

ニューラルネットワークを用いた水中ロボット マニピュレータのシミュレーション

平 雄一郎*・森元映治*・横田源弘*・中村 誠*

Numerical Simulation of Underwater Manipulators with Neural Networks

Yuichiro Taira*, Eiji Morimoto*, Motohiro Yokota*, and Makoto Nakamura*

This report deals with a numerical simulation method for underwater manipulators. A simulation study is cost-effective in comparison with an experimental one, especially in such an expensive system as underwater robot that requires a huge tank, waterproof robots, and so on. Therefore, several simulation models for underwater manipulators have been reported. The models contain some of the hydrodynamic effects acted on the manipulator such as drag and lift forces, buoyancy forces, and added mass forces. It, however, is well known that the complete hydrodynamic model in the form of an equation may be unavailable. That is, the accurate model for underwater manipulators cannot be obtained. Hence, the simulation results may be drastically different from the experimental ones. In this report, a simulation method with neural networks has been developed. Neural networks can be used as a modeling method that reproduces and generalizes experimental data by training. That is, the model in this report is based not on equations representing laws of nature but on a set of data collected during a practical experiment.

1 緒 論

従来、海洋におけるロボット技術は軍事および海底石油開発の需要に伴って発達してきた。しかし現在、この需要は減少の傾向にあるが、その代わりに魚介類行動生態調査などの水産業分野、海底汚染源除去作業などの環境保全分野、防波堤建設などの港湾土木分野への適用が期待されている¹⁾。このような海洋ロボット技術において、人間にとって危険な海中環境で自律的に作業を遂行する自律型海中ロボットの開発は特に重要である。さらに、この海中ロボットは人間の腕に相当する作業用マニピュレータを搭載しておくことが望ましい。そこで本報では、海中を含む水中ロボットマニピュレータを対象として、その手先位置の自動制御法の開発・解析に必要なモデル化およびシミュレーション法を取り扱う。

一般に、制御法の開発においては、実機を用いた実験によりその性能を解析・評価するが、水中ロボットマニピュレータのように、充実した実験環境を得ることが容易では

ない対象ではこのことが支障となる。なお、水中ロボットマニピュレータの制御実験としては、例えば文献2)-4)が報告されているが、安全性を確保できるくらい大きな水槽や防水したロボットなど、費用のかかる実験装置を用いている。このような場合、計算機による数値シミュレーション解析が有効である。これに対して、水中ロボットマニピュレータのモデルやシミュレーション法がいくつか報告されている⁵⁾⁻¹⁰⁾。水中ロボットマニピュレータの挙動は、流体抗力、浮力、付加質量など、周囲の流体による影響を受けるので、通常のマニピュレータのダイナミクスだけでなく、これらの影響も考慮しなければならない。しかしながら、流体による影響は様々な種類があり複雑であるので^{11), 12)}、数式として正確なモデルを得るのは困難である。すると、数式モデルによる方法⁵⁾⁻¹⁰⁾では、得られたシミュレーション結果と本来の実験結果が著しく異なる可能性がある。

一方、生物の脳・神経系の構造を模した人工ニューラルネットワークを用いたモデル化手法が様々な問題に適用さ

れ、その有用性が報告されている¹⁸⁾。ニューラルネットは、複雑な非線形性を有する入出力関係であっても、数多くの実データを学習することにより、それを極力再現するような結合強度に変化させる。すなわち、ニューラルネットモデルは運動方程式などの自然法則による数式モデルではなく、実データを再現・汎化するモデルであり、水中ロボットマニピュレータのような正確な数式モデルを得ることが困難である対象に有用である。

本報では、ニューラルネットワークを用いた水中ロボットマニピュレータの数値シミュレーション法を検討する。なお、ニューラルネットの学習には、水中ロボットの実データが必要である。ここでは、まず第1段階として、実機を用いた実験により得られたデータではなく、従来モデルを用いたシミュレーションで得られたデータを実データとして利用し、従来モデルを再現可能であるかを確認する。

