

ニューロコンピュータによるロボットの行動制御

長田 茂美, 関口 実, 吉沢 英樹, 渡部 信雄, 木本 隆, 浅川 和雄
株式会社 富士通研究所

ニューロコンピュータは、人間の脳をモデルとして、多数の並列的に動作するニューロン素子を相互に結合したネットワークを構成し、高度な並列分散処理と学習機能を実現できる“柔らかい”情報処理装置である。

本稿では、脳細胞の層状構造を模倣した多階層のニューラル・ネットワーク・モデルに着目し、その動作原理を述べるとともに、仮想インピーダンス制御法と呼ぶ階層ネットワークの学習アルゴリズムと、ネットワークをモジュール化したロボット制御のための構造化階層ネットワーク・モデルを提案する。

さらに、多数のセンサを備えた移動ロボットの制御への応用例について述べ、その有効性を示す。

Neurocomputer and its application to robot control

Shigemi NAGATA, Minoru SEKIGUCHI, Hideki YOSHIZAWA, Nobuo WATANABE, Takashi KIMOTO, and Kazuo ASAKAWA

FUJITSU LABORATORIES LTD.,

1015 Kamikodanaka, Nakahara-ku, Kawasaki, 211 JAPAN

A structured network model for robot control and its learning algorithm are presented. The model is divided into two sub-networks connected each other with a short term memory to process time dependent data. We have developed the learning algorithm, pseudo impedance control, which has damped oscillation characteristic to avoid the local minimum problem.

To evaluate the network model and the learning algorithm, small mobile robots controlled by neural networks have been developed. They were taught variety of habits to play a cops-and-robbers game. Through training, the robots learned habits such as capture and escape.

1. まえがき

人間の脳は、140億以上のニューロン（神経細胞）からなる情報処理装置である。この膨大な数のニューロンは複雑に結合しあってニューラル・ネットワーク（神経回路網）を形成し、高度な並列分散処理を実現している。

ニューロコンピュータはこの脳をモデルとして、真に“柔らかい”情報処理装置の実現を目指すアプローチである。多数の並列的に動作するニューロン素子を相互に結合した構造を持つニューロコンピュータは、その高速動作とニューロン素子間の結合の重みを自動調整することによって、人間と同様な学習機能を実現できる。

ニューロコンピュータの応用により、極めて柔軟な機能を持つシステムが構築できると期待されており、運動制御^{1), 2), 3)}、パターン認識^{4), 5)}、知識処理⁶⁾など様々な分野で研究が進められている。なかでも、ロボットの運動制御分野においては、運動と制御対象の複雑さ、センサ信号等の処理系の煩雑さのために、従来の逐次処理型のノイマン型コンピュータでは限界があり、ニューロコンピュータは、自由度数や搭載センサ数の拡大、非線形領域を含んだ制御等の面で、従来の制御能力の限界を打破できるものと大きな期待が寄せられている。

そこで、筆者らは、こうした特長を持つニューロコンピュータを移動ロボットの行動制御に適用した。移動ロボットには、外部環境を捉えるための多数のセンサ、時々刻々と変化する環境に対応するための適応能力、学習能力といった柔軟性、移動中の外部環境を確実に捉えるための実時間処理等が必要である。しかし、従来のノイマン型コンピュータでその行動を制御する移動ロボットは、これらを逐次処理で実行しなければならず、しかも、センサ等の信号処理をすべての状況に対応できるよう予めプログラムしておかなければならない。それに較べると、ニューロコンピュータは、適応能力や学習能力を持つため、予想外の状況下においても、今までに学習してきた事柄に基づく補間機能により、ある程度柔軟に対応することが可能となる。

本稿では、ニューロコンピュータの核となるニューラル・ネットワーク・モデルとして、脳の階層性のアナロジーである階層ネットワークに着目し、仮想インビダンス制御法と呼ぶ階層ネットワークの学習アルゴリズムと、ネットワークをモジュール化したロボット制御のための構造化階層ネットワーク・モデルを提案する。さらに、多数のセンサを備えた移動ロボットの制御への応用例について述べ、その有効性を示す。

