多次元情報抽象化レベルを有する

キュービックニューラルネットワークによる知的制御[†]

木 通 秀 樹*・吉 田 和 夫**・神 谷 勝*

Intelligent Control Using Cubic Neural Network with Multi-Levels of Information Abstraction

Hideki KIDOHSI*, Kazuo YOSHIDA** and Masaru KAMIYA*

Neural network control has been applied to various objects, and showed its validity. From the viewpoint of intelligent control, however, it is considered that most of utilizations of neural network are no more than a technique in which a part of information processing is replaced with that using a neural network and they are not able to cope with abnormal situation. This study introduces a new concept of hierarchical information abstraction. This study presents a new technique of intelligent control using the neural network, called "Cubic Neural Network" (CNN) which possesses multi-levels of information abstraction, and furthermore presents a systematic synthesis method of CNN. Therefore, qualitative knowledge or information can be used in addition to ordinary feedback control using only quantitative measured values. And a robust control which has large applicable regions is realized. And, as an example, the control of inverted pendulum including supposed or not supposed large parameter change is dealt with, and the usefulness of the present method is shown.

Key Words : neural network, intelligent control, parallel processing, multi-levels processing, hierarchical abstraction, inverted pendulum

1. はじめに

ニューラルネットワーク¹⁾は、その学習機能、非線形画像機能等の特徴を生かして、様々な分野で応用研究が行われてきた。制御の分野においてもオペレータのノウハウを学習したり、制御対象の逆モデルを学習することによって、従来の手法では困難であった様々な対象を制御することが可能になってきた。

しかし、知的制御の観点から見ると、これらは情報処理のニューラルネットワークによる部分的な置き換えであり、異常事態などを含めた様々な場合に対応することができないものが多い。これでは十分に知的であるとはいえない。

多くの場合、知的な制御を実現するには、通常の計測値をフィードバックする定量的な制御だけでなく、定性的な情報や知識などを用いた制御を、同時に行うことが有効である。しかし、そのような知的な制御を実現するためには、これらの定量的な情報と定性的な情報をそれぞれ適切に処理し、適切な関係を持って組み合わせなければならない。また、定性

的情報には複数のレベルが存在すると考えられるので、さらに関係のある情報を何らかの基準に基づいて整理することによって、定性的情報を適切に処理するようになさなければならない。

この実現方法として、これまで多くの概念と手法が提案されてきた^{2)~6)}。代表的な概念である Rasmussen のモデル²⁾は、人間の認知的挙動を、スキルベース-ルールベース-知識ベースという3段階に分け、これらを効果的に統合したものである。また、Saridis の IPDI (the principle of Increasing Precision with Decreasing Intelligence) は³⁾、階層的構造によって、より具体的な戦略と大まかな戦略を実現するものである。しかし、これまでの実現方法は、各層の制御戦略や、階層構造の構成方法などが問題個別に議論され、思考錯誤的に設計されるものであった。

そこで、本研究においては、知的制御における情報処理法として並列処理を基本とし、その統一的基準として情報の抽象化レベルに着目した方法を提案する。このようにすると、前述の複数の定性的レベルの情報と定量的情報は、抽象度という統一的な尺度によって階層的に整理することができる。この概念図を Fig.1 に示す。

Fig.1では、抽象度の高い情報を取り扱う制御を上位レベルに、抽象度の低い情報を取り扱う制御を下位レベルに示した。抽象度の高い情報を取り扱う上位レベルでは、定性的な制御を行うようになり、情報の精度低下や制御対象の特性変動に

[†] 日本機械学会第72期通常総会講演会で発表 (1995・3)

* 慶応義塾大学大学院 横浜市港北区日吉 3-14-1

** 慶応義塾大学理工学部 横浜市港北区日吉 3-14-1

*** Faculty of Science and Technology, Keio University, Kohoku-ku, Yokohama

(Received November 24, 1995)

(Revised September 11, 1996)