2 方 法

2.1 運動方程式を用いた従来のマニピュレータモデル

水中ロボットマニピュレータの数式モデルとしては、Articulated-Body法⁸⁾やKane法⁹⁾を用いて導出されたものが提案されているが、ここでは、Lagrange法とNewton-Euler法による標準的なモデル^{5),7)}を考える。すると、 n リンク回転関節水中ロボットマニピュレータの運動方程式は一般に次式で表現される。

$$\{M(q) + M_A(q)\} \ddot{q}(t) + b(q, \dot{q}) + b_A(q, \dot{q}) + f_D(t) + f_G(t) = \tau(t) \quad (1)$$

ただし、 $M = R^{n \times n}$ は慣性行列、 $M_A = R^{n \times n}$ は付加質量行列、 $b = R^n$ は遠心力・コリオリ力ベクトル、 $b_A = R^n$ は付加質量による遠心力・コリオリ力ベクトル、 $f_D = R^n$ は流体抗力と浮力から構成される外力ベクトル、 $f_G = R^n$ は重力から構成される外力ベクトル、 $\tau = R^n$ はマニピュレータの関節トルクベクトル、 $q = R^n$ はマニピュレータの関節角度ベクトルである。ここで、 $M_A = 0$ 、 $b_A = 0$ および $f_D = 0$ とすると、通常の陸上ロボットマニピュレータの運動方程式¹¹⁾となる。すなわち、 $M_A \ddot{q}$ 、 b_A と f_D が流体による影響を表現している項である。付加質量による力 $M_A \ddot{q}$ と b_A は、マニピュレータが流体中を運動する際に周囲の流体の一部もマニピュレータと共に動かされるために生じる力である^{11),12)}。一方、流体抗力はマニピュレータの動作に対して流体から受ける反力である^{11),12)}。一般には、流体抗力は低

Reynolds数の場合は速度に比例し、高Reynolds数の場合は速度の2乗に比例することが知られている¹²⁾。なお、提案されている水中ロボットマニピュレータモデルのほとんどは、速度の2乗に比例する流体抗力モデルを用いている^{5),10)}。また、浮力はよく知られているように物体の上下面の圧力差により生じ、物体が排除した流体の質量に比例する流体力である¹¹⁾。その他にも、式(1)で表現されていない、マニピュレータの動作により生じる水流や渦の影響など、様々な流体による影響が考えられる。

2.2 ニューラルネットワークを用いたマニピュレータモデル

一般に、ニューラルネットワークは階層型とリカレント型に大別される¹³⁾。前者は、信号の流れが入力層から出力層の順に一方方向であり、ダイナミクスを考慮する必要がないフィードフォワード構造である。したがって、フィードバック構造を有しないため、パターン識別のような静的な写像問題に適用される。一方、後者はある層の出力が同層の入力となることが可能であり、フィードバック構造を有する。したがって、差分または微分方程式で記述されるような動的な写像問題にも適用できる。

さて、運動方程式(1)からも分かるように、水中ロボットマニピュレータの挙動は微分方程式で記述される動的な写像問題であるので、階層型ではなくリカレント型を使用する。ある層の出力が同層の入力になるということは、離散時間系の観点からすると、ある時点の出力が次時点の入力になると考えることができる。すなわち、各層のユニットを、各時点のマニピュレータの状態(関節角度・角速度)とみなして、時間空間を各層に空間展開した階層型ニューラルネットと考えることができる。このような考えに基づく方法として、EBPTT (Epochwise Back Propagation Through Time) 法¹³⁾が提案されており、これをマニピュレータモデルに適用する。

本報で提案する n リンク回転関節水中ロボットマニピュレータのニューラルネットモデルをFig. 1に示す。ユニットの種類を明確にするため、出力ユニットを丸で、隠れユニットを二重丸で、入力ユニットを四角で記述している。本文中では、出力ユニットの集合を O 、隠れユニットの集合を H 、入力ユニットの集合を I として区別する。また、 $t = t_0, t_0 + 1, \dots, t_{N-1}, t_N$ は各時点の時刻、 N はデータのサンプル数、 $d_i(t)$ は時刻 t における出力ユニット i の実データ(教師データ)、 w_{jk} はユニット k からユニット j への結合強度である($i = O = j = H = O = k = I = H = O$)。さらに、

