

不規則移動物体を追跡可能な逐次学習型 ニューラルネットワーク制御システム

大谷, 正幸 / OHTANI, Masayuki / YUBASAKI, Naoyuki /
MOTEGI, Atsushi / TSURUMARU, Takayuki / HIROTA, Kaoru /
廣田, 薫 / 鶴丸, 尚之 / 茂木, 篤史 / 湯場崎, 直養

(出版者 / Publisher)

法政大学工学部

(雑誌名 / Journal or Publication Title)

法政大学工学部研究集報 / 法政大学工学部研究集報

(巻 / Volume)

29

(開始ページ / Start Page)

67

(終了ページ / End Page)

80

(発行年 / Year)

1993-03

(URL)

<https://doi.org/10.15002/00003861>

不規則移動物体を追跡可能な 逐次学習型ニューラルネットワーク制御システム

廣田 薫*・鶴丸尚之**・茂木篤史*
湯場崎直義***・大谷正幸***

A Successive Learning Control System shooting Irregular Moving Object by using Neural Network

Kaoru HIROTA*, Takayuki TSURUMARU**, Atsushi MOTEGI*
Naoyuki YUBASAKI***, Masayuki OHTANI***

Abstract

As an example of realizing human real-time learning process, a successive learning control system has been developed.

A 3-freedom arm robot shoots a ball at an irregular moving basket, where the hitting rate will be improved by the successive learning process based on a neural network technique with a genetic algorithm.

§ 1 はじめに

近年知能ロボットの制御分野の研究に多くみられる制御器のパラメータ調整の方法には、ニューラルネットワークによる学習や最急降下法による学習などが多用されている^{1) 2) 3)}。しかし、これらの方法では、試行錯誤的な実験を繰り返し、膨大な入出力データを採取し解析するため、多大な労力を必要とする。

人間はある行動において成功や失敗を繰り返す毎に徐々に能力を向上することが可能である。この動作では、人間が試行を行う毎に1組の入出力データを得て、その直後にそれらのデータを生かし、次の動作に役立てるといった実時間的な学習処理が行われている。

*工学部電気工学科計測制御専攻

**大学院工学研究科システム工学専攻

***マイコム(株)

本研究の目的は、この過去の経験を近未来に反映させようとする人間行動をアームロボットを用いて実現させることである。その 1 例として、不規則に移動するバスケットにボールをシューティングする毎に学習を行い、徐々に能力を向上させる逐次学習型ロボットを構築する。

また、今回の研究には制御器としてニューラルネットワークを使用し、さらに、ニューラルネットワークの学習には、実時間的かつ 1 組のデータを得る毎に学習を行うため、遺伝的アルゴリズム (GA) を使用することにより学習の高速化を図った。

§ 2 シューティングシステムの構成

2.1 シューティングシステムの構成

本研究に使用した機器を以下に示す。また、シューティングシステムの概略図を図 1 に示す。

a) シューティングシステム

- ロボット…………… 3 自由度垂直多関節型アームロボット (マイコム株式会社)
- 画像処理装置……………画像処理ボード (TRUE VISION TARGA M8)
 CCDカメラ (SONY XC-37)
 画像モニタ (teli BE1251A1)
- 計算機……………パーソナルコンピュータ (32bit、20MHz)
 (マイコム株式会社 MPC 3440E-2)
 CRT (mitac MTR-14VC)

b) バスケット移動システム

- ロボット…………… 5 自由度垂直多関節型アームロボット
 (三菱電機 MOVEMASTER EX)
- 計算機……………パーソナルコンピュータ (16bit、5 MHz)
 (NEC PC-9801)
 CRT (NEC PC-8853K)

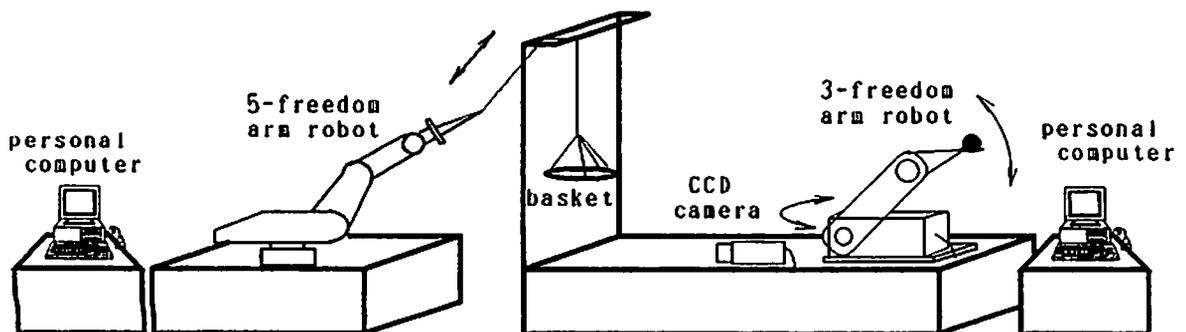


図 1 : シューティングシステムの構成

バスケットをアームロボットにより不規則に運動させ、その行動をCCDカメラから取り込み、ニューラルネットワークの入力情報とする。ニューラルネットワークにより、次の離散時刻におけるバスケットの位置を予測し、それに向かってもう1台のアームロボットがボールをシューティングする。シューティング後、ボールとバスケットを再びCCDカメラから取り込み、そのデータを用いてニューラルネットワークをGAを用いて学習させ、制御器を向上させる。

