

ノードゲート付きニューラルネットワークとその非線形システム制御への応用

村田, 純一

九州大学大学院システム情報科学研究科電気電子システム工学専攻

柿原, 知英

九州大学大学院システム情報科学研究科電気電子システム工学専攻 : 修士課程 | 本田技研工業(株)

藤本, 雅貴

九州大学大学院システム情報科学研究科電気電子システム工学専攻 : 修士課程

平澤, 宏太郎

九州大学大学院システム情報科学研究科電気電子システム工学専攻

<https://doi.org/10.15017/1515707>

出版情報 : 九州大学大学院システム情報科学紀要. 5 (2), pp.243-248, 2000-09-26. 九州大学大学院システム情報科学研究所

バージョン :

権利関係 :

ノードゲート付きニューラルネットワークと その非線形システム制御への応用

村田 純一*・柿原 知英**・藤本 雅貴***・平澤 宏太郎*

Neural Network with Node Gates and its Application to Nonlinear System Control

Junichi MURATA, Tomohide KAKIHARA, Masaki FUJIMOTO and Kotaro HIRASAWA

(Received June 16, 2000)

Abstract: Neural networks with node gates are proposed to solve complicated or large problems with 'divide and conquer' approach. Each hidden node of the network has a node gate on its output channel which controls the flow of the output from the node. By opening and closing depending on situations, the node gates form a sub-network dynamically which gives the solution suited for the current situation. When the situation changes, the gate openings are also changed accordingly, and a different sub-network will emerge to give a new solution. In the paper, a mechanism that controls the gate opening is proposed as well as the learning method of the network weights and the parameters contained in the gates. The network is applied to nonlinear system control problem where a number of different situations occur and demand for different control strategies. The results show that the proposed network can deal with the change of the situations appropriately.

Keywords: Neural networks, Node gates, Nonlinear system control, Divide and conquer scheme, Function localization

1. はじめに

階層型ニューラルネットワークは、有界連続関数を任意の精度で近似できるという特徴を持っているため、さまざまな問題に応用されている。しかし、対象とする問題が複雑あるいは大規模になると、ネットワーク規模を極めて大きくしない限り、十分な性能を発揮することができない。複雑あるいは大規模な問題に対しては、これを小問題に分け、各小問題に対する個々の解を合成することにより全体の解を得る分割統治 (divide and conquer) の考え方がある。これは、見方を変えると、小問題ごとに解を算出する機構が異なるという機能局在の考え方と同じである。本論文は、階層型ニューラルネットワークにこれらの考え方を導入したノードゲート付きニューラルネットワークを提案し、複数の異なる状況において異なる解が必要となる非線形システム制御問題に適用することにより、その有効性を検証する。

ノードゲート付きニューラルネットワークは、階層型ニューラルネットワークの各中間層ノードの出力端に、ノードから出力される信号の伝達を制御するゲートを設けたものである。開ゲートを持つノードがサブネット

ワークを構成し、これによって一つの解が実現される。さらにこの解は状況に応じた各ゲートの開閉によって切替えられる。

類似の試みは、ファジィ・ニューラルネットワーク¹⁾や、Hierarchical Mixtures of Experts²⁾などでなされているが、ゲート付きニューラルネットワークは、これらと異なり、ノード毎にその使用/不使用を決定する。このため、各ノードは一つの小問題だけでなく複数の小問題で使われてもよい。したがって、各小問題の解を構成するサブネットワークは小問題ごとに完全に分離されている必要はなく、サブネットワーク内の一部のノードが複数の小問題に重複して使用されることもあり得る。このような構造上の重複があることによって、未経験の小問題についても一定の性能を得ることができ、また、小問題間の滑らかな切替えも可能となる。また、ノード毎に使用/不使用の切替えを行うネットワークとして、著者等の研究室ではLearning Petri Networks³⁾を提案しているが、ノードゲート付きネットワークでは、解の実現機構と解の切替え機構を分離することにより、より自由度の高い分割統治あるいは機能局在を実現することを目指している。