Fig. 1 Each abstraction level's controller ability

対応することが可能になるなど、下位レベルに比べて定量的誤差は増大するものの、適用範囲の広いロバストなコントローラとすることが期待できる。逆に、抽象度の低い情報を取り扱う下位レベルでは定量的な制御を行い、精度の良い情報が得られれば、上位レベルに比べて高い制御性能が得られることが予想される。

さらに、この知的制御のシンセシスの方法として、多次元の情報処理を行う立体的なニューラルネットワークの応用を提案する。この方法を用いることで、前述の複数の抽象化レベルを有する、階層的な情報処理の適切な組み合わせを実現する。このニューラルネットワークを本研究ではCNN (Cubic Neural Network) と呼ぶ。

本論文では、CNNの基本概念とその構造、さらに、このCNNを用いた知的制御手法の概念と、具体的な設計手法を述べる。そして、応用例として大きなパラメータ変動がある倒立振子の制御について考察し、さらに、設計時には考慮していないパラメータ変動が発生した場合についても考察する^{7) 8)}。

2. CNNの概念

CNNの基本概念の構造をFig.2に示す。本CNNはFig.2のように階層型のニューラルネットワークを、層状に積み重ねる構造を持つ立体型のニューラルネットワークによって実現され

Fig. 2 A basic structure of CNN

る。本論文では、このCNNの構造を制御器として用いる。この際、各レベルのニューラルネットワークは、取り扱う情報の抽象度を統一した、複数のニューラルネットワークによって構成される。

上位レベルのニューラルネットワークほどより抽象度の高い定性的な情報処理を行い、逆に、下位のレベルに行くほど定量的な情報処理を行う。これらのニューラルネットワークによる情報処理は最終的には統合化されて出力を行う。そこで、各レベルには、現在の状況においてそのレベルの情報処理が適切かどうかを自律的に評価するニューラルネットワークを併せ持つようにする。

すなわち、これらのニューラルネットワークは並列に計算を行い、得られた各レベルの出力のうち、どのレベルの出力が最も適切かを評価しながら、自律的にレベルを切り替えて最終的な出力を行う。

本研究では、このような各レベルのニューラルネットワークを、学習によって獲得することで実現する。更に、自律的切り替えを行う統合化部については、遺伝的アルゴリズム(GA)を用いた学習によって獲得する。

3. CNNによる知的制御の設計概要

CNNを設計するためには、前章で述べたように、定量的なニューラルネットワークと、複数のレベルで抽象化された定性的なニューラルネットワーク、さらに、各レベルのニューラルネットワークを評価するニューラルネットワークをそれぞれ設計しなければならない。

本章ではCNNを知的制御手法として用いるための設計方法の基本的な考え方として、その抽象化方法と各レベルのニューロコントローラ、評価するニューラルネットワークの設計概念について述べる。

3.1 抽象度の設定と抽象化の方法

CNNでは、各レベルでどのような情報を取り扱うかを決定することが最も重要な問題である。

情報表現の基本構造である階層的抽象化方法として、ここでは情報処理の性質として上位が下位を包含することを目的とし、Fig. 3のように上位のレベルの情報の属性が下位のレベルの情報の属性を包含するように抽象化を行う。これは、フレーム型の情報表現における上位レベルと下位レベルの関係と同様の構造となっている。また、具体的な情報の抽象化の方法として、ここではファジィニューラルネットワークによる抽象

Fig. 3 Numerical hierarchical abstraction of information

化法を用いる。

計測情報は主に数値であるが、この数値情報の階層的抽象化の例をFig.3に示す。各レベルでは、Fig.3で示すような各レベルの抽象度の情報のみを用いて処理を行うようにする。すると、CNNの情報処理は上位レベルであるほど抽象度の高い処理を行う。また、この抽象化は各レベルのニューラルネットワークの内部で行う。ここでは、抽象化(定性化)の方法として、計測情報を実数軸上の境界で分ける量子化を行い¹⁰⁾、各領域内では計測情報は一定とし、その領域分けの度合いによって抽象度を決定する。例えば、Fig.3のレベル3では、計測値が正の値であればPositiveという領域に統一されるとする。