2. ニューラル・ネットワークと学習

2.1 ニューロンのモデル

通常、ユニットと呼ばれる、ニューロンの動作を情報

処理の機能素子としてモデル化した工学モデルを図1に示す。ユニットは入力側からの信号を重み付けによる多数決論理で処理し、一つの出力信号を発生する。これがまた、他の多数のユニットの入力信号として使われる。このモデルは、入力信号と重みとの積和演算と、その積和演算結果に対する閾値処理の二つの処理を、ニューロンの基本的性質として捉えたものである。シグモイド関数を用いると、ユニットの入出力特性は次のように定式化される。

$$x_j(t) = \sum_i y_i(t) w_{ji}(t), \quad (1)$$

$$y_j(t) = 1 / [1 + \exp \{-x_j(t) + \theta_j(t)\}] \quad (2)$$

但し、

x_j : ユニット j の総入力

y_j : ユニット j の出力

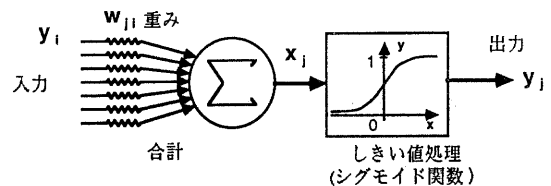
θ_j : ユニット j のバイアス値

y_i : ユニット j の入力側に結合した
ユニット i の出力

w_{ji} : ユニット i とユニット j との結合の重み

t : 学習（重みの更新）回数

である。



$$x_j = \sum_i y_i w_{ji}$$
$$y_j = \frac{1}{1 + e^{(-x_j + \theta_j)}}$$

図1. ニューロンの工学モデル

2.2 ニューラル・ネットワーク

ニューラル・ネットワークは、上述した多数のユニットを相互に結合した計算モデルであり、その結合形態によって様々なバリエーションが生まれる。ここでは、脳細胞の層状構造を模倣した階層ネットワークの動作原理を説明する。

入力層・中間層・出力層の3層からなる階層ネットワーク構造を図2に示す。階層ネットワークにおける各ユニットは、入力層から中間層、中間層から出力層という向きに結合しており、各層内での結合および出力層から入力層に向かう結合は存在しない。

入力層に入力パターンが与えられると、入力層の各ユニットはその入力信号を中間層に渡す。中間層の各ユニットは式(1)、(2)に基づいて動作し、出力層の各ユニットに出力信号を渡し、最終的にはネットワークの出力パ

ターンが出力層の各ユニットから出力される。つまり、ネットワークは、入力を常に“前向き”に処理することによって、ユニット間の結合の重み w から決定される一定の変換規則（データ処理法）を入力パターンに適用し、出力層にその変換パターンを出力する。

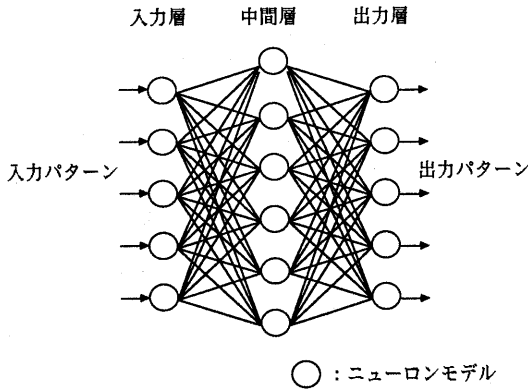


図2. 階層ネットワーク

2.3 学習アルゴリズム—仮想インピーダンス制御法—

収束の速い学習方式として、機械振動系とのアナロジーからネットワークの挙動を二次の強制振動系として取り扱う方式を考案した。これを“仮想インピーダンス制御法”と呼ぶ。ネットワークの学習とは、与えられる個々の入力パターンに対して、ネットワークが実際に出力する出力パターンと“望ましい”出力パターン（教師信号）とが一致する（または、充分に似ている）ような w の集合を見出すことである。このような階層ネットワークの学習方式として、Rumelhart らが提案したバックプロパゲーション法⁷⁾がよく知られている。バックプロパゲーション法は、階層ネットワークに図3に示すような誤差フィードバックを導入し、この誤差フィードバックに基づいて、結合の重み w とバイアス値 θ とを適応的に自動調整する。その結果、ネットワークは入力パターンと出力パターンとの間の連合を学習し、適応的なデータ処理を実現できるようになる。

仮想インピーダンス制御法は、このバックプロパゲーション法の基本的な考え方を継承し、これに w の更新過程と機械振動系とのアナロジーに着目したパラメタ設定を適用した、より高速な学習を可能とする方式である。

ある入力パターン c を与えたときの、実際の出力値 $y_{j,c}(t)$ と望ましい出力値 $d_{j,c}$ との誤差を、

$$E_c(t) = \frac{1}{2} \sum_j (y_{j,c}(t) - d_{j,c})^2 \quad (3)$$

但し、

c : 入出力パターン対のインデックス

j : 出力ユニットのインデックス
と定義する。このとき、すべての入出力パターン対に対するネットワークの全誤差 $E(t)$ は、次式で表現される。

$$E(t) = \frac{1}{2} \sum_c \sum_j (y_{j,c}(t) - d_{j,c})^2 \quad (4)$$

仮想インピーダンス制御法は、この誤差を減らす方向に w を順次更新していき、最終的に出力層のユニットの出力値と望ましい出力値との誤差が、所定の値以下になると学習を終了する。