2.2 シューティングロボットの動作方法

シューティングに用いたアームロボットは、図2のように3つの可動部（ウェスト、エルボー、リスト）とスピードを制御することが可能である。

ボールのシューティング動作は、始めに水平方向からエルボーを33度、リストをエルボー軸から147度上げた状態をホームポジションとし、そこからリストを170度回転させて、ボールをシューティングする。また、ボールをシューティングする方向は、図3のようにウェストを-3～+3度、ショルダーを-1～+4度の範囲で決定し、さらに、ボールをシューティングするスピードは一定とした。

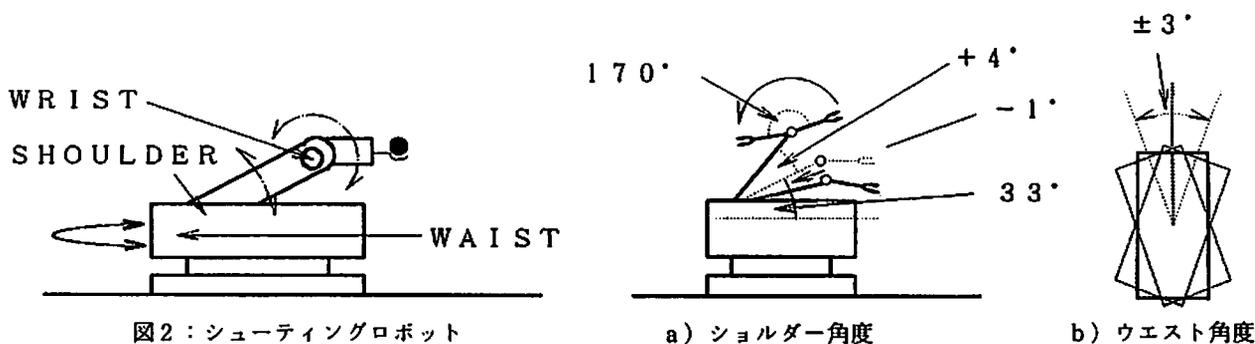


図2：シューティングロボットの3つの可動部

図3：シューティング方向の決定

2.3 バスケット移動ロボットの動作方法

バスケットを可変速度で不規則に移動させるアームロボットは、図4のように5つの可動部（ウェスト、ショルダー、リストピッチ、リストロール）を持つ。バスケットはアームロボットにより不規則に動かす。ここで言う不規則とは、例えば、熱気球が風に吹かれてふらふら移動する程度の比較的単調な動きを意味する。

アームロボットは、図5のように31×41のXY方向に等間隔の座標を持つ格子を空間上に用意し、その中の1点にリストの先端を移動させることにより動作させる。格子上のX座標は次の(1)(2)式により決定した。

$$X = \text{int} (15 \times \sin (\theta / P + Q) + 16 + 0.5) \quad (1)$$

$$\theta = (\pi/20) \times Y \quad (Y = 1, 2, 3, \dots, 41) \quad (2)$$

ただし、 $1 \leq P \leq 2$ 、 $0 \leq Q \leq 2\pi$ はランダムに決定した。

Y座標は、1~41の間でランダムに+1ずつ増加させたり-1ずつ減少させたりすることで決定し、その動作を繰り返し行った。X座標は、Y座標の決定に伴って(1)(2)式により決定した。

この動作により、図5のバスケットは、リストの先端の格子上での動きに伴って不規則に運動する。

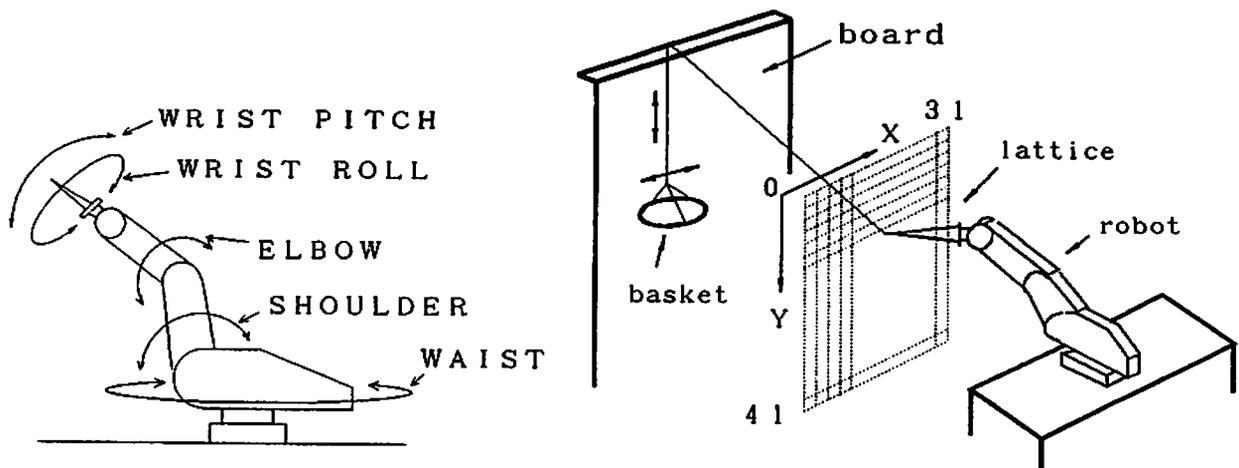


図4：バスケット移動ロボットの5つの可動部

図5：バスケットを運動させる方法

§ 3 シューティングシステムのアルゴリズム

シューティングシステムのアルゴリズムを図6に示す。

始めに、画像処理部でバスケットの座標を一定の時間間隔で検出する。現時刻のバスケットの座標を (x_n, y_n) とし、そのときの速度 (v_x, v_y) 、加速度 (a_x, a_y) を次式により計算する。