ここで適切な領域分けをする手段として、本研究ではニューラルネットワークの第一中間層において、しきい値処理による領域分けを行うファジィニューラルネットワーク¹¹⁾を用い、結合の重みを学習すると同時に、領域の境界も学習によって獲得するようにした。Fig.4にその構成例を示す。本ニューラルネットワークに入力された信号は、(1)式に示すような切りたったシグモイド関数によってしきい値処理を行う第一中間層ニューロンを用いて、情報量の低減を行うような抽象化を行う。ここで、Fig.4では第一中間層ニューロンを3つ持ち、それぞれで異なるしきい値を持つことによって、一つの信号を大きく4つの領域に分割する。この変換によりこの場合は、Positive Big, Positive Small, Negative Big, Negative Smallのように4つの領域分割される。これらの領域分割によってファジィモデルにおけるメンバシップ関数を表現する。また、第2、第3中間層では(2)式で示すようなシグモイド関数を持つ中間層ニューロンによってファジィルールを実現する。この際、本ネットワークの出力が同じレベル内の他のネットワークに再度入力する場合には、Fig.4の出力層に示すように第一中間層と同じしきい値特性を持つ切りたったニューロンによって再度抽象化を行う。

$$f(x) = \frac{2}{1 + \exp(-x)} - 1 \quad (1)$$

$$f(x) = \frac{2}{1 + \exp(-100x)} - 1 \quad (2)$$

3.2 抽象度の異なるコントローラの設計方法

3.2.1 定量的なコントローラの設計

レベル1では通常の計測値が用いられ、抽象度が最も低い定量的な制御を行う。この場合は、通常のニューロコントローラの設計を行う^{12) 13)}。

Fig. 4 The method of abstraction by fuzzy neural network

3.2.2 定性的なコントローラの設計

上位に行けば行くほど、抽象度を上げるために領域分けの個数を減少させる。この場合は、前節で示した第一中間層で抽象化を行うファジィニューラルネットワークを用いて、レベル1で設計したニューロコントローラの入出力特性を学習する。

3.3 各レベルのコントローラの評価方法および評価ネットワークの学習方法

定性的コントローラと定量的コントローラを統合するためには、各レベルの評価が必要になる。本研究では、この評価部をFig.5のようにシステムニューラルネットワークと評価ネットワークによって構成する。

システムニューラルネットワークは、コントローラと同様にファジィニューラルネットワークを用いて、各レベルの抽象度に応じた情報により、制御対象の入出力特性を学習する。そして、学習したシステムニューラルネットワークに、1サンプル過去の計測値を入力したときの出力値 $x_t^A[k]$ と、現在の計測値を抽象化した値 $x_t^A[k]$ との誤差を計算する。そして、この誤差を評価ネットワークへの入力とする。

評価ネットワークの出力は、そのレベルのコントローラが現在の状況に適しているかどうかを評価する評価指数 p であり、(3)式のように表される。

$$p = \frac{1}{1 + \exp \left[\sum_{i=1}^n \{ -\theta_i (x_t^A[k] - \hat{x}_t^A[k]) \} + \theta_0 \right]} \quad (3)$$

ここで、 $x_t^A[k]$ 、 $\hat{x}_t^A[k]$ 、 θ_i は m 次のベクトルであり、 t はベクトルの転置を表わす。また、 θ_i はネットワークの重みであり、 m は領域分けの個数を示す。

また、本ニューラルネットワークの学習では、コントローラネットワークに対しては最適制御によって設計されたコントローラを教師として与えることができ、また、システムニューラルネットワークに対しては制御対象の入出力特性を教師として与えることが可能であるが、評価ネットワークに対しては、適切な教師情報を作成することができないため、教師を必要としない学習則であるGA¹⁴⁾を用いた。

まず、重み、しきい値を染色体上にコード化し、多数の個体を用意する。ただし、各染色体上の遺伝子は-1.0~+1.0の範囲でランダムに決定する。次に、あるレベルを選択し、そのレベルのコントローラと上位レベルのコントローラを適当に切り替えてシミュレーションを行う。ここで、二つのレベルのみでシ