以下に、仮想インピーダンス制御法のアルゴリズムを、図3を参照しながら具体的に説明する。式(1)、(2)から、ネットワークの各ユニットの入出力関係は、次のように表現できる。

$$x_{i,c}(t) = \sum_h y_{h,c}(t) w_{ih}(t) \quad (5)$$

$$y_{i,c}(t) = 1 / [1 + \exp \{-x_{i,c}(t) + \theta_i(t)\}] \quad (6)$$

$$x_{j,c}(t) = \sum_i y_{i,c}(t) w_{ji}(t) \quad (7)$$

$$y_{j,c}(t) = 1 / [1 + \exp \{-x_{j,c}(t) + \theta_j(t)\}] \quad (8)$$

但し、

$y_{i,c}$: ユニットの出力

$x_{i,c}$: ユニットへの入力の総和

w_{ih} : h 番目のユニットと i 番目ユニットの結合の重み

θ_i : ユニットのバイアス値

ここで、 h, i, j は各々、 H 層（入力層）、 I 層（中間層）、 J 層（出力層）の h, i, j 番目のユニットを表すインデックスで、 c は入出力パターン対のインデックスである。

仮想インピーダンス制御法では、この誤差 $E(t)$ を最急降下法を用いて最小化するために、各々の $w(t)$ に関する誤差 $E(t)$ の偏微分を計算する。式(3)、(4)から次式が成立する。

$$\partial E(t) / \partial w_{ji}(t) = \sum_c \partial E_c(t) / \partial w_{ji}(t) \quad (9)$$

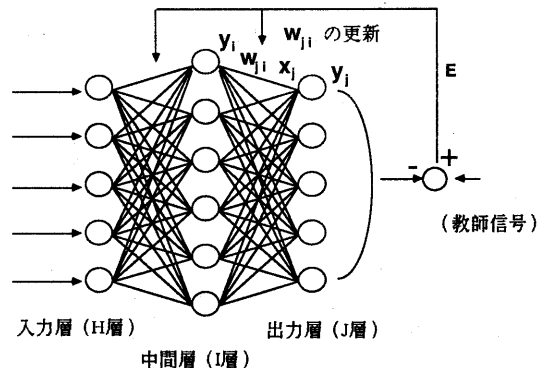


図3. ニューラル・ネットワークの学習

$$\partial E(t)/\partial w_{ih}(t) = \sum_c \partial E_c(t)/\partial w_{ih}(t), \quad (10)$$

これらの式は、出力層-中間層、中間層-入力層の間の $w(t)$ の変化に対する $E(t)$ の変化率を表している。したがって、この値が常に負になるように、 $w(t)$ を変化させて $E(t)$ を漸近的に 0 とすることができる。

ここで、仮想インピーダンス制御法では、ネットワークの挙動に時間変化を適用して、 $-\partial E(t)/\partial w(t)$ を外力とする次のような二次の強制振動系として取り扱い、学習（重みの更新）回数を低減している。

$$M\Delta \ddot{w}(t) + D\Delta \dot{w}(t) + K\Delta w(t) = -\partial E(t)/\partial w(t), \quad (11)$$

但し、

- M: ネットワークの仮想的な質量
- D: ネットワークの仮想的な粘性制動係数
- K: ネットワークの仮想的なスティフネス
- t: 学習（重みの更新）回数
- w(t): 結合の重み
- $\Delta w(t)$: t における結合の重み w の変化
- E(t): 誤差

この二次の微分方程式を離散近似し、 $\Delta w(t)$ について解くと次式が得られる。

$$\begin{aligned} \Delta w(t) = & \{-\Delta t^2 \partial E(t)/\partial w(t) \\ & + (2M+D\Delta t)\Delta w(t-\Delta t) \\ & - M\Delta w(t-2\Delta t)\} / (M+D+K\Delta t^2). \end{aligned} \quad (12)$$

ここで、 $\Delta t = 1$ とすると、

$$\begin{aligned} \Delta w(t) = & -\varepsilon \partial E(t)/\partial w(t) \\ & + \alpha \Delta w(t-1) + \beta \Delta w(t-2). \end{aligned} \quad (13)$$

ここに、

$$\begin{aligned} \varepsilon &= 1/(M+D+K), \\ \alpha &= (2M+D)/(M+D+K), \\ \beta &= -M/(M+D+K). \end{aligned} \quad (14)$$