以下では、ゲートの開閉の実現機構と、ニューラルネットワークの結合荷重とゲート開閉を制御するパラメータとの学習による決定法を提案し、これをクレーン

平成12年6月16日受付

* 電気電子システム工学部門

** 電気電子システム工学専攻修士課程 (現在 本田技研工業(株))

*** 電気電子システム工学専攻修士課程

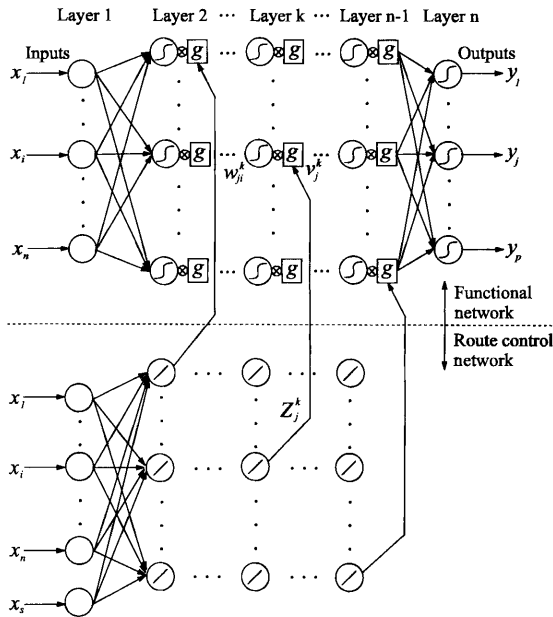


Fig.1 Structure of neural network with node gates.

の制御問題に適用して有効性を検証する。

2. ノードゲート付きニューラルネットワーク

2.1 ネットワーク構造

ノードゲート付きニューラルネットワークの構造を Fig.1 に示す。図の上部が解を算出するための機能用ネットワークであり、各中間層ノードにゲートが配置されており、中間層ノードの出力はゲートの開度 $g(z_j^{(k)})$ ($0 \leq g(z_j^{(k)}) \leq 1$) に応じて次式のように伝達される。

$$v_i^{(1)} = x_i \tag{1}$$

$$v_j^{(k)} = f\left(\sum_{i=1}^{N_{k-1}} w_{ji}^{(k)} v_i^{(k-1)}\right) g(z_j^{(k)}) \tag{2}$$

$$v_j^{(n)} = y_j = f\left(\sum_{i=1}^{N_{n-1}} w_{ji}^{(n)} v_i^{(n-1)}\right) \tag{3}$$

ここで、 f は入出力関数、 N_k は第 k 層のノード数、 w_{ji}^k は第 $k-1$ 層の第 i ノードと第 k 層第 j ノード間の結合荷重である。

図の下部は、ゲートの開度を決定し、したがって機能用ネットワーク内の信号経路を決定する情報を出力する経路用ネットワークである。この経路用ネットワークへの入力によって各ゲートの開閉が決定されることになる。また、この入力の値が変化すれば、それに応じてゲートの開閉したがってノードの使用/不使用が動的に変化する。

2.2 ゲート構造

ゲートの開度を表す関数 $g(z)$ には、シグモイド関数

$$g(z) = \frac{1}{1 + \exp\{-a(z-b)\}} \tag{4}$$

を用いる。 $g(z)$ は a, b (以下ではゲートパラメータと呼ぶ) と z の値に応じて、0 (ゲートが閉)、1 (全開)あるいはその中間の値(半開)をとることができる。また、ゲートパラメータを変化させることにより、 z に応じた開閉の様子を変化させることができる。