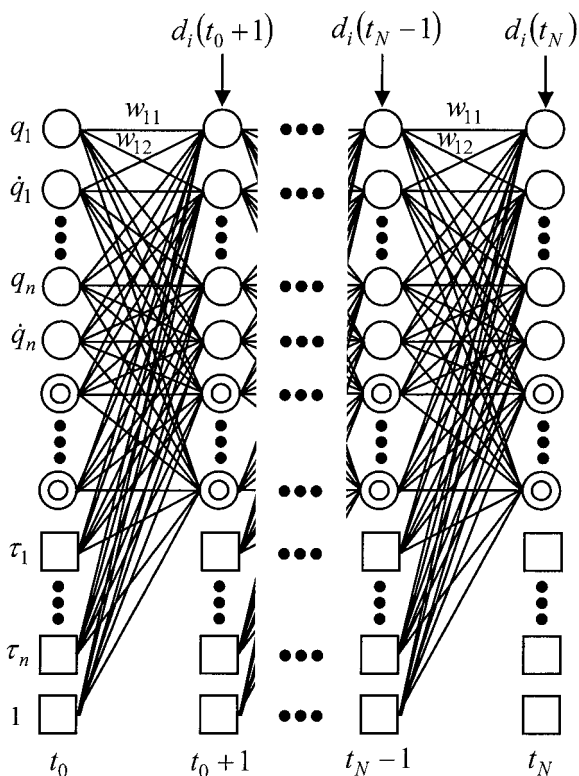


Fig. 1. Neural network model for n-link underwater manipulators.

$q_i(t)$ と $\dot{q}_i(t)$ はそれぞれ時刻 t における第 i 関節の角度と角速度, $\tau_i(t)$ は時刻 t における第 i 関節のトルク入力である ($i=1, 2, \dots, n$)。なお, 閾値の学習を行うため, 入力ユニットの中に常に 1 を出力する仮想的なユニットが含まれている¹³⁾。

まず第 1 層において, 出力ユニットに初期時点の関節角度 $q_i(t_0)$ と角速度 $\dot{q}_i(t_0)$, 隠れユニットすべてに初期値 0, 入力ユニットに初期時点の各関節駆動トルク $\tau_i(t_0)$ が入力されて, つぎに第 2 層以降では, 結合強度 w_{jk} を係数として階層的に各層の状態 $q_i(t)$ と $\dot{q}_i(t)$ が決定される

$$(i=1, 2, \dots, n \Rightarrow H=O \Rightarrow k=I=H=O \Rightarrow t=t_0+1, \dots, t_N)。$$

これを式で表現すると

$$y_i(t) = \sum_{j=I} w_{ij} x_j(t-1) + \sum_{j=H=O} w_{ij} y_j(t-1) \quad (i=H=O \Rightarrow t=t_0+1, \dots, t_N) \quad (2a)$$

$$y_i(t_0) = \begin{cases} q_i(t_0) \text{ or } \dot{q}_i(t_0) & \text{if } i=O \\ 0 & \text{if } i=H \end{cases} \quad (j=1, 2, \dots, n) \quad (2b)$$

$$x_i(t) = \tau_i(t) \text{ or } 1 \quad (i=I; j=1, 2, \dots, n; t=t_0, \dots, t_N) \quad (2c)$$

となる。ただし, y_i と x_j はそれぞれモデルの出力と入力

である ($i=H=O \Rightarrow j=I$)。さらに, 結合強度は実データである教師データにモデル出力が一致するように次式の学習則で調整する¹³⁾。

$$w_{ij}(k) = \begin{cases} w_{ij}(k-1) - \eta \sum_{t=t_0}^{t_N-1} \delta_i(t+1) x_j(t) & \text{if } j=I \\ w_{ij}(k-1) - \eta \sum_{t=t_0}^{t_N-1} \delta_i(t+1) y_j(t) & \text{if } j=H=O \\ & (i=H=O) \end{cases} \quad (3a)$$

$$\delta_i(k) = \begin{cases} e_i(t) & \text{if } t=t_N \\ e_i(t) + \sum_{j=H=O} w_{ji} \delta_j(t+1) & \text{if } t=t_0+1, \dots, t_N-1 \\ & (i=H=O) \end{cases} \quad (3b)$$