ここでは、二次の系として取り扱ったが、より一般的に n 次の系として考えることもできる。

式(12)において、 $\Delta t = 1$, $M = 0$, $1/(D+K) = \varepsilon$, $D/(D+K) = \alpha$ とおくと、仮想インピーダンス制御法は、以下に示す学習定数 ε 、学習速度定数 α を導入したバックプロパゲーション法と等価になる。

$$\Delta w(t) = -\varepsilon \partial E(t)/\partial w(t) + \alpha \Delta w(t-1). \quad (15)$$

但し、t は学習（重みの更新）回数である。つまり、仮想インピーダンス制御法は、学習の加速度項までを考慮し、さらに、 ε と α との相関も考慮した適切な ε と α の組み合わせを決定できるため、学習の収束速度を上げ

ることができる。

結合の重み $w(t)$ の変化に対する誤差 $E(t)$ の変化率 $\partial E(t)/\partial w(t)$ の導出は、バックプロパゲーション法と全く同様に、“後向き”の処理、すなわち、誤差フィードバックを導入し、誤差の微分値を出力層から入力層側に伝播させればよい。

まず、誤差 $E_c(t)$ と出力層（J 層）の出力値との関係を得るため、式(3) を $y_{j,c}(t)$ に関して偏微分すると、

$$\begin{aligned} \partial E_c(t)/\partial y_{j,c}(t) &= y_{j,c}(t) - d_{j,c} \\ &= \delta_{j,c}(t). \end{aligned} \quad (16)$$

となる。次に、出力ユニットへの総入力の変化が誤差にどのように影響するかを知るために、誤差 $E_c(t)$ を $x_{j,c}(t)$ で偏微分すると、

$$\begin{aligned} \partial E_c(t)/\partial x_{j,c}(t) &= \delta_{j,c}(t) y_{j,c}(t) \{1 - y_{j,c}(t)\}. \end{aligned} \quad (17)$$

を得る。したがって、I-J 層間の結合の重み $w_{ji}(t)$ の変化に対する誤差 $E_c(t)$ の変化の関係は、

$$\begin{aligned} \partial E_c(t)/\partial w_{ji}(t) &= \delta_{j,c}(t) y_{j,c}(t) \{1 - y_{j,c}(t)\} y_{i,c}(t). \end{aligned} \quad (18)$$

となる。

次に、中間層（I 層）の出力値 $y_{i,c}(t)$ に対する誤差 $E_c(t)$ の変化を求めると、

$$\begin{aligned} \partial E_c(t)/\partial y_{i,c}(t) &= \sum_j \delta_{j,c}(t) y_{j,c}(t) \\ & \cdot \{1 - y_{j,c}(t)\} w_{ji}(t). \end{aligned} \quad (19)$$

を得る。さらに、I 層ユニットへの総入力 $x_{i,c}(t)$ の変化に対する誤差 $E_c(t)$ の変化を計算すると、

$$\begin{aligned} \partial E_c(t)/\partial x_{i,c}(t) &= (\sum_j \delta_{j,c}(t) y_{j,c}(t) \{1 - y_{j,c}(t)\} w_{ji}(t)) \\ & \cdot y_{i,c}(t) \{1 - y_{i,c}(t)\}. \end{aligned} \quad (20)$$

となる。したがって、H-I 層間の結合の重み $w_{ih}(t)$ の変化に対する誤差 $E_c(t)$ の変化の関係は、

$$\begin{aligned} \partial E_c(t)/\partial w_{ih}(t) &= (\sum_j \delta_{j,c}(t) y_{j,c}(t) \{1 - y_{j,c}(t)\} w_{ji}(t)) \\ & \cdot y_{i,c}(t) \{1 - y_{i,c}(t)\} y_{h,c}(t). \end{aligned} \quad (21)$$

と表現される。

このようにして、式(9)、(10) と式(18)、(21)から、各層間の結合の重み $w(t)$ の変化に対する誤差 $E(t)$ の変化率が求まる。したがって、この値が常に負になるように、結合の重み $w(t)$ を式(12)に基づいて変化させることにより、誤差 $E(t)$ を漸近的に 0 にできる。

2.4 学習の収束性に関する実験

仮想インピーダンス制御法を評価するため、式(13)の各項の係数 ε , α , β を変化させ、その収束の様子を調べた。図4(a)は、後述するロボットの教示パターン約60個を階層ネットワークに学習させたときの誤差の2乗和Eと学習回数との関係を調べたものである。図では、 $\varepsilon=0.1$, $\alpha=0.5$ と固定し、 β を 0, 0.2 に変化させた場合の例を示した。この例では、 β の項を加えたことによる効果は顕れていない。これは、このロボットの教示パターンがデジタル信号であり、しかも非線形性を持たないパターンであるため、重み空間の誤差曲面に極小値(ローカル・ミニマ)がそれほど多く存在しないためと考えられる。つまり、重み空間の誤差曲面の勾配を下っていく際に、その軌跡に振動を発生させる β の項は、却って収束を遅くしていることになる。しかし、ロボット・アームのトルク制御のようなアナログ値で、しかも非線形性を持ったパターンを学習する際には、重み空間の誤差曲面は、誤差の減少方向が一樣でなく、多数のローカル・ミニマを持つと思われる。