Fig. 5 Structure of each level's evaluation part

ミュレーションするのは、どのレベルでも、適用範囲は上位のコントローラが広く、性能は下位のレベルのコントローラが高いと想定しているからである。そして、選択交配、交差と突然変異を繰り返して行なうことによって世代交代を重ね、適応度が高くなるように評価ネットワークの重みを更新する。

これにより、各レベルのコントローラが現在の状態に適しているかを評価することができるようになる。この評価は各レベルで独立に行われる。

3.4 各レベルのコントローラの統合化方法

このようにして、各レベルのコントローラとそのシステムニューラルネットワークおよび評価ネットワークを実現する。そして、全ての評価ネットワークを統合することによって、CNNは最も適した出力を行うようになる。その統合化された構造をFig.6に示す。

前節の学習により、各レベルではコントローラの出力を抑制すべきかどうかを判断できるようになる。しかし、下位のレベルほど詳細な情報を持っているので、下位のコントローラが制御できるときには、それを用いるのが適切な制御方策と考えられる。そこで、各レベルでは自分が出力して良いときは、自らの出力を優先し、上位のコントローラの出力を抑制する。そして、自分が出力してはいけないときに、自らの出力を抑制し、上位のコントローラ出力を優先するという構造を用いた。

この判断は(3)式のしきい値処理によって行う。出力すべきでないときは p が1となり、出力すべきときは0となる。この関係を(4)式に示す。

$$u_i[i] = (1-p[i]) \cdot u_c[i] + p[i] \cdot u_i[i+1] \quad (4)$$

ここで、 i レベルのコントローラの出力を $u_c[i]$ 、 $(i+1)$ レベル(上位)以上のレベルのコントローラまでで統合化された最終的なコントローラの出力を $u_i[i+1]$ とし、統合後の最終的な i レベル

Fig. 6 An example of 3 levels structure of CNN

の出力を $u_i[i]$ とする。この計算は評価部を持たない最上位レベルから順に下位レベルへと進める。このようにすると、レベル1の評価後の出力がCNNの最終的な出力となる。

これにより上位のレベルが下位のレベルを包含する構造を実現できる。この切り替えは前節で述べたシグモイド関数によって滑らかにかつ補間的に行われる。

4. 倒立振り子制御への応用

本章では、提案するCNNを用いた制御が、異常な事態にも対応可能な知的制御手法であることを示す。

ここでは、非線形特性を有し、大きなパラメータの変動が発生する制御対象として、センサに故障を起こす倒立振り子を選び、CNNによる安定化制御を行った。

4.1 倒立振り子モデル

倒立振り子をFig.7に示す。倒立振り子は、タイミングベルトとボールねじを介して、DCサーボモータによって駆動される。ここでは制御入力であるDCサーボモータへの入力電圧値を e 、そのときの電流値を i とする。また、振り子の角度を θ 、台車位置を x と表わす。ここで、台車、振り子の運動に関する運動方程式およびモータの電圧方程式を次のように記述し¹⁵⁾、用いた各定数をTable1に示す。

$$\begin{aligned} & \left\{ M_c + M_p + \frac{d_1^2}{d_1} (I_b + I_{gb} + r_g^2 I_m + r_g^2 I_{gm}) \right\} \ddot{x} \\ & + M_p L_p \sin \theta(t) \cdot \ddot{\theta}(t) = - (C_c + B \frac{d_1^2}{d_1} r_g^2) \dot{x}(t) - g_2(x, i) f_r \\ & + M_p L_p \sin \theta(t) \cdot \dot{\theta}(t)^2 + d_2 r_g K_g g_1(\dot{x}, i) \cdot i(t) \end{aligned} \quad (5)$$

$$\begin{aligned} & M_p L_p \cos \theta(t) \cdot \dot{x}(t) + (I_p + M_p L_p^2) \cdot \ddot{\theta}(t) \\ & = -C_p \dot{\theta}(t) + M_p L_p \sin \theta(t) \cdot g \end{aligned} \quad (6)$$

$$L i(t) + K_e \frac{r_g}{d_1} \dot{x}(t) + R i(t) = e(t) \quad (7)$$

ここで、

$$g_1(\dot{x}, i) = \begin{cases} 0 & (|\dot{x}| \leq \delta, \text{ and } |d_2 r_g K_g i(t)| \leq f_r) \\ 1 & (\text{else}) \end{cases} \quad (8)$$