2.3 経路用ネットワーク

ゲートの開閉を決定する変数 z の値は、Fig.1 下部の経路用ネットワークによって決定される。このネットワークの第 k 層第 j ノードの出力値 z_j^k によって、機能用ネットワークの第 k 層第 j ノードのゲートの開閉が制御される。経路用ネットワークへの入力は、機能用ネットワークへの入力と同一であっても異なってもよい。小問題の解を実現する機能用ネットワークと、解の切替を行う経路用ネットワークを分離することにより、自由度の高い分割統治が実現できる。

Fig.1 では経路用ネットワークのノード内の関数を線形関数として描いているが、これは線形関数に限らない。経路用ネットワークのノード内の関数や結合荷重は、適当な事前情報が得られる場合はそれに基づいて決定する。事前情報が得られない場合は乱数などを用いて設定する。いずれにしても、ゲートの開閉状況はゲートパラメータによって変更できるため、経路用ネットワーク内の係数等は学習による変更の対象とはしない。

2.4 ネットワークの学習

ノードゲート付きニューラルネットワークにおいては、結合荷重とゲートパラメータという意味や役割の異なる2種類のパラメータを学習する必要がある。各ノード出力は結合荷重の値に影響を受け、ゲート開度はゲートパラメータに影響される。これらが各ノードの出力端で乗ぜられるため、両パラメータは互いに強い依存関係を持ってネットワーク出力に影響を与える。そのため、単に両パラメータをネットワーク誤差が最小となるように学習しても、希望通りの学習結果を得ることは困難であり、何らかの方法で学習を誘導することが必要である。著者らは、様々な小問題について豊富に学習用データが得られる場合には、ネットワーク誤差に適切な付加項を加えたものを最小化することによって、両パラメータの同時学習が可能であることを示した⁴⁾。しかし、非線形システムの制御のようなダイナミクスを含む適用対象においては、多くの小問題についてのデータを用いるのは計算時間の点で適切ではない。そこで、ここでは、一部に

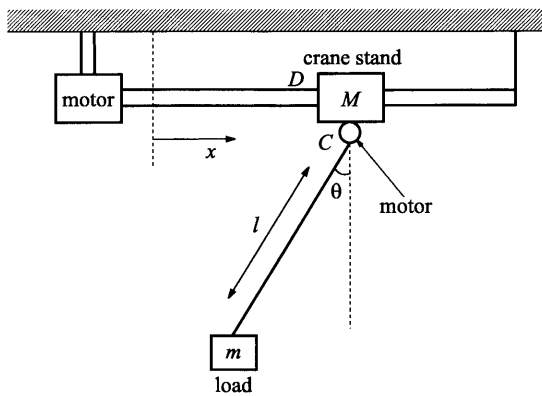


Fig.2 Crane system.

人間による誘導を交えて、両パラメータについて順に学習を行う方法を用いる。

まず、ゲートパラメータの値を、適当な事前知識に基づくかあるいは適当な乱数を用いて設定する。ゲートパラメータをこの値に固定した上で、適切な評価関数 E を最小化するように、結合荷重をBack-Propagation (BP) 法を用いて学習する。典型的な E としては、ネットワーク出力の誤差

$$E = \frac{1}{2} \sum_{k=1}^p (y_k - d_k)^2 \quad (5)$$

が考えられる。ここで、 y_1, \dots, y_p はネットワーク出力、 d_k は各出力に対応した教師信号である。ただし、結合荷重の更新はゲートの開度に応じた量となる。特に、ゲート開度が0のノードについては学習が行われない。

結合荷重の学習が設定回数行われると、次にゲートパラメータの学習を行う。この時、結合荷重の値は上で得られた学習結果の値に固定しておく。この学習もBP法を用いて評価関数 E を最小にするように行うが、 E の値の計算は結合荷重の学習に用いたデータとは異なるデータを用いることにより、汎化能力の向上を図ることも可能である。