$$v_x = x_n - x_{n-1} \quad (3)$$

$$v_y = y_n - y_{n-1} \quad (4)$$

$$a_x = (x_n - x_{n-1}) - (x_{n-1} - x_{n-2}) \quad (5)$$

$$a_y = (y_n - y_{n-1}) - (y_{n-1} - y_{n-2}) \quad (6)$$

次に、ニューラルネットワーク制御部で (v_x, v_y) 、 (a_x, a_y) より次時刻のバスケットの座標 (x_{n+1}, y_{n+1}) を予測する。

ロボット動作部では、ニューラルネット制御部で予測した (x_{n+1}, y_{n+1}) をシューティングロボットのウエストの角度 θ_w 、ショルダーの角度 θ_s に変換する。 θ_w 、 θ_s は次式により算出する。

$$\theta_w = a_w x_{n+1} + b_w \tag{7}$$

$$\theta_s = a_s y_{n+1}^3 + b_s y_{n+1}^2 + c_s y_{n+1} + d_s \tag{8}$$

ただし、(7)(8)は実験的に決定し、また a_w 、 b_w 、 a_s 、 b_s 、 c_s 、 d_s は、最小2乗法により決定した。

(7)、(8)より求めた θ_w 、 θ_s からシューティングロボットのウエストとショルダーを決定し、そこから170度回転させてボールのシューティングを行う。

ボールのシューティング後、待ち時間によりタイミングを図ってバスケットの座標を検出する。

予測座標と実座標の差 (e_x 、 e_y) を求め、 $|e_x| > 20$ なら x 方向の予測について、さらに、 $|e_y| > 20$ なら y 方向の予測についても GA により学習を行う。ここでの座標とは、画像上の画素の x y 方向の位置を意味しており、以下同様の意味で用いる。

以上のことを繰り返し、逐次シューティング能力を向上させる。

なお、画像処理部、ニューラルネット制御部、GA による学習部には C 言語を、ロボット動作部には G 言語を、システム全体のリンクには QuickBASIC 言語を用いた。

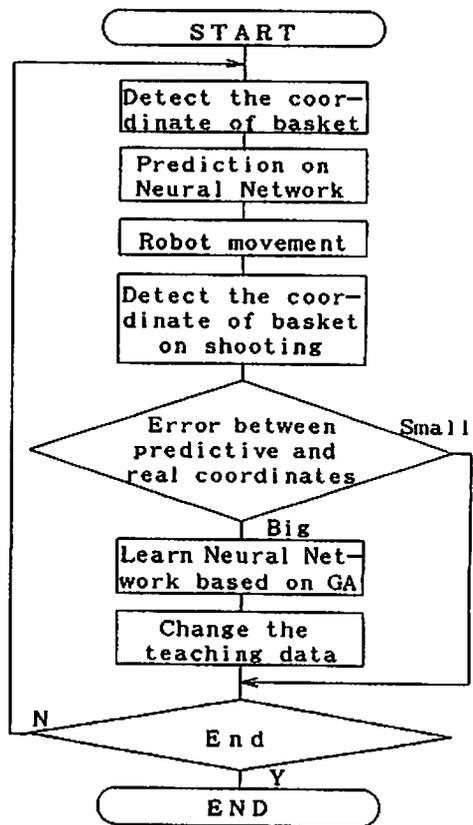


図6：シューティングシステムのアルゴリズム

§ 4 画像処理部

4.1 画像処理部

画像処理部はバスケット座標を検出するセンサ的な働きをしている。画像処理装置は 256×256 画素を持つ白黒表示のフレームメモリで、グレースケールは256階調で表現され、純黒が0、純白が255で表現される。CCDカメラからの画像を固定しきい値法により、バスケットを黒く、背景を白く表示させた。その様子を図7に示す。

画像処理を行う際にセンサ等を使用する場合と異なる点は、大量のメモリを処理するため処理に時間がかかることである。画像上では、バスケットの大きさは1画素に比べて十分大きいこと

を考慮し、図 7 の x 方向、y 方向をそれぞれ 2 画素おきに予め設定しておいた固定しきい値と比較することで、処理速度の短縮を図った。図 8 に簡略化した画像処理の様子を示す。