Fig.7 Inverted pendulum

Constants		Value
M_c	Mass of cart	1.896kg
M_p	Mass of pendulum	0.08kg
$2L_p$	Legnth of pendulum	2.92×10^{-1} m
I_p	Moment of inertia of pendulum around the center of gravity	6.61×10^{-4} kg · m ²
C_c	Damping coefficient of cart	0.0N · s/m
C_p	Damping coefficient of pendulum axis	6.17×10^{-4} N · s/m
δ	Insensible zone for the velocity of cart	1.0×10^{-7}
I_b	Moment of inertia of ball screw	5.81×10^{-5} kg · m ²
I_{gb}	Moment of inertia of gear pully(ball screw side)	7.17×10^{-6} kg · m ²
I_m	Moment of inertia of motor axis	1.809×10^{-4} kg · m ²
I_{gm}	Moment of inertia of gear pully(motor side)	7.17×10^{-6} kg · m ²
r_g	Gear ratio($= r_g / r_m$)	1.0
d_1	Transfer coefficient from ball screw rotation to cart displacement	2.55×10^{-3} m/rad
d_2	Transfer coefficient from ball screw torque to cart force[5-3]	3.92×10^3 1/m
d	Diameter of ball screw	1.6×10^{-2} m
f_r	Dynamic friction of cart	44.8N
K_t	Induced voltage constant of motor	2.05×10^{-1} N · m/A
B	Braking constant of motor	8.26×10^{-4} N · m · s/rad
L	Inductance of motor	3.13×10^{-3} H
R	Resistance of motor	4.25Ω
K_e	Coil induced voltage constant of motor	2.04×10^{-1} V · s/rad

Table1 Constants of inverted pendulum model

$$g_2(\dot{x}, i) = \begin{cases} 1 & (\dot{x} > \delta), \text{ or, } (|\dot{x}| \leq \delta, \text{ and, } d_2 r_g K_t i(t) > f_r) \\ 0 & (|\dot{x}| \leq \delta, \text{ or, } |d_2 r_g K_t i(t)| \leq f_r) \\ -1 & (\dot{x} < -\delta), \text{ or, } (|\dot{x}| \leq \delta, \text{ and, } d_2 r_g K_t i(t) < -f_r) \end{cases} \quad (9)$$

またここではセンサの故障として、(10)式のようにθのセンサ倍率が係数kによって変化する場合について考察する。

$$\hat{\theta} = k \cdot \theta \quad (10)$$

4.2 CNN の設計

4.2.1 レベル 1 の設計

まず初めに、上記倒立振子を線形モデルで記述し、最適レギュレータを設計した。そして、これらの入出力関係をFig.8 の一つのニューラルネットワークに学習させた¹²⁾。学習にはBP法を用いた。

このときのコントローラの性質の一例をFig.9に示す。制御入力は、サーボモータの出力限界から、35 [v] 以上、-35 [v] 以下はそれぞれ一定値とした。

Fig. 8 Neural network controller of level1

Fig. 9 Charactorisics of neural network controller of level1

(a) Simulation result of Level1 (k=0.7)

(b) Simulation result of Level1 (k=0.6)

Fig. 10 Simulation result of Level1

レベル 1 のコントローラは、kを1.0から0.7程度まで変化させたパラメータ変動が発生しても、倒立振子を安定化することができた。Fig.10 にkが0.7と0.6の場合のシミュレーション結果を示す。このとき、シミュレーションの初期値はθが0.1 [rad], xが0.2 [m] でそれぞれの微分値は0とした。