以降必要に応じて、結合荷重とゲートパラメータの学習を、適当な判断規準によって停止判断が出されるまで繰り返す。

3. 非線形システム制御への適用

制御対象の非線形システムとして、Fig.2 に示すクレーンシステムを用いる。クレーンシステムにおいては、荷が軽い時と重い時では、制御方法を変えた方がより正確かつ効率良く荷を運べると考えられる。そこで、荷の重さに応じた小問題への分割を行い、荷の重さという状況が変化しても良い制御が行える制御器を、ノードゲ

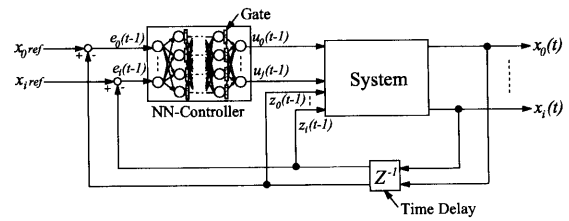


Fig.3 Control system.

ト付きニューラルネットワークを用いて構成する。

3.1 制御系

本来、クレーンシステムは連続時間系であるが、ニューラルネットワークを用いた制御器による制御を行うため離散時間系に変換する。クレーンシステムの時刻 t における状態変数 \mathbf{x} 及び入力電圧 \mathbf{u} は以下の式で表される。

$$\mathbf{x}(t) \equiv [x(t), \dot{x}(t), \theta(t), \dot{\theta}(t), l(t), \dot{l}(t)]^T \quad (6)$$

$$= \mathbf{f}(\mathbf{x}(t-1), \mathbf{u}(t)) \quad (7)$$

$$\mathbf{u}(t) \equiv [u_d(t), u_m(t)]^T \quad (8)$$

ここで、 \mathbf{f} はクレーンの運動方程式、 $x(t)$ は荷を吊している台車の移動距離、 $\theta(t)$ は荷を吊すロープの揺れ角、 $l(t)$ はロープ長、 $u_d(t)$ は横移動用モータへの入力電圧、 $u_m(t)$ は縦移動用モータへの入力電圧である。

制御系を Fig.3 に示す。ノードゲート付きニューラルネットワーク制御器から出力された値がクレーンを駆動するモータへの入力電圧となり、クレーンからのフィードバック値と目標値との偏差が制御器の機能用ネットワークに入力される。経路用ネットワークには、荷の重さのみが入力される。

3.2 学習

学習の際に用いる評価関数として以下の2次形式の評価関数を用いる。

$$E = \sum_{t=1}^{t_f} \{ \mathbf{e}(t)^T \mathbf{Q} \mathbf{e}(t) + \mathbf{u}(t)^T \mathbf{R} \mathbf{u}(t) \} \quad (9)$$

$$= \sum_{t=1}^{t_f} \left(\sum_{i=1}^6 q_i e_i(t)^2 + \sum_{i=1}^2 r_i u_i(t)^2 \right) \quad (10)$$

ここで、 t_f は最終時刻、 $\mathbf{e}(t)$ は時刻 t におけるクレーンの状態変数 \mathbf{x} と目標値との差、 q_i, r_i は重み係数である。

ネットワーク内のパラメータを更新すると、ネットワークの出力値が直接影響を受けると同時に、その値を受け取ったクレーンからのフィードバック値も影響を受ける。このため、制御器として用いる場合の学習はBack-Propagation Through Time法を用いて行う。学習の第 $k+1$ 回目の繰り返しにおける結合荷重 w の更新式

は以下の式で表される.

$$w_{k+1} = w_k - \alpha \left. \frac{\partial^{\dagger} E}{\partial w} \right|_{w_k} + \beta \Delta w_k \quad (11)$$

$$\Delta w_k = w_k - w_{k-1} \quad (12)$$

$\partial^{\dagger} E / \partial w$ は, 他のパラメータを固定した場合の w の変化による直接・間接の影響をすべて考慮した E の変化を表し, α, β は学習係数である. ゲートパラメータについても同様の式にしたがって値の更新を行う.