そこで、筆者らは、ロボットアームの加減速時によく採用される SIN関数の学習実験を試みた。教示パターンとしては、次式に示す SIN関数の1周期分を20分割したデータを使用し、入力ユニット1, 中間ユニット10, 出力ユニット1からなる3層ネットワークに、100,000回の学習を実行させた。

$$\begin{aligned} Y_i &= \sin(2\pi X_i / 20), \\ X_i &= i, \quad (i = 0, 1, \dots, 19). \end{aligned} \quad (22)$$

但し、 X_i は入力パターン、 Y_i は“望ましい”出力パターンである。

図4(b)は、 $\varepsilon=0.6$ について、① $\alpha=0, \beta=0$, ② $\alpha=0.9, \beta=0$, ③ $\alpha=1.2, \beta=-0.4$ のパラメータ値

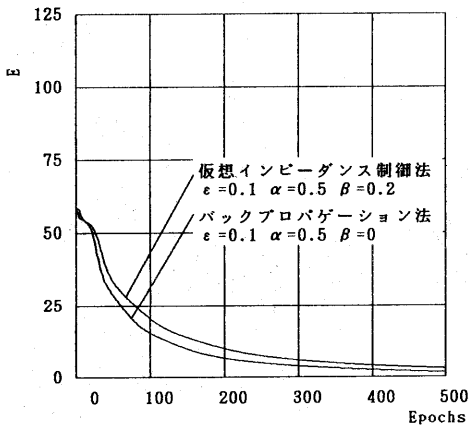


図4(a) シミュレーション結果(行動パターン)

に対する収束の様子を示したものである。収束を速めるためには、 ε を可能な限り大きく設定した方がよいが、①の場合には、振動が大きくなり収束しない。②の場合には、 α の項の持つダンパ効果が効いて収束するが、学習が進むにつれ、その収束速度は次第に遅くなり、誤差は減少しなくなる。これは、アナログ信号で、かつ、非線形性を持つ関数の学習であるため、数多く存在するローカル・ミニマに落ち込み、その状態を脱出できないためと考えられる。そこで、③の場合のように、 β の項を加えると、振動的にはなるが、このようなローカル・ミニマを脱出し、グローバル・ミニマムの状態に到達できることが分かる。

このように、仮想インピーダンス制御法は、 β の項の効果により、ローカル・ミニマムを脱出し、より誤差の少ないグローバル・ミニマムに向かって学習を進行させることができる。

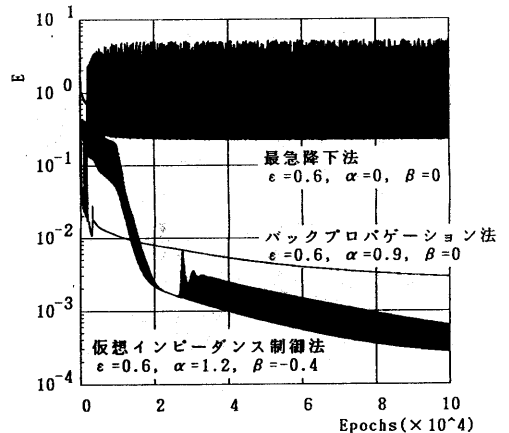


図4(b) シミュレーション結果(SIN関数)

3. ロボットの行動学習制御

ニューロコンピュータの採用によって、きわめて柔軟な機能をもつシステム構築が可能となり、運動制御、パターン理解、知識処理など、様々な応用分野が考えられる。筆者らは、運動制御分野をとりあげ、ニューラル・ネットワーク・モデルを小型移動ロボットの制御に適用し、ニューロコンピュータの有効性の実証を試みた。

3.1 小型移動ロボット

今回開発した小型移動ロボットの主要諸元を表1に、その概観を図5に示す。ロボットは四つの車輪と二つのモータ(推進用と操舵用)を持ち、4輪を同時に同方向に操舵しながら移動できる。また、外部環境や自己の内部状態を認識するための超音波センサ、光センサなどを