4.2.2 レベル 2 の設計

レベル 2 では、定性的なコントローラを設計する。ここでは各変数を第一中間層の3つのニューロンによって、Negative Big, Negative Small, Positive Small, Positive Big の4つの領域に分ける抽象化を行う。

レベル 2 のニューロコントローラは、前述のファジィニューラルネットワークを用いて、レベル1のニューロコントローラの入出力特性を抽象化して学習することによって設計した。ここで、第一中間層のしきい値も同時に学習した。このとき用いたニューロコントローラをFig.11に、そのときの出力特性をFig.12に示す。

このようにレベル 2 のコントローラを設計すると、kが1.0から0.3程度までは倒立振子を安定化することができるようになった。しかし、Fig.13のように制御出力がレベル1のコン

Fig. 11 Neural network controller of level2

Fig. 12 Characteristics of neural network controller of level2

Fig. 13 Simulation result of Level2 ($k=0.3$)

トローラに比べて滑らかでなくなり、多くのエネルギーが必要となるため、制御性能は低下してしまう。

4.2.3 レベル3の設計

レベル3ではより抽象度の高いコントローラを設計する。ここでは、各変数を Negative, Positive の2つの領域に分ける抽象化を行う。学習はレベル2と同様な方法で行う。このときのコントローラをFig.14に示し、このコントローラの出力特性をFig.15に示す。また、このコントローラを用いた場合は、Fig.16のシミュレーション結果から、 k が0.1になっても倒立振子を安定化することができることが分かる。しかし、この場合の制御入力 e_1 は激しく変動し、レベル2よりも多くのエネルギーが必要となる。しかも台車位置の応答が悪くなってしまう。

Fig. 14 Neural Network Controller of Level3

Fig. 15 Characteristics of Neural Network Controller of Level3

Fig. 16 Simulation result of Level3 ($k=0.1$)

4.2.4 評価部の設計

レベル1のコントローラの評価を行うシステムニューラルネットワークをFig.17に示す。その学習は、プラント入出力値、すなわち、抽象化されていない計測値そのものを用いてコントローラの学習と同様の方法で行う。

また、レベル2のシステムニューラルネットワークの学習もコントローラと同様に、レベル1のシステムニューラルネットワークの入出力特性を学習する。レベル2のシステムニューラルネットワークをFig.18に示す。入力 e_1 は第一中間層において抽象化され、出力層において、再度同じ抽象化の領域に分けられるようにしている。そのため教師信号は計測値そのものではなく、レベル2のシステムニューラルネットワークの最終層と同様な抽象化を行ったものを用いる。レベル2とレベル3の切り替えは、レベル2のシステムニューラルネットワークを用いるので、レベル3はシステムニューラルネットワークおよび評価ネットワークを持たない。

前章で述べたように、統合化を行うために用いられる各レベルの評価ネットワークは各レベル別々にGAを用いて学習する。ここでは、前節までの比較と同様に、パラメータ k のみを変化させて、どのような状態量の差があったときにコントローラ出力を抑制すべきかを学習させた。

Fig. 17 System Neural Network of Level1

Fig. 18 System Neural Network of Level2

具体的には、 k を $1.0 \sim 0.1$ の間で 0.1 刻みのセンサ倍率で変化させ、各センサ倍率で数通りのシミュレーションを行なった。この際、適応度は倒立している時間とし、倒立時間が長いものほど安定とした。しかもできるだけ上位レベルを用いないようにすることで、制御性能も同時に得られるように適応度を(11)式のように設定して学習を行った。

$$J = \sum_{k=1}^N \frac{t_s}{\sum_{l=0}^L S_{Level}} \quad (11)$$

ここで、 N はシミュレーション回数、 t_s は倒立振子が安定性を維持している時間、 t_a は制御時間、 S_{Level} は選択されたレベル数を示す。

そして、評価ネットワークの重み、しきい値を染色体上にコード化し、50個体用意した。ただし、各染色体上の遺伝子は $-1.0 \sim +1.0$ の範囲でランダムに決定した。そして、適応度比例戦略とエリート保存戦略を併用し、2点交差と確率 0.5 の突然変異を行なった。数通りの初期値からのシミュレーションを、50通りの各重みのネットワーク全てについて行い、この際得られた結果から適応度を計算し、これに基づいて次世代の個体を決定した。