4. シミュレーション

4.1 条件

ノードゲート付きニューラルネットワークによるクレーン制御のシミュレーションを行う. 制御の目的は, クレーンの台車位置 (x) を 0[m] から 1[m] まで移動させ, 荷を $l = 1$ [m] から $l = 0.5$ [m] まで巻き上げることである.

用いる機能用ネットワークは, それぞれ10個のノードからなる二つ中間層を持つ, 6入力2出力の階層型ネットワークである. 入力は6個の状態変数と各目標値との偏差であり, 出力はクレーンへの入力電圧 u_d, u_m である. 中間層ノード内の関数として, $-1 \sim 1$ の出力が得られる \tanh 関数を用い, 出力層ノードは線形ノードとする. 実際にはこのネットワークを二つに分割し, 一方を u_d (横移動) 用ネットワーク, もう一方を u_m (縦移動) 用ネットワークとして用いる. 横移動用モータは, 全ての偏差が0になれば停止する必要があるため, u_d 用ネットワークの各中間層ノードには定数項であるしきい値を含めない. 一方, 巻き上げ用モータは, 全ての偏差が0になっても重力と釣合うだけのトルクを発生する必要があるため, u_m 用ネットワークの各中間層ノードにはしきい値を入れている. また, このしきい値を荷の重さに応じて変化させることができるよう, u_m 用ネットワークの第2中間層ノードにはゲート付きのしきい値を2個ずつ設ける. この2個のしきい値は定数1を出力する第1中間層ノードであると解釈して, そのゲートの開閉を決定する.

経路用ネットワークの中間層は, 機能用ネットワークと同じ構成であるが, 入力層ノードは1個とし, 入力として荷の重さ m を与える. また, 経路用ネットワーク内のパラメータは, 想定する荷の重さ m の範囲 $0 \sim 10$ [kg] を事前情報として, 経路用ネットワークの第1中間層からは, 半数のノードから $z = m - 5$ が, 残りの半数から $z = -(m - 5)$ が, 第2中間層からは, 半数から $z = (m - 5) - 2.5 \cdot \text{sgn}(m - 5)$ が, 残り半数からはこれと逆符号の値が, それぞれ出力されるように設定した. ここで, $\text{sgn}(\cdot)$ は符号関数である.

機能用ネットワークの結合荷重の初期値は乱数により与え, ゲートパラメータの初期値は全て $a=1.0, b=0.0$ とした. これにより, 機能用ネットワークの各中間層の

Table 1 Initial states of node gates.

m [kg]	1A	1B	2A	2B
0.0 - 2.5	Open	Closed	Open	Closed
2.5 - 5.0	Open	Closed	Closed	Open
5.0 - 7.5	Closed	Open	Open	Closed
7.5 - 10.0	Closed	Open	Closed	Open

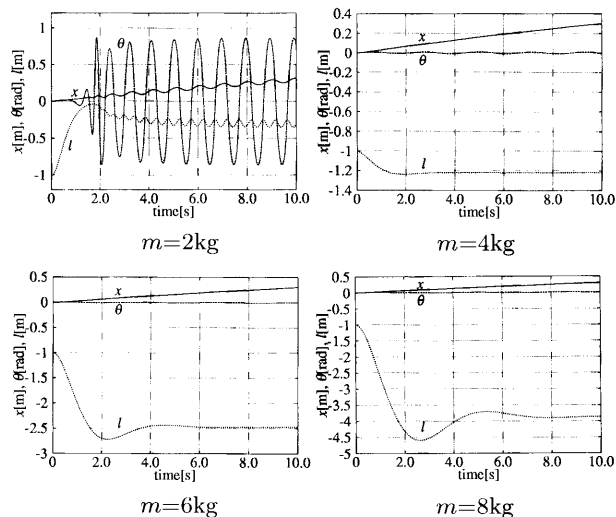


Fig.4 Responses of crane controlled by ordinary neural network.