備えている。

このロボットを素材として、ニューラル・ネットワークの“柔軟さ”を実証するシステムを検討し、複数のロボット達に追跡劇を演じさせる実証システムの開発を目指した。

この実証システムの開発には、二つのねらいがある。一つは、ロボットが様々なセンサ入力情報に基づいて、

表1. ロボットの主要諸元

最大加速度	0.9 m/s ²	発光LED	16個		
最高速度	1.0 m/s	搭載センサ	光センサ	4個	
操舵速度	142 °/s		超音波	23KHz	1個
M P U	MB89702		センサ	40KHz	2個
電源部	6Vバッテリー2個直列		タッチセンサ	2個	
走行操舵系	DCモータによるON-OFF制御		リミットセンサ	3個	
寸法	φ260×315mm	合計	12個		
重量	5.4 kg	その他	圧電ブザー、ランプ		

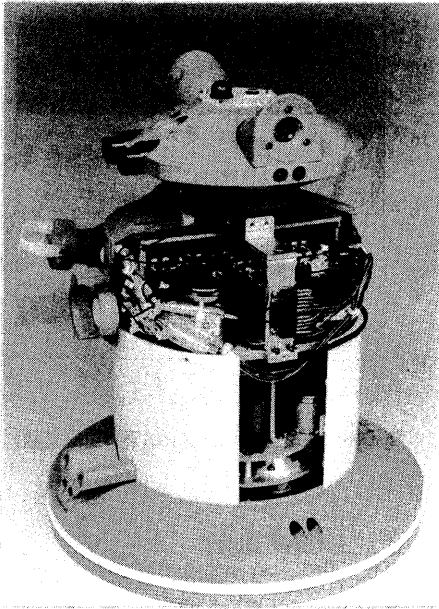


図5. 小型移動ロボット

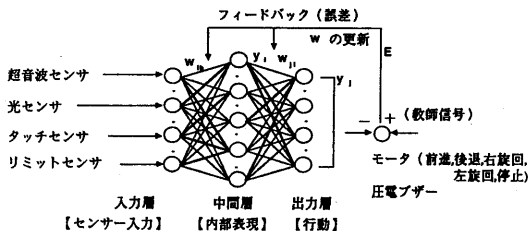


図6. 階層ネットワークによるロボット制御

自らの行動パターンをニューラル・ネットワークを通じて学習し、時々刻々と変化する外部環境に応じて適切に行動できるように成長させ、ニューラル・ネットワークの学習機能を実証することであった。そのため、ニューラル・ネットワークをロボットに搭載し、図6のようにその入力パターンとしてセンサ入力情報を与え、出力パターンとしてその時点での適切な行動パターン (モータの制御信号など) を出力させることを目指した。

もう一つは、学習のさせ方、すなわち、センサ入力情報と行動パターンの対応関係を変え、同一構造のロボットから様々な習性 (行動パターン) をもつロボットを生み出し、ニューラル・ネットワークの柔軟性の実証を試みることであった。

3.2 実証システム -ロボット追跡劇-

上述の目的で開発した実証システムとして、複数のロボット達が演じる追跡劇の概要を図7に示す。同一の機構およびニューラル・ネットワーク構造をもつ複数のロボットは、学習によって“捕獲”ロボットの“ホームズ”にも、“逃走”ロボットの“ルパン”にもなり得る。

“ホームズ”は、頭部の超音波センサと光センサによって“ルパン”を探索し、その方向に追い掛ける。“ルパン”の頭部には、超音波発信器と90度間隔で4方向に取り付けられた発光ダイオードがあり、その信号を検出すればよい。

一方、“ルパン”は、腹部の光センサで“ホームズ”の腹部にある90度間隔で4方向に取り付けられた発光ダイオードの光を検出し、“ホームズ”から逃げる。このように、センサ入力情報と行動パターンとの基本的な対応関係を教えるだけで、一方は追い掛け、もう一方は逆に逃げるようになる。

ロボットの胴体にはタッチセンサが前後に付いており、

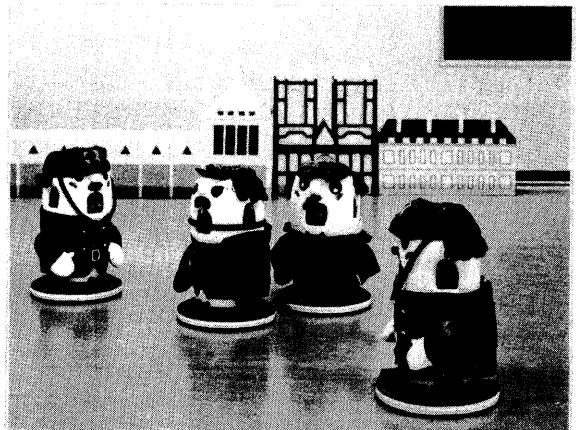


図7. 実証システム

衝突を検出するとロボットは直ちに停止して体の向きを衝突した側に向け、相手を識別する。この識別には、光センサと発光ダイオードの点灯状態を利用して、“ホームズ”または“ルバン”と障害物とを区別する。“ルバン”は“ホームズ”に捕まると、頭部に付けられたブザーを鳴らす。

