた時系列の速度を横軸，加速度を縦軸にとったグラフを Fig.4-(b)に示している．

Fig.4-(b)の各リミットサイクルのグラフの位置は，Fig4-(a)の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いる．制御入力が一定であることから，M（質量及び付加質量）が増加すると加速度の最大値が減少することから，左側から右側に向かって M が増加していることが分かる．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．C が上昇すると加速度の減衰が大きくなることから，右から左に向かって C の値が上昇していることが分かる．

3.2. コントローラマップの作成

提案手法の Phase2 として，CMMs 作成シミュレーションを行った．作成した FMMs と乱数でウェイトを初期化した 6x6 格子状の CMMs を結合し，FMMs の各 RNN のウェイトを固定して，CNM のウェイトを調整する．目標位置を 0〜25[sec]で 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 10[Hz]で与えた．学習を 15,000 回行った結果得られた CMM を Fig.5 に示す．

各正方形は CMM しており，Fig4-(a)の同位置に配置された FMM に対応したコントローラを実現しているニューラルネットワークを持っている．各正方形の中には，時間[sec]を横軸に，黒太線は目標位置[m]，灰色線はロボットの位置[m]，黒細線は制御入力[10N]をプロットしている．各 CNM は全て目標値を追従しており，与えられた FMMs の表現するダイナミクスに対応したコントローラを生成できていることが確認できる．

4. 結論

mnSOM を用いた適応制御手法を提案し，Phase1 として，ロボットのダイナミクスを表現するマップ生成，Phase2 として，作成したダイナミクスマップを用いてコントローラマップの作成を行った．ダイナミクスマップにおいて，入力した時系列の特徴が表現されており，また，入力時系列の中間値を補間したダイナミクスも同時に得ることができた．コントローラマップは対応したダイナミクスをもったモデルに対して目標軌道を追従する制御ができた．

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.I186-I191

[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] 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

[5] T. Kohonen, (1982), “Self-organized formation of topologically correct feature maps”, Biological Cybernetics, vol.43, pp.59-69

[6] 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

[7] 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

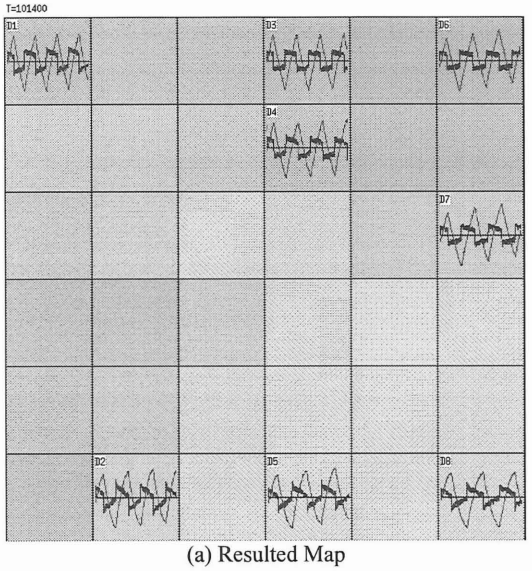
[8] K. Tokunaga, T. Furukawa, (2005), “Nonlinear ASSOM Constituted of Auto associative Neural Modules”, 5th Workshop on Self-Organizing Maps

[9] 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 Nonliner

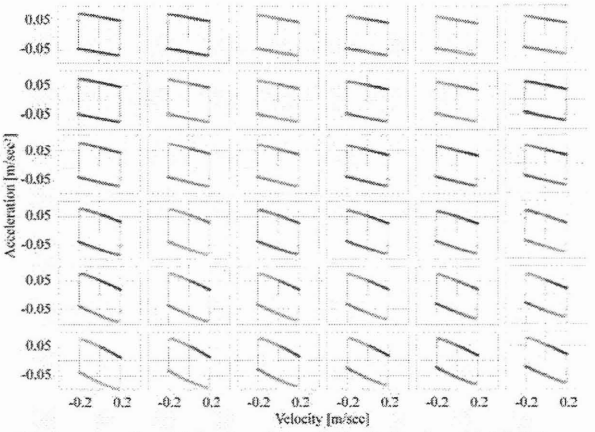
Theory and its Applications, pp.231-234

Table 1 Parameters of FMMs Simulation

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



(a) Resulted Map



(b) Acceleration-Velocity plots of Limit Cycle with FMMs

Fig.4 Simulation Resulted Map and Estimated Parameters of Equation

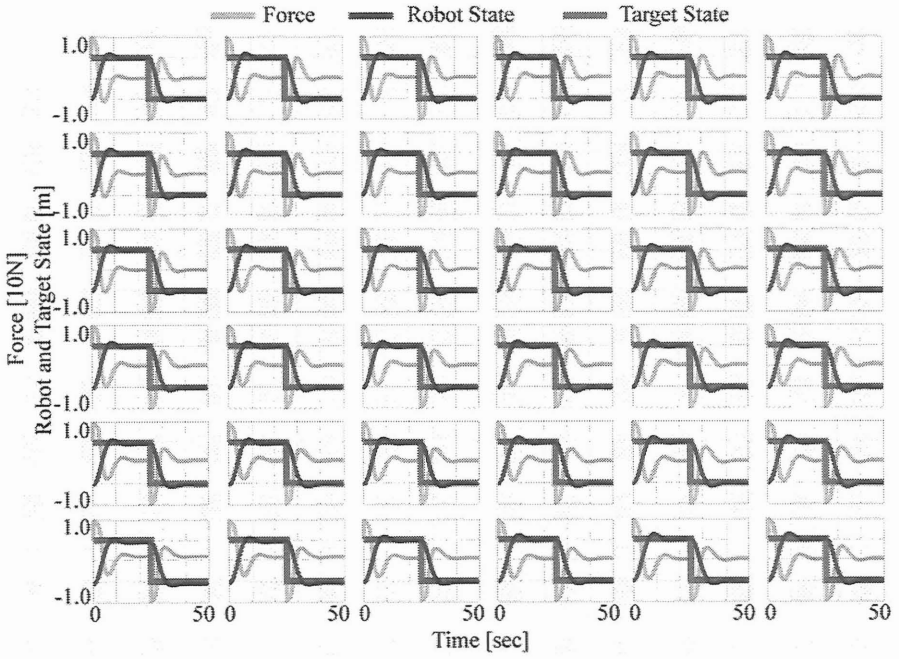


Fig.5 Simulation Resulted Map of CNMs