ノードはゲートの開閉の様子により, それぞれ二つのグループに分けられる. 第1中間層のノードのグループを 1A, 1B, 第2中間層のノードグループを 2A, 2B と名付けると, 学習前の各々のノードグループのゲート開閉は Table 1 に示す通りとなる. このように, 各グループとも, 四つある荷の重さ m の区間のうち, 2区間でゲート開となり, 重複して使われることになる.

比較のために, ノードゲートを用いない通常のニューラルネットワークを制御器として使用することも検討した. ここで, ネットワークへの入力, 公平を期すために, 6個の偏差と荷の重さ m の7個とした.

4.2 結果

4.2.1 ノードゲートの効果

まず, ノードゲートの効果を確認するために, ゲートパラメータを前節で述べた初期値に固定し, 結合荷重だけを学習した場合について, 通常のニューラルネットワークとの性能比較を行う. 学習は, 荷の重さ 2, 4, 6, 8[kg] のいずれに対しても良い制御が行える制御器を構成することを目的とした. これら複数の荷の重さ m 全てに対応するためには, 特定の m に偏らない学習とする必要がある. そこで, 学習の繰り返しを1回行うごとに m の値を変更することとした.

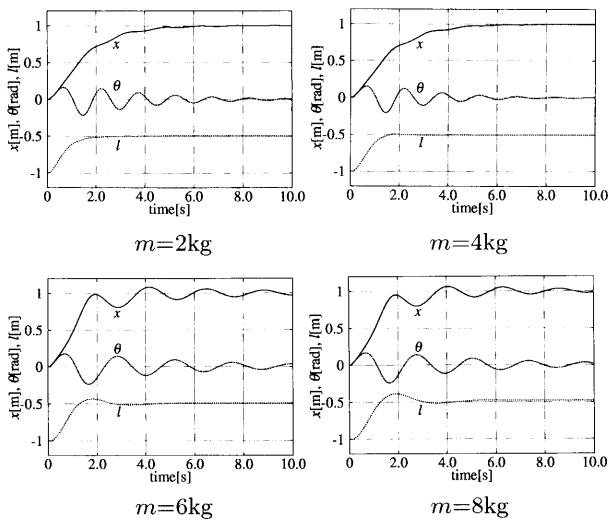


Fig. 5 Responses of crane controlled by neural network with node gates.

このような複数の m に対して良好な制御を可能とするには、制御器の自由度(可調節パラメータ数)が高い方が望ましい。そこで、通常のネットワークについては第1中間層のノードを、ノードゲート付きニューラルネットワークの2倍の20個として比較を行った。

Fig. 4に、ノードゲートを持たない通常のニューラルネットワークによる台車位置 x 、ロープ振れ角 θ 、ロープ長 l の応答を示す。これから、どの状況 (m の値) についても中途半端にしか学習が行われず、全ての状況において適切な制御が行われていないことがわかる。一方、**Fig. 5**に示したノードゲート付きニューラルネットワークによる制御器の制御応答は、ほぼ全ての状況において、与えられた時間内に目標値に到達できていることを示している。

以上より、ノードゲート付きニューラルネットワークによる制御器は、ノードゲートの機能によって、複数の状況に対応した制御器となり得ることが示された。

4.2.2 ゲートパラメータ学習の効果

上述の結果は、ゲートパラメータを初期値に固定したまま結合荷重のみを学習して得たものである。そこで次にゲートパラメータの学習を行う。結合荷重の学習により、各ノードがどのような機能を持つかが既に決定されているので、ゲートパラメータの学習が果たすべき役割は、この各ノードをいつ(どのような m の値に対して)用いるかを決定することである。これが適切に行われることにより、結合荷重の学習に用いた値 ($m = 2, 4, 6, 8$) 以外の m の値に対しても良好な制御が行えること、すなわち汎化能力の向上が期待される。