ただし,

$$e_i(t) = \begin{cases} y_i(t) - d_i(t) & \text{if } i=O \\ 0 & \text{if } i=H \end{cases} \quad (t=t_0+1, \dots, t_N)$$

であり, k は学習時点, η は学習率である。

3 結果と考察

本報におけるニューラルネットモデルの有用性を検証するため, 2リンク垂直型水中ロボットマニピュレータを対象とした数値シミュレーションを行った。ニューラルネットモデルでは, 学習(教師)データとして, ロボットの実データが必要である。本報では, まず第 1 段階として, 従来モデル(1)によるシミュレーションで得られたデータを学習データとして利用する。すなわち, ここでは, ニューラルネットモデルが従来モデルの挙動を再現可能であることを確認する。また, シミュレーションはつぎの 3 通りについて行った。

- 1) 第 1 関節のみに一定トルクを加える
($\tau_1 = 2.0[\text{N}], \tau_2 = 0.0[\text{N}]$)。
- 2) 第 2 関節のみに一定トルクを加える
($\tau_1 = 0.0[\text{N}], \tau_2 = 1.0[\text{N}]$)。
- 3) 第 1 関節と第 2 関節の両方に一定トルクを加える
($\tau_1 = 3.0[\text{N}], \tau_2 = 1.0[\text{N}]$)。

シミュレーションで対象としたマニピュレータの物理パラメータを Table 1 に示す。なお, このパラメータは文献 4) で使用されたロボットの実機を参考にして設定した。学習データは, 従来モデル(1)を Runge-Kutta 法¹⁵⁾ で数値計算して得た。ただし, 学習に使用するデータの周期は 0.05 [s] とした。また, ニューラルネットモデルの中間ユニッ

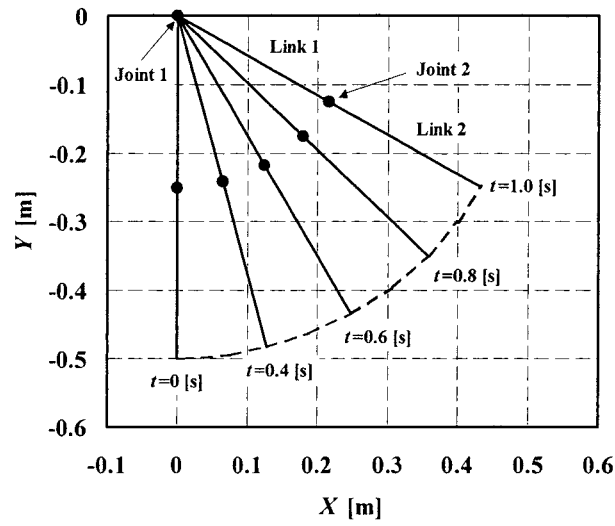
トの個数は10に設定した。さらに、学習則(3)に関して、その初期値 $w_{ij}(0)$ は小さな乱数に設定し¹⁸⁾、学習率 η はケース1)と2)では0.0001, ケース3)では0.00005とした。

ケース1)から3)の結果をそれぞれFig. 2から4に示す。なお、各図の(a)は従来モデルの挙動, (b)はニューラルネットモデルの挙動, (c)はニューラルネットの学習

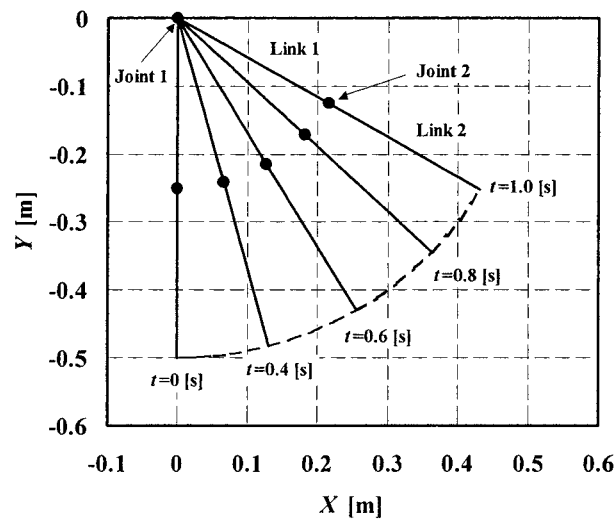
誤差である。各図(a)と(b)より、ニューラルネットモデルは従来モデルの挙動を十分に再現できていることが分かる。さらに、各図(c)より、初期時点において学習誤差は急速に減少し、最終的に零に収束しており、学習に成功していることが分かる。以上より、本報におけるニューラルネットモデルの有用性を確認した。

Table 1. Physical parameters of 2-link underwater manipulator.