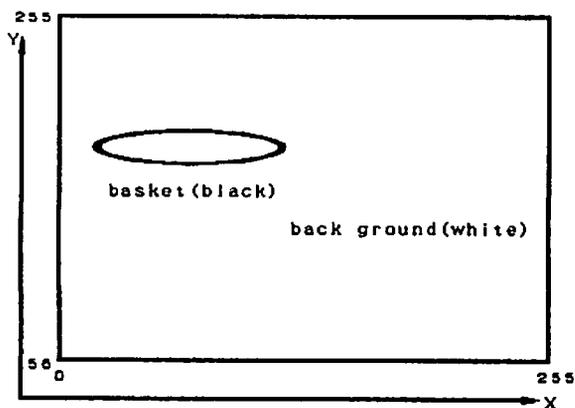


図 7 : 画像モニタにおけるバスケットの様子

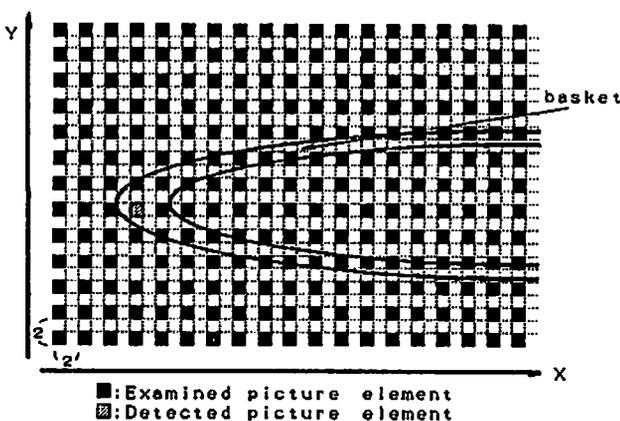


図 8 : 画像処理の簡略化

次に、画像処理部の構成について述べる。画像処理部は以下の 2 つで構成される。

- a) 不規則に運動するバスケットの中心座標の逐次検出。
- b) シューティング後のバスケットの中心座標の検出。

両者とも、1 画像のフレームにおけるバスケットの検出方法については、同一のアルゴリズムを用いている。図 9 のように、画像の左下から①の矢印の方向に検索し、最初に現れる黒い画素 (x_a, y_a) を得る。次に右上から②の矢印の方向に検索し、最初に現れる黒い画素 (x_b, y_b) を得る。そして、両座標よりバスケットの中心座標 (x, y) を算出する。また、そのアルゴリズムを図 10 に示す。

$$x = (x_a + x_b) / 2 \tag{9}$$

$$y = (y_a + y_b) / 2 \tag{10}$$

a) では、k 回目の (x, y) を (x_k, y_k) とすると、(11) で算出される θ_k, θ_{k-1} が共に ± 10 度以内になるまで逐次画像を取り込み、このアルゴリズムを繰り返す。 θ_k, θ_{k-1} とは、図 11 のように k 回目、k-1 回目におけるバスケットの進路変更の角度を示している。 θ_k, θ_{k-1} が共に ± 10 度以内の範囲とは、バスケットがほぼ直線に運動していることを意味し、その時がシューティングのタイミングとなる。

$$\theta_k = | \tan^{-1} (l_{y_{k-1}} / l_{x_{k-1}}) - \tan^{-1} (l_{y_k} / l_{x_k}) | \tag{11}$$

(ただし、 $l_{y_k} = y_k - y_{k-1}$ 、 $l_{x_k} = x_k - x_{k-1}$)