ゲートパラメータは、結合荷重の学習では用いなかった $m = 3, 5, 7$ を使い、制御性能の評価関数式(10)を最小化するように、BP法を用いて学習する。学習の1回の

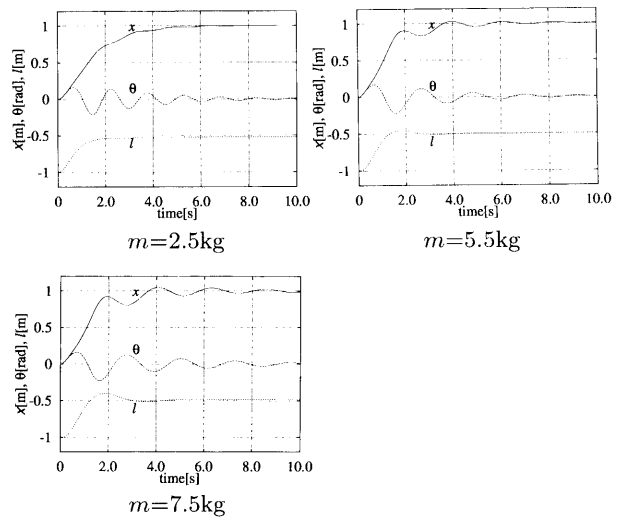


Fig. 6 Responses of crane controlled by neural network with node gates after gate parameter learning.

Table 2 Performance indices before/after gate parameter learning.

m [kg]	Before learning	After learning
2.5	5184.9	4997.2
3.5	8458.2	8007.1
4.5	11994.2	11971.2
5.5	17143.8	16799.3
6.5	23034.8	22941.2
7.5	30139.9	29985.3

繰り返しごとに m の値を変更する。また、結合荷重は先に学習で得られた値に固定しておく。

Fig. 6 にゲートパラメータの学習を行った後の、 $m = 2.5, 5.5, 7.5$ に対する制御応答を示す。これらの m の値は、結合荷重の学習の際にもゲートパラメータの学習の際にも用いられていない値である。これより、未学習状況においても有効な制御が行われていることがわかる。なお、**Table 2** に、学習に用いていないいくつかの m の値に対して、ゲートパラメータの学習の前後での制御性能の評価関数式(10)の値を示す。これより、ゲートパラメータの学習によって、学習を行っていない状況での評価関数値も改善され、制御性能が向上していることがわかる。

以上より、ノードゲート付きニューラルネットワークによる制御器はゲートパラメータを学習させることにより、未学習状況にも対応できる制御器となることが示された。

4.2.3 構造上の重複の効果

Table 1 に示したように、中間層ノードの各グループは、 m の4区間の値のうち2区間でゲートが開き、使用される。このようにノードが使用される状況は単一では

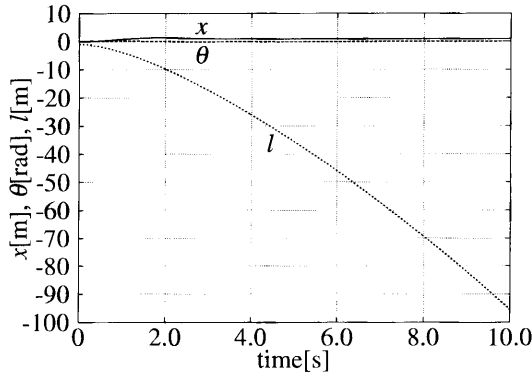


Fig.7 Responses of crane with 6[kg] load controlled by neural network with node gates trained with 2, 4 and 8[kg] loads.