以上のように、ロボット達は、時々刻々と変化するセンサ入力情報をもとに、自らの習性を学習したニューラル・ネットワークを辿りながら、その状況に応じた適切な行動を決定し、追跡劇を演じていく。なお、各ロボットは、自分の向いている方向と左右に回転する限界位置を知るためのリミットセンサを備えており、体がねじれるのを防止している。

3.3 構造化階層ネットワーク

ロボットの行動制御には、一連の行動パターンの系列（行動パターン系列と呼ぶ）の処理機能が必須となる。この行動パターン系列の処理を実現するために、筆者らは、一種のネットワークの構造化手法である複数の3層ネットワークを階層化する方式を採用し、図8に示す“構造化階層ネットワーク・モデル”を構築した。

これは“第一ネットワーク”と“第二ネットワーク”の2種類の階層ネットワークを、短期記憶型ユニットを介して結合した構成をとる。短期記憶型ユニットとは、ある一定期間その興奮状態を維持するユニットである。第一ネットワークは、ある瞬間のセンサ入力情報と行動パターンとの対応関係、すなわち、“静的”な入出力関係を学習する。例えば、“頭部の光センサが光を検出したならば、前進せよ”といった行動パターンが処理される。一方、第二ネットワークは、ある時間幅でのセンサ入力情報と行動パターン系列との対応関係、すなわち、

“動的な”入出力関係を学習する。例えば、“頭部の光センサが光を検出するまで、右回転、左回転を繰り返せ”といった行動パターン系列が処理される。

今回の実証システムでは、この構造化階層ネットワークが“探索行動”と“回避行動”と呼ぶ2種類の行動パターン系列を取り扱えるように構成した。すなわち、図8に示したように、第一ネットワークの出力ユニットと第二ネットワークの入力ユニットとの間を、興奮性と抑制性の2種類の信号で結合し、これらの信号で二つのネットワークの動作の切り換えを制御する。第一ネットワークから第二ネットワークへの抑制性の信号は、ロボットのいずれのセンサも反応していないとき、または、タッチセンサのみが反応したときだけ、解除するように学習させている。すなわち、センサが反応しなくなると、第一ネットワークから短期記憶型ユニット#2を介して、興奮性の信号が出力され、第二ネットワークに制御が渡される。このとき、第二ネットワークは、相互抑制性の短期記憶型ユニット#1を利用して、右回転、左回転のモータ制御信号を交互に出力し、第一ネットワークからの抑制性の信号が入るまでの間、ロボットに探索行動パターン系列をとらせることができる。

一方、回避行動は、ロボットが障害物に衝突したとき、ある一定時間後退して障害物を回避するといった一連の連続動作を必要とする行動パターン系列の一つである。この回避行動は、短期記憶型ユニット#2を利用し、第二ネットワークが回転と後退のモータ制御信号を発生するよう学習させればよい。この#2は一定の時間幅で興奮し続けるので、第一ネットワークからの抑制性信号が入るまでの間、ロボットに障害物を回避する行動をとらせることができる。

以上のように、センサから得られる外界情報が豊富で

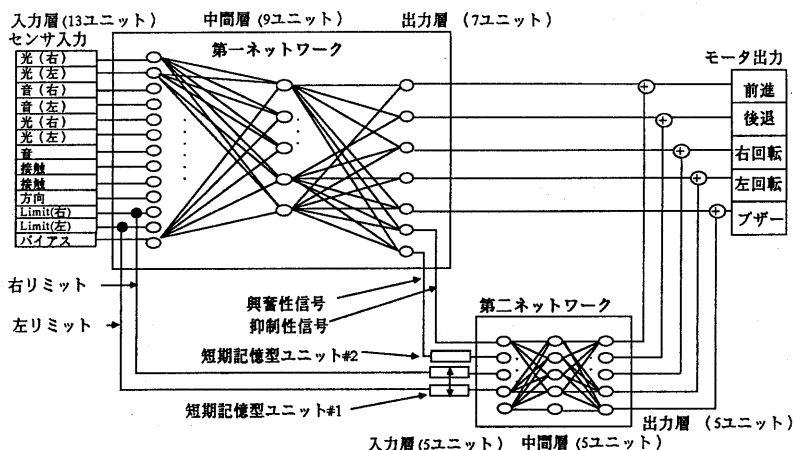


図8. 構造化階層ネットワーク

あれば、第一ネットワークのみで適切に行動させ得る。不足しているときには、第二ネットワークの出力パターンでセンサ情報が豊富になる方向、すなわち、第一ネットワークで行動できる方向に導くことができる。