4.3 シミュレーション結果

本節では、 k が変化してそのレベルのコントローラの安定限界に達したときに、CNNの各レベルのコントローラがどのように切り替わることによって対応しているかをシミュレーションの結果によって示す。

4.3.1 k が 0.7 のときの制御性能

前述のように、 k が 0.7 になったときは、レベル1のコントローラだけでは振子は倒れてしまう。しかしこの際にレベル2に切り替わることによって安定化が図られていることがFig.19のシミュレーション結果から分かる。

4.3.2 k が 0.2 のときの制御性能

前述のように、 k が 0.2 になったときは、レベル1ではもちろんのこと、レベル2のコントローラを用いても振子は倒れて

Fig. 19 Simulation result of level1-2-3 CNN ($k=0.7$) Fig. 20 Simulation result of level1-2-3 CNN ($k=0.2$)

しまう。しかしこの際にもレベル3に切り替わることで安定化されていることを、シミュレーション結果の Fig.20 に示す。

4.4 実験結果

実験では各レベルで不安定になる k の値がシミュレーションよりも大きくなり、レベル1で0.8、レベル2で0.4程度になった。それぞれの実験結果を以下に示す。

4.4.1 k が 0.8 のときの制御性能

k が 0.8 のときの制御実験を行った結果を Fig.21 に示す。シミュレーションと同様に、レベル1からレベル2に適切に切り替わって安定化していることが分かる。

4.4.2 k が 0.4 のときの制御性能

k が 0.4 のときの制御実験を行った結果を Fig.22 に示す。こ

Fig. 21 Experimental result of CNN controller ($k=0.8$)

Fig. 22 Experimental result of CNN controller ($k=0.4$)

こでも、シミュレーションと同様にレベル2からレベル3に切り替わることによってほぼ安定化、すなわち倒立状態を維持していることが分かる。しかし、倒立を重視した統合部の学習を行っているため、上位のコントローラの出力が多くなれば位置決めの効果が減少し、台車の位置が移動しやすくなる結果となっている。また、Fig.12, 15に示すように、レベル2, 3では制御出力特性が階段状になっているため、状態量の微小な変化に対して出力が大きく変化する。しかも、その境界が倒立の安定点である原点にきているために、倒れはしないが振子を静止させることはできない。

4.5 予期せぬパラメータ変動に対するロバスト性の検証

前節では、センサ倍率に変化があった場合に、統合化部においてこの事態を想定した学習を行い、得られたコントローラを用いることによって安定化が可能であることを示した。これに対し本節では、想定していない予期せぬパラメータ変動に対するロバスト性について調べた。ここでは例として、振子の長さに変化がある場合について考える。

本CNNコントローラでは、レベル1のコントローラだけでも高いロバスト性を有し、振子の長さが学習時の0.4倍程度に短くなった場合でも安定化が可能である。しかしそれ以上短くなると倒れてしまう。そこで、学習時の0.3倍になった場合のCNNコントローラのシミュレーションを行った。結果を Fig.23 に示す。この結果から、学習時には想定しなかったパラメータ変動に対してもロバスト性を有することが示された。

5. おわりに

本論文では、定量的な制御だけでなく複数の定性的な制御を、情報の抽象度という尺度に基づいて整理し、これらを同時に並列処理する知的制御手法を提案した。そして、この考え方の実現方法としてCNNを用いた体系的な設計手法を与えた。

Fig. 23 Simulation result of CNN (pendulum length of 0.3 times)

これにより、多次元の定性的な制御を適切に利用することができるようになり、適用範囲の広いロバストな知的制御を実現することができた。

そして、本手法の有効性を確認するために、大きなパラメータ変動がある倒立振子の制御実験を行った。その結果、抽象度の高い定性的な情報を使ったコントローラは、実際の計測値を用いた定量的な制御よりも性能については劣るものの、高い安定性を得ることができ、さらにこれらを統合することによって、性能と安定性が適切に評価された知的なコントローラを作成することができることが示された。さらに、振子の長さが増えるような、設計時に想定していない予期せぬパラメータ変動に対しても、ロバスト性を有することが示された。