b) では、ボールをシューティング後、ボールがバスケットの近傍に来るまで待ち時間を稼ぎ、

タイミングよく画像を取り込み、バスケット座標を得る。
 これら a) b) のアルゴリズムを図12に示す。

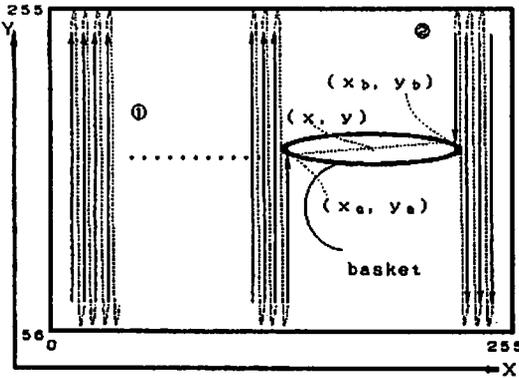


図9：画像処理の概略

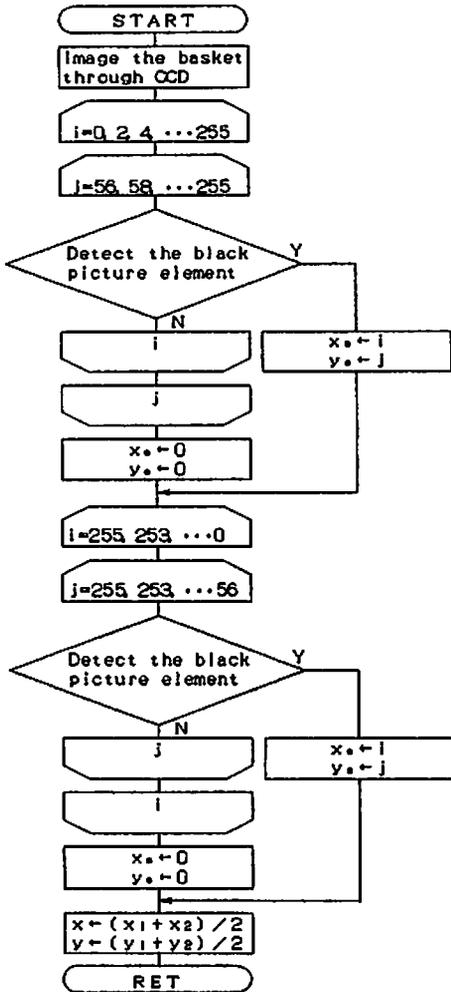


図10：バスケット中心座標
 検出アルゴリズム

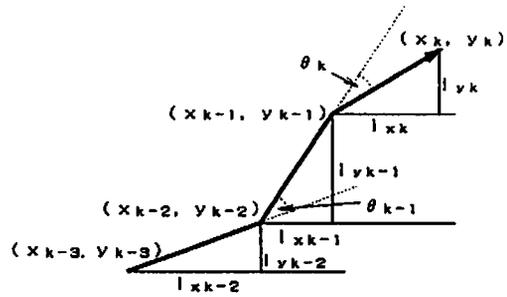
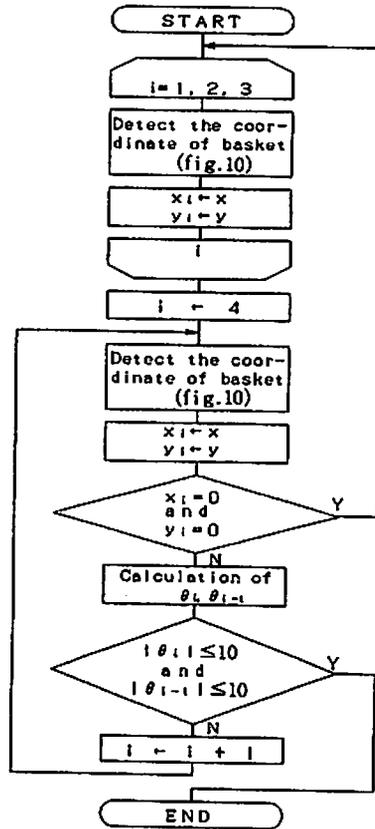
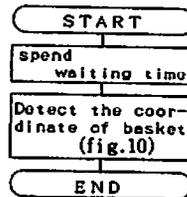


図11：バスケットの進路変更の様子



a)



b)

図12：実際の画像処理アルゴリズム

4.2 待ち時間の決定法

ここでは、ボールのシューティングが行われてからボールがバスケットの近傍に到達した時の画像を取り込むまでのタイミングの決め方について述べる。

バスケットは上下左右の不規則に運動するため、ロボットからバスケットまでの距離は浮動である。そのため、ボールがバスケット近傍に到達する前後に画像を取り込むことを防ぐため、待ち時間 t を次式のように線形式を用いて実験的に決定した。

$$t = a_1 \theta_1 + b_1 \quad (12)$$

ここで、 a_1 、 b_1 の値は最小2乗法によって決定した。ボールの動きは非常に速いが、バスケットはボールに比べてスピードが極端に遅いため、(12)の線形式で十分であり、精度もそれほど落ちることはない。

§5 ニューラルネットワーク制御部

ニューラルネットワーク制御部では、図13に示すようにバスケットの速度と加速度を入力とし、予測直前のバスケットの座標からの距離を出力とする、2入力1出力の3層ニューラルネットワーク(中間層6ユニット)を用いている。

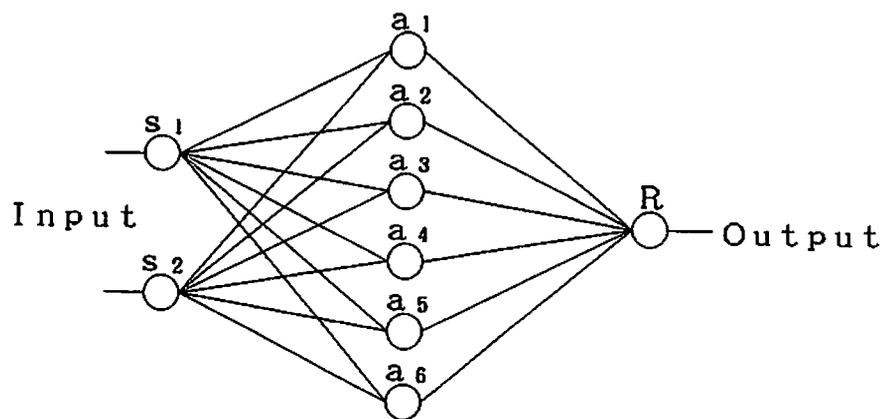


図13: 3層ニューラルネットワーク

実際には、学習の収束速度を向上させるため、 x 方向と y 方向の別、また、速度の正負の別、さらに加速度の正負の別に分割されており、計8個の小規模なニューラルネットワークが用意される。例えば、速度が $v_x > 0$ 、 $v_y < 0$ 、且つ、加速度が $a_x < 0$ 、 $a_y < 0$ であるなら、図14の太線で示される部分の2つのニューラルネットワークが使用される。図14における dx 、 dy は、