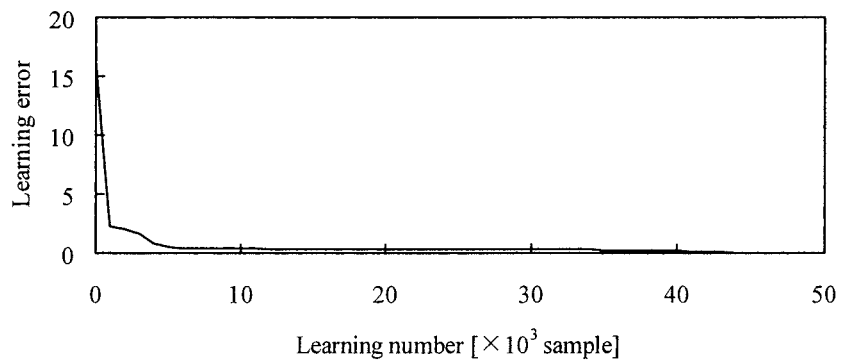
	Link 1	Link 2
Mass [kg]	4.25	1.23
Volume [m ³]	2.43×10^{-3}	0.83×10^{-3}
Moment of inertia [kg m ²]	0.19	0.012
Link length (x axis) [m]	0.25	0.25
Link length (y axis) [m]	0.04	0.04
Link width [m]	0.12	0.12
Added mass (x) [kg]	1.31	0.1
Added mass (y) [kg]	3.57	2.83
Added moment of inertia [kg m ²]	0.11	0.06
Drag coefficient (x)	0	0
Drag coefficient (y)	1.2	1.2