今後は、精度の良いモデルの獲得が難しく、しかも、知的制御を必要としているプラントなどの対象への適用を試みる。

参 考 文 献

- 1) D. E. Rumelhart, J. L. McClelland and the PDP Reserch Group : Parallel Distributed Processing, MIT Press (1986)
- 2) J. Rasmussen : Skills, Rules and Knowledges ; Signals, Signs, and Symbols, and Other Distinctions in Human Performance Models, IEEE Trans. on Systems, Man and Cybematics, **13**-3, 257/266 (1983)
- 3) G. N. Saridis : Toward the Realization of Intelligent Controls, Proceedings of IEEE, **67**-8, 1115/1132 (1979)
- 4) K. S. Fu : Learning Control System and Intelligent Control Systems, An Intersection of Artificial Intelligence and Automatic Control", IEEE Transactions on Automatic Contral, **16**-1, 70/72 (1971)
- 5) J. A. Albus, R. Lumia, J. Fiala and A. Wavering : NASREM the NASA/NBS Standard Reference Model for Telerobot Control System Architecture, Proceedings of 20th ISIR, **1**-7, (1989)
- 6) R. A. Brooks : A Robust Layered Control System for a Mobile Robot, IEEE Journal of Robotics and Automation, **RA**-2-1, 14/23 (1986)
- 7) 吉田, 木通 : 多次元ニューラルネットワークによる制御手法, 日本機械学会講演論文集, No.940-10 (I), 118/119 (1994)
- 8) 木通, 吉田, 神谷 : 自動生成統合化部を持つ多次元ニューラルネットワークによる制御手法, 日本機械学会講演論文集, No.95-1 (I), 99/101 (1995)
- 9) 上野, 石塚 : 知識の表現と利用, オーム社 (1987)
- 10) 溝口, 古川, 安西 (編) : 定性推論, 共立出版 (1989)

- 11) 古橋 : ニューラルネットワークによるファジィ制御, システム/制御/情報, Vol. 38, No. 11, 607/612 (1994)
- 12) 今泉, 山海 : 非線形系の最適ニューラルネットワーク制御 (LQN 制御), 計測自動制御学会第 32 回学術講演会予稿集, 245/246 (1993)
- 13) 長門, 吉田 : ニューラルネットワークによるアクティブ動吸振器の制御, 日本機械学会論文集, Vol. 58, No.550, C, 1755/1761 (1992)
- 14) 西田 : 定性推論の基礎, 人工知能学会誌, Vol.4, No.5, 34/39 (1989)
- 15) 市川 : GA の制御設計問題への応用, 日本機械学会講演論文集, No. 93043, 31/32 (1993)
- 16) 吉田, 下郷, 村野 : 弾性構造系のデジタル最適制御 (弾性回転アームの位置決め制御), 日本機械学会論文集, Vol.52, No.484, C, 3073/3081 (1986)

「著 者 紹 介」

木 通 秀 樹

1988年、慶應義塾大学理工学部機械工学科卒業。同年、石川島播磨重工業(株)入社。現在、慶應義塾大学大学院後期博士課程在学中。主にニューラルネットワークを中心とする、知的制御技術に関する研究に従事。日本機械学会の会員。

吉 田 和 夫 (正会員)

1978年、慶應義塾大学大学院工学研究科機械工学専攻博士課程修了(工学博士)。現在、同大学理工学部教授。主に確率システム制御、運動と振動の制御および知的制御の研究に従事。システム制御情報学会、日本機械学会、日本建築学会、電気学会、日本ロボット学会、自動車技術会などの会員。

神 谷 勝

1995年慶應義塾大学大学院理工学研究科機械工学専攻修士課程修了。同年、日本電装(株)(現(株)デンソー)入社、現在に至る。在学中は、ニューラルネットワークの知的制御への応用に関する研究に従事。