予測直前のバスケットの座標からの x 方向、y 方向の距離を示している。すなわち、 $d x = x_{n+1} - x_n$ で表現される。

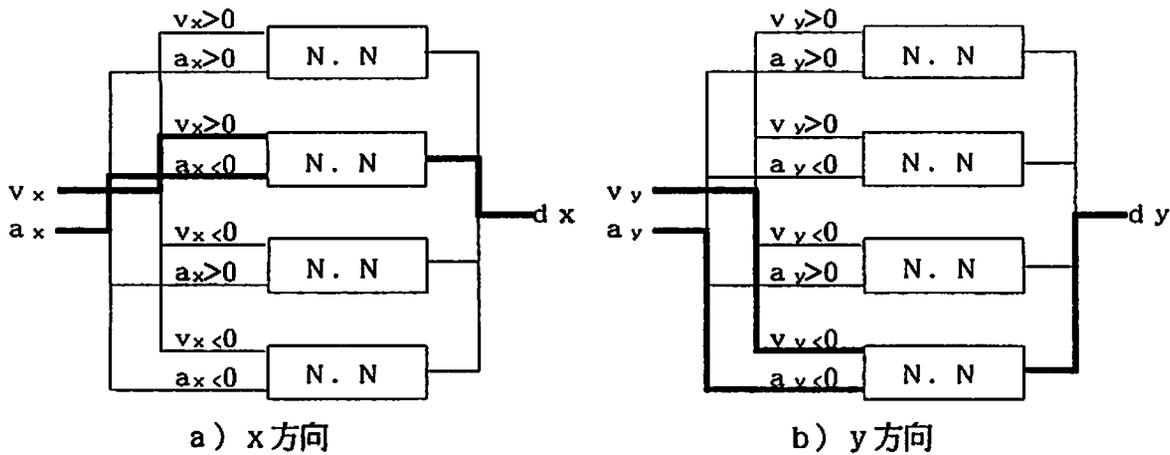


図14：制御部の内部構造

図13のニューラルネットワークの計算について述べる。

入力を x_1, x_2 、出力を y 、 x_i から a_j への結合荷重係数を w_{ji} 、 a_j のしきい値を $-h_j$ 、 a_j の出力を c_j で表すと、 c_j は (13) (14) により計算する。

$$s_j = \sum w_{ji} x_i + h_j \tag{13}$$

$$c_j = 1 / \{ 1 + \exp(-s_j) \} (= f(s_j)) \tag{14}$$

(s_j は a_j への入力 の 総和 を 示す。)

a_j から R への結合荷重係数を w_{rj} 、 R のしきい値を $-h_r$ とすると (14) 式で求められた c_j を用いて y は次のように計算する。

$$t = \sum w_{rj} c_j + h_r \tag{15}$$

$$y = f(t) \tag{16}$$

(t は R への入力 の 総和 を 示す。)

§ 6 遺伝的アルゴリズムによる学習部

遺伝的アルゴリズム (GA, genetic algorithm) とは、生物の進化をモデルにした計算理論であり、現代計算理論の難題とされる最適化問題を極めて高速に解くことが可能である⁴⁾。遺伝的アルゴリズムの基本的な手順は以下の通りである。

- 1) 遺伝子の配列をランダムに構成し、その遺伝子の集団を作る。これを遺伝子プールと呼ぶ。
- 2) 遺伝子プール中の各遺伝子の適合性を決定する。適合性とは、適用しようとする問題における遺伝子の優秀さである。
- 3) 適合性による期待値に応じて、親の遺伝子を選択し、その複製を得る。原則として、2つの遺伝子を選択する。
- 4) これらの遺伝子の間で交叉を行わせる。交叉とは適当に遺伝子を断片し、両方の遺伝子の断片を適宜に入れ替え、親の遺伝子と同じ長さの遺伝子を作ることである。
- 5) さらに、ある確率で突然変異を起こさせる。
- 6) 2)～5)を繰り返す。

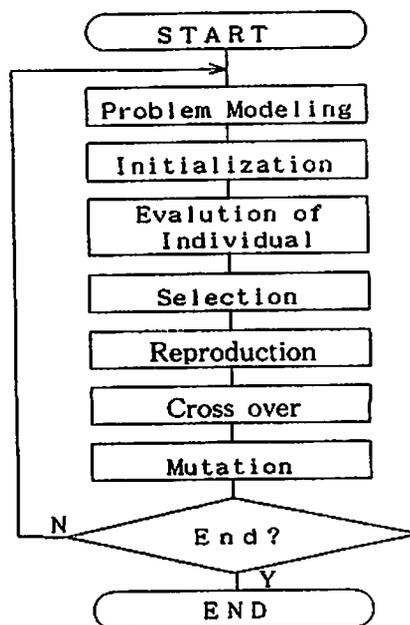


図15：遺伝的アルゴリズム

この遺伝的アルゴリズムをニューラルネットワークの結合荷重係数、しきい値の学習に用いた。

遺伝子型の表現としては、各結合荷重係数、しきい値を表1の4ビットのグレイ表現で表し、それを1列に並べたものを用いる。個体数は、50に固定し、最大30世代の計算を行った。以下に実際に行った遺伝的アルゴリズムの手順を示す。