(a) Conventional model

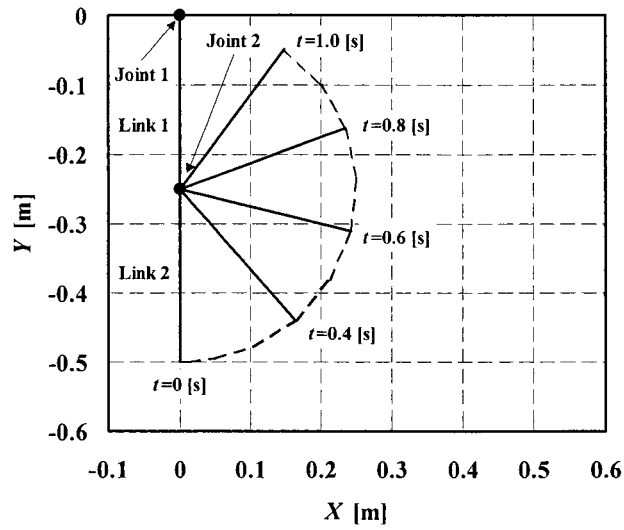


(b) Neural network model

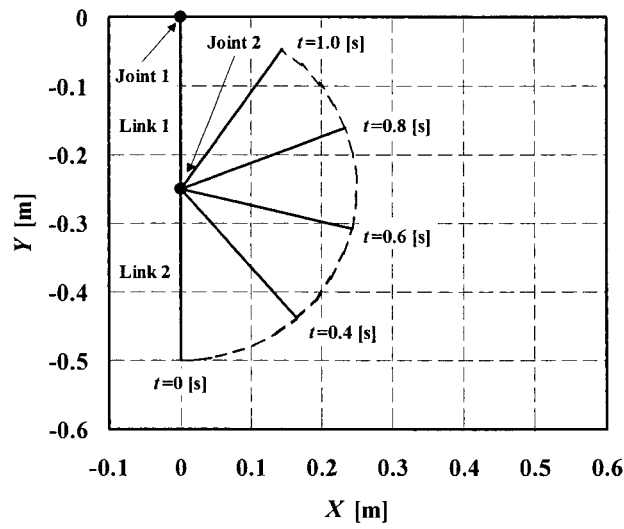


(c) Learning error

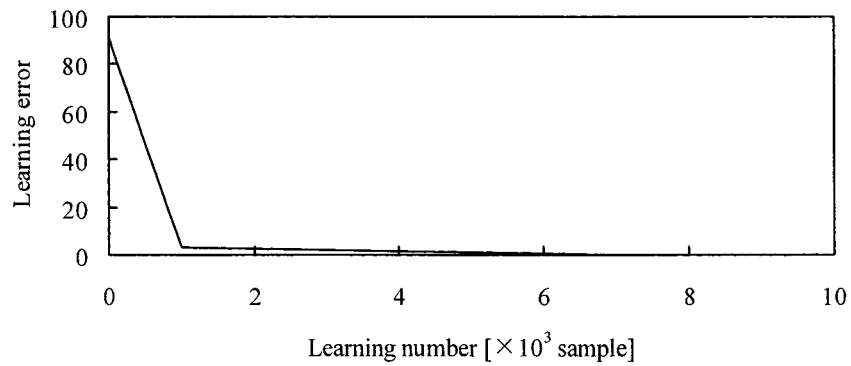
Fig. 2. Simulation results: case 1.



(a) Conventional model

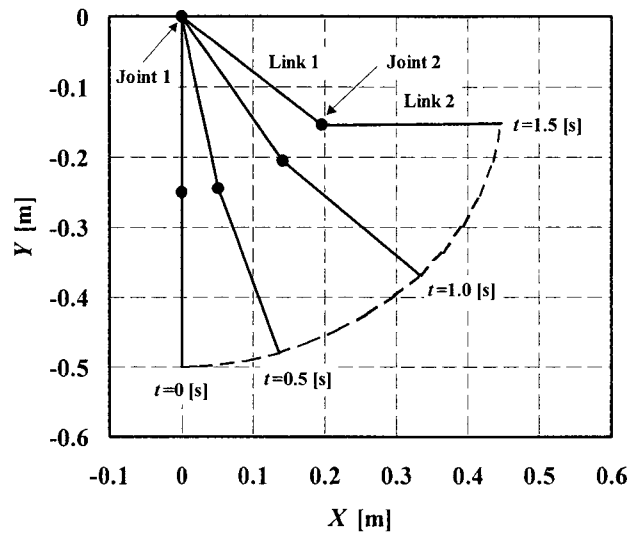


(b) Neural network model

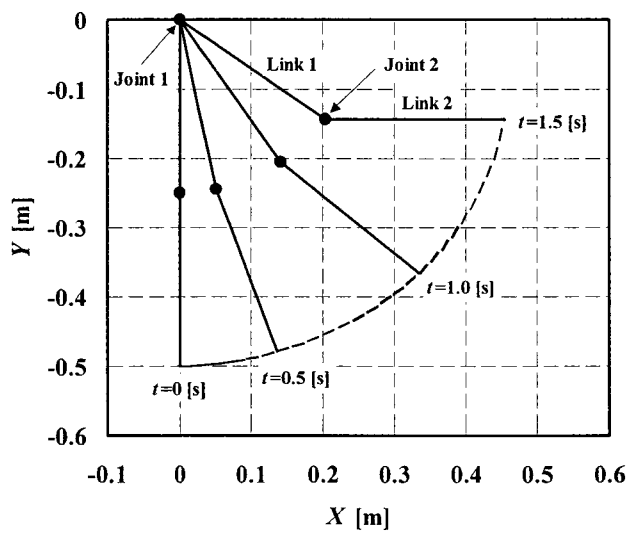


(c) Learning error

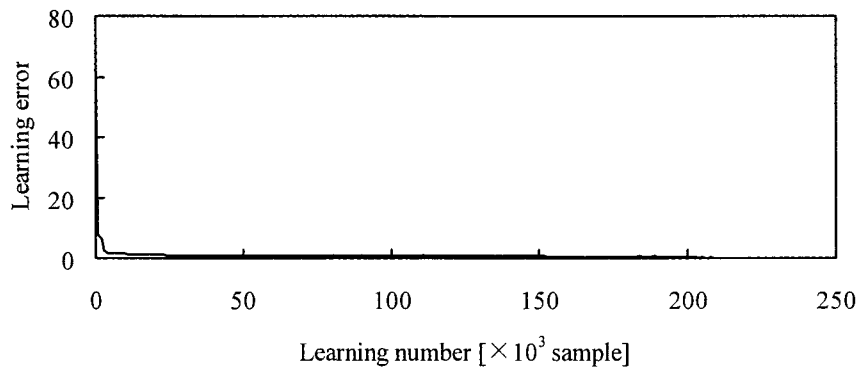
Fig. 3. Simulation results: case 2.