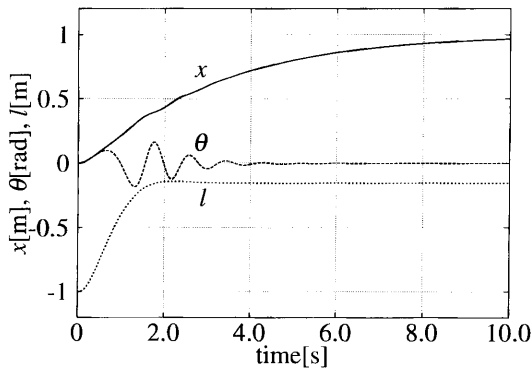


Fig.8 Responses of crane with 4[kg] load controlled by neural network with node gates trained with 2, 6 and 8[kg] loads.

なく、複数の状況で重複して使用される。そこで、一つの状況のみを対象として学習を行っても、重複している他の状況にも対応可能であると期待できる。例えば、上述の結果では、結合荷重の学習は $m = 2, 4, 6, 8$ に対して行っており、Table 1に示した m の4区間がすべて考慮されている。しかし、例えば $m = 4$ を学習の際に考慮しなくても、 $m = 4$ で使用されるノードのグループ1A, 2Bは、それぞれ $m = 2, m = 8$ においても使用されるので、それらのノードの機能は $m = 2, 8$ を対象とした学習によって獲得されると期待できる。

そこで、結合荷重の学習において、

1. $m = 2, 4, 8$ を対象として学習する ($m = 6$ は対象としない).
2. $m = 2, 6, 8$ を対象として学習する ($m = 4$ は対象としない).

の2通りを試みた。なお、ゲートパラメータの学習はいずれの場合も前節と同様に行った。

Fig.7 に $m = 6$ を除いて学習とした場合の $m = 6$ に対する制御応答を示す。この場合、荷が落下してしまい、全く制御できていないことがわかる。これは、ネットワークの持つ自由度が高く、3種類の m の値を指定しただけでは、学習によって期待通りのパラメータ値が獲得できるとは限らないためであると考えられる。一方、Fig.8に示した $m = 4$ を省いて学習した場合の $m = 4$ に対する制御応答は、 l に偏差があるものの、Fig.7に比べると制御が機能していることを示している。

以上より、ノードが複数の状況で重複して使用されるという特徴を活かすことにより、学習の対象とする状況が少ない場合でも、多くの状況に対応した制御器とすることができる可能性があることが示された。

5. おわりに

本論文では、経路用ネットワークを備えたノードゲート付きニューラルネットワークを提案した。このノードゲート付きニューラルネットワークは、ゲートの開閉により信号伝達経路を動的に変化させることができ、状況の変化に柔軟に対応することができる。これを非線形システム制御器へ適用することにより、このネットワークの有効性を示した。

通常のニューラルネットワークによる制御器では、複数の状況の全てを良好に制御することが困難であるのに対し、ノードゲート付きニューラルネットワークによる制御器では、複数の状況に対応できることが確かめられた。また、ノードが複数の状況で重複して使用されるという特徴によって、学習対象の状況が不足している場合でも、未学習の状況にある程度対応できることの可能性が示された。

参考文献

- 1) H.Takagi and I.Hayashi: NN-driven fuzzy reasoning, *Int. J. Approx. Reasoning*, **5**, 191-212, 1991.
- 2) M.J.Jordan: Hierarchical Mixtures of Experts and the EM algorithm, *Proc. of 1993 Int. Joint Conf. Neural Networks*, 1339-1344, 1993
- 3) K.Hirasawa, M.Ohbayashi, S.Sakai and J.Hu: Learning Petri Network and Its Application to Nonlinear System Control, *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, **28**, 781-789, 1998
- 4) J.Murata, T.Nakazono, K.Hirasawa: Neural Networks with Node Gates, *Proc. 1999 Int. Conf. Elec. Engineering*, **1**, 358-361, 1999
- 5) 柿原: ノードゲート付きニューラルネットワークとその非線形システム制御への応用, 九州大学大学院システム情報科学研究科電気電子システム工学専攻修士論文, 2000