- 1) 10の個体（全体の20%）に前回の最大適応度を持つ個体の遺伝子型をコピーする。残りの個体にランダムな値の遺伝子型をセットする。
- 2) 個体k (k=1, 2, 3, …, 50) の遺伝子型をニューラルネットワークを適用し、教師データi (i=1, 2, 3, …, 9) の入力パターンに対する出力値r_iを求める。教師データの出力パターンをt_iとすると個体kの適応度f_kは、

$$f_k = 1 / \{ \sum (t_i - r_i)^2 \} \quad (17)$$

表1：グレイ表現

-8	0 0 0 0	0	1 1 0 0
-7	0 0 0 1	1	1 1 0 1
-6	0 0 1 1	2	1 1 1 1
-5	0 0 1 0	3	1 1 1 0
-4	0 1 1 0	4	1 0 1 0
-3	0 1 1 1	5	1 0 1 1
-2	0 1 0 1	6	1 0 0 1
-1	0 1 0 0	7	1 0 0 0

と定義する。50の個体それぞれの適応度を計算する。

- 3) 50の個体の平均適応度Fを求め、適応度がFより低い個体に対して適応度がFより高い個体を任意に選びコピーする。
- 4) 50の個体からランダムにペアを作り、交叉確率より90%の割合で交叉を行う。交叉は結合荷重係数、しきい値のそれぞれに対して4ビットのランダムな位置で切断し、対応する部分を交換する一点交叉によって行った。

- 5) 各個体に対し、突然変異率10%の割合で、ランダムに選ばれた部分に対立遺伝子に置き換える。
- 6) 以上を1世代として、30世代になるまで2)～5)を繰り返して進化を起こさせる。
- 7) 最大適応度を持つ個体を使って教師データの修正を行い、さらにその個体を使って制御を行い、1)へ戻る。

なお、1)で前回の最大適応度を持つ個体の遺伝子型をコピーしているのは、収束を速めるためである。バックプロパゲーションにおいても同様の工夫をしているものが見られる。その場合は収束を速めることができるが、同時に局所解に陥り易くなる。その点遺伝的アルゴリズムでは、初期値を複数選択することができるため、大域的に探索を行うことが可能であり、バックプロパゲーションよりも局所解に陥りにくいと言われている。

教師データは、8個のニューラルネットワークそれぞれに対して9個ずつ用意する。学習を行うのは実際に制御に使用した2個のニューラルネットワークに対して行い、同時に教師データの修正も行う。以下に教師データの修正方法を述べる。また、図16に教師データの修正方法を示す。

- a) 制御器の初期の起動には、人間が経験により適当に作成した大まかな教師データを用いる。
- b) 1度目のシューティングが終えたとき、画像から1組の正確な入出力データを得る。
- c) 入力空間上でこの実データに最も近い教師データを取り除き、新たにそのデータを教師データに加えてニューラルネットワークに遺伝的アルゴリズムを用いて学習させる。
- d) 学習後、ネットワークに先ほど取り除いた教師データの入力データを入力して出力を計算し、その入出力パターンを教師データに加え、実データを取り除く。
- e) 再びアームロボットによるシューティングを行い、a)～d)を繰り返す。

以上のことを繰り返しニューラルネットワークを学習させると共に教師データも最適になるように修正していく。

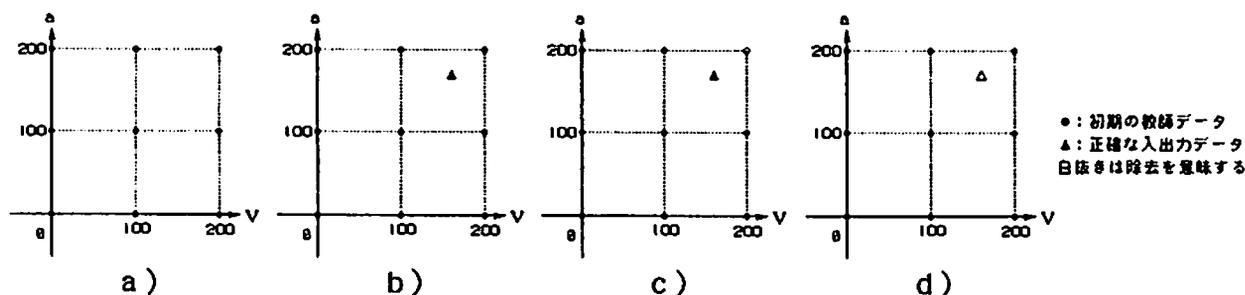


図16：教師データの修正方法

§ 7 実行結果と考察

本システムを用いてシューティング実験を行った。シューティングは240回行い、試行結果として命中率、修正命中率なるものを用いた。命中率とは、240回試行中のボールがバスケットに命中した回数の比率を示しており、修正命中率とは、命中はしなかったがバスケットにぶつかった回数を0.5倍したものの比率を命中率に含めたものを意味している。

$$p = q / n \tag{18}$$

$$p' = p + 0.5 \times (q' / n) \tag{19}$$

(n : 試行回数、q : 命中回数、q' : 修正命中回数)

また、学習の評価値として誤差距離 e_1 なるものを用いた。次時刻におけるバスケットの予測座標と実座標の差 (e_x, e_y) より e_1 は次式により求められる。前述のように $|e_x|, |e_y| \leq 20$ のもとでは学習は行なわれないため、次式より $e_1 \leq 28$ の時は、学習を行っていない。

$$e_1 = (e_x^2 + e_y^2)^{1/2} \tag{18}$$

試行回数に対する命中率、修正命中率のグラフを図17に示す。また、試行回数に対する誤差距離のグラフを図18に示す。グラフは、両者とも20回毎の平均値を折れ線グラフで示している。