単一の階層ネットワークでも、これらの行動を行わせることは可能であるが、その場合、ネットワーク規模が大きくなり、実時間制御が難しくなる。また、教示パターン数、さらには、学習時間が増大するという問題が発生し、実用的でない。構造化階層ネットワークの構成法は、ある一連の出力パターン系列をニューラル・ネットワークの構造の中に、先天的に埋め込もうとする試みであり、一種のプリワイヤ手法といえる。こうした構成法により、教示パターン数とネットワークの結線数を減らすことができ、効率的な学習が可能となる。

3.4 マルチフェーズ学習

次に示す二つのフェーズからなる学習過程を踏ませることで、ネットワーク構成の検討とそのネットワークの学習を効率よく実行することができた。

1) 行動シミュレータによる学習

ニューラル・ネットワーク・シミュレータと、複数のロボットのセンサ入力情報や行動をシミュレートできる行動シミュレータをワークステーション上に構築し、構造化階層ネットワークを開発した。

ネットワークの学習パターンを選択し、ネットワークに学習させた後、この行動シミュレータで実際の動きを検証する。この過程を繰り返して、適切な学習パターンの選択と学習を効率よく実行できた。また、最終的に決定したネットワークの結合の重みは、次の学習フェーズである実地訓練の初期値として用いられる。

なお、これらのシミュレータ開発は、すべてC言語で行い、構造化階層ネットワークは約 800ステップ、行動シミュレータは約 4000 ステップの規模となった。また、実装した構造化階層ネットワークは、ワークステーション上で開発したものを移植した。

2) ロボットの実地訓練

学習の第二フェーズは、上述したワークステーションとロボットを回線で接続した形態での実地訓練である。ワークステーション上の理性と本能ネットワークは、ロボットから生のセンサ情報を取り込み行動パターンを出力する。この実地訓練によって、初期値として与えられた結合の重み（学習の第一フェーズで決定した結合の重み）を微調整する。

実地訓練の過程で、ロボットが“拙い”行動をとった場合、教師である人間が“巧みな”行動をとれるような学習パターンを追加する。

実地訓練後、ロボットはネットワークの結合の重みを

ワークステーションから受け取り、自分自身の構造化階層ネットワークを使って、一人歩きを始める。

4. むすび

ニューロコンピュータの核となるニューラル・ネットワーク・モデルとして、脳の階層性のアナロジーである階層ネットワークに着目し、仮想インピーダンス制御法と呼ぶ学習アルゴリズムと、ネットワークをモジュール化した構造化階層ネットワーク・モデルを提案した。さらに、さらに、その応用例として、小型移動ロボットをニューラル・ネットワークで制御するシステムを構築し、その有効性を実証した。

今後は、新しい学習アルゴリズムの検討などの基礎研究と並行して、運動制御、パターン理解、知識処理などへの応用研究を進めていく予定である。

〔参考文献〕

- 1) H. Miyamoto et al.: Feedback-error-learning neural network for trajectory control of a robotic manipulator, Neural Networks, in press, 1988.
- 2) C. A. Niznik et al.: MODELING AND MATHEMATICAL FORMALISM FOR ROBOTIC CEREBELLAR NEURAL NETWORK PATHWAY INFORMATION MEASURES, IEEE ICNN Proceedings, pp. IV583-IV591 (1987).
- 3) A. G. Barto et al.: Neuron-like adaptive elements that can solve difficult learning control problems, IEEE Transactions on Systems, Man, and Cybernetics 13, pp. 834-846 (1983).
- 4) B. Widrow et al.: Neural Nets for Adaptive Filtering and Adaptive Pattern Recognition, IEEE, pp. 25-39 (1988).
- 5) S. Grossberg: Competitive Learning: From Interactive Activation to Adaptive Resonance, COGNITIVE SCIENCE 11, pp. 23-63 (1987).
- 6) S. J. Leven et al.: EFFECTS OF REINFORCEMENT OF KNOWLEDGE RETRIEVAL AND EVALUATION, IEEE ICNN Proceedings, pp. II 269-II 277 (1987).
- 7) D. E. Rumelhart et al.: Parallel Distributed Processing, MIT Press.
- 8) S. Nagata et al.: Control of Mobile Robot with Neural Networks, INNS, vol.1, pp.349 (1988).
- 9) 木本他: ニューラル・ネットワークの学習方式の検討, 昭和63年度人工知能学会全国大会, p. 123 (1988).
- 10) 長田他: ニューラル・ネットワークによる移動ロボットの制御, 昭和63年度人工知能学会全国大会, p. 127 (1988).