(a) Conventional model



(b) Neural network model



(c) Learning error

Fig. 4. Simulation results: case 3.

4 結 論

本報では、水中ロボットマニピュレータに対するニューラルネットワークを用いたモデル化および数値シミュレーション法を考案するとともに、2リンク垂直型水中ロボットマニピュレータを対象としたシミュレーションによりその有用性を確認した。シミュレーションでは、従来モデルから得られたデータを学習データとして使用し、ニューラルネットワークモデルが従来モデルの挙動を学習・再現可能であることを確認した。なお、シミュレーションは2リンクマニピュレータに対して行ったが、本モデルは一般の n リンクマニピュレータに対して適用可能である。

つぎの段階として、水中ロボットマニピュレータの実機による実験データを入手し、それを本モデルが学習・再現可能であるかを検証する予定である。

最後に、本シミュレーションのプログラム作成およびデータ取得にご協力をいただいた本校学生の正田美奈子君（現九州工業大学大学院）ならびに吉田隼人君（現阪神低温株式会社）に感謝の意を表します。

文 献

- 1) 浦環・高川真一：海中ロボット，成山堂（1997）。
- 2) T. W. McLain, S. M. Rock, and M. J. Lee: Experiments in the Coordinated Control of an Underwater Arm/Vehicle System, *Autonomous Robots*, **3**, pp. 213-232 (1996).
- 3) S. Sagara, T. Tanikawa, M. Tamura, and R. Katoh: Experiments on a Floating Underwater Robot with a Two-Link Manipulator, *Artificial Life and Robotics*, **5**(4), pp. 215-219 (2001).
- 4) S. Sagara, K. Shibuya, and M. Tamura: Experiment of Digital RAC for an Underwater Robot with Vertical Planar 2-Link Manipulator, *Proc. of the Ninth Int. Symp. on Artificial Life and Robotics*, pp.337-340 (2004).
- 5) 五百井清・伊藤憲治：水中マニピュレータのモデリングとシミュレーション，*日本ロボット学会誌*，**6**(2)，pp. 160-168(1988)。
- 6) 足達康行・吉田和夫：水中浮遊ロボットの作業領域と動的可操作性，*日本ロボット学会誌*，**12**(6)，pp. 863-871(1994)。
- 7) B. Levesque and M. J. Richard: Dynamic Analysis of a Manipulator in a Fluid Environment, *Int. J. of Robotics Research*, **13**(3)，pp. 221-231 (1994)。
- 8) S. McMillan, D. E. Orin, and R. B. McGhee: Efficient Dynamic Simulation of an Underwater Vehicle with a Robotic Manipulator, *IEEE Trans. on System, Man, and Cybernetics*, **25**(8)，pp. 1194-1206 (1995)。
- 9) T. J. Tarn, G. A. Shoults, and S. P. Yang: A Dynamic Model of an Underwater Vehicle with a Robotic Manipulator Using Kane's Method, *Autonomous Robots*, **3**, pp. 269-283 (1996)。
- 10) T. W. McLain and S. M. Rock: Development and Experimental Validation of an Underwater Manipulator Hydrodynamic Model, *Int. J. of Robotics Research*, **17**(7)，pp. 748-759 (1998)。
- 11) J. N. Newman: *Marine Hydrodynamics*, MIT Press (1977)。
- 12) 今井功：流体力学前編，裳華房（1973）。
- 13) S. Haykin : *Neural Networks: A Comprehensive Foundation* (2nd Edition), Prentice Hall (1999)。
- 14) T. Yoshikawa : *Foundations of Robotics: Analysis and Control*, MIT Press (1990)。
- 15) 戸川隼人：数値計算法，コロナ社（1981）。