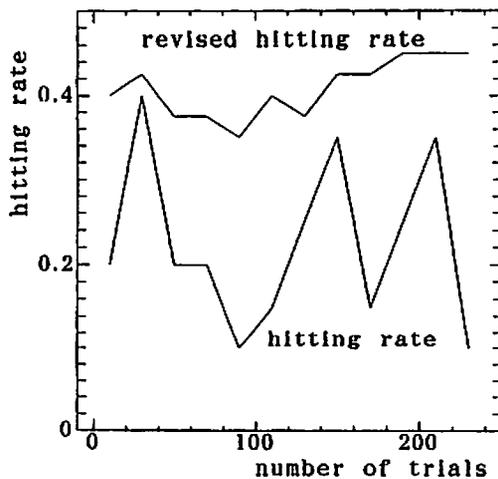


図17：命中率・修正命中率のグラフ

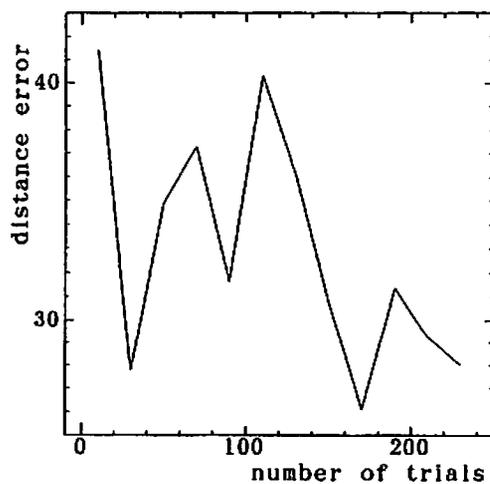


図18：誤差距離のグラフ

図17の命中率では、20回毎の平均ではあるが最高で40%を、総平均で23%を記録し、修正命中率では、半分以上が40%を超える結果となった。

実験中の学習の様子は、図18のように160回以前の試行では e_1 の値が28以上でばらつきが見られるが、160回以上では e_1 の値は28付近に停滞している。これは、シューティングを行う毎に徐々に制御部が学習され、学習を行う回数も段々と減っていることを示していると思われる。

それに反して、図17では命中率は160回以降もばらつきが見られる。この原因としては、今回の実験では、シューティングロボットの動作は(7) (8)式により固定で動作させたために発生するロボット動作時の偶然誤差、(7) (8)式におけるパラメータの未調整などが挙げられる。また、修正命中率は、160回以降は緩やかではあるが向上しているのが確認される。今回使用した画像処理では3値化が不可能であり、画像上のボールとバスケットを同時に認識することが困難であった。ボールを認識することが可能になればパラメータ調整等の負担を軽減し、ロボットはさらに精度良く動作するものと思われ、今回の命中率は修正命中率に近づき、そして、今回の修正命中率はさらに向上することが予想される。

今回のシステムでは、一般的に知られているバックプロパゲーション等の学習アルゴリズムは用いず、遺伝的アルゴリズムによりニューラルネットワークの学習を行った。バックプロパゲーションによる学習では学習の収束が初期状態に大きく左右され、初期状態が良い時はむしろ遺伝的アルゴリズムによる学習より速い場合があるが、初期状態が悪いときは収束が困難になる。それに対して遺伝的アルゴリズムでは複数の初期状態が探索を行うため、その内最良の初期状態を中心に学習が進められ、突然変異に起こすことによりその初期状態に片寄らない探索を行うことが知られている。従って、今回のシューティング実験においても常に安定して収束することが可能であり、収束が困難になるようなことは認められなかった。

また、従来のニューラルネットワークでは膨大なデータの採取を行い、それから多数の教師データを作成するが、本システムではデータ採取はまったく行わず、初期の段階では人間の経験だけにとよっている少数の任意のデータを教師データとして用い、その後はシューティングを行う毎にシステム自身が独自で学習を行いながら教師データも向上させていくので、オペレータの負担は大幅に軽減され、ほぼ全自動に近い学習制御システムを構築することができた。

§ 8 おわりに

過去の経験を近未来に反映させようとする人間の実時間的な学習処理を実現させる1例として、不規則に移動するバスケットにボールをシューティングする毎に学習を行い、徐々に能力を向上させる逐次学習型ロボットシステムを構築した。

その結果、命中率では最高で40%、総平均で23%を記録し、修正命中率では、半分以上が40%を超える結果となった。また、本システムでは、アームロボットが緩やかではあるが徐々にシューティング能力を向上させる様子が確認された。

今後は、ロボット動作部のパラメータの調整について検討していく方針である。

参考文献

- 1) 廣田, 箕輪: 推論部未調整なファジィ制御におけるニューラルネットを用いた目標計画決定, 第15回知能システムシンポジウム資料, pp.63/68, 1992
- 2) 堀川, 他: ニューラルネットワークによる学習型ファジィ制御器, 計測自動制御学会論文集, Vol.27, No.2, pp.208/215, 1991
- 3) 関口, 菅坂, 長田: 構想化ニューラルネットを用いた倒立振子の制御, ロボティクス・メカトロニクス講演会'92講演論文集, pp.857/862, 1992
- 4) D.E.Goldberg: Genetic Algorithm in Search, Optimization, and Machine Learning, Addison Wesley Publishing Company, 1989