パタン認識の制御への応用

統計数理研究所　石　黒　真木夫

(1976年12月　受付)

An Application of Pattern-Recognition to System Control.

Makio Ishiguro
(The Institute of Statistical Mathematics)

An adaptive control system called MAC (Multi-purpose Adaptive Controller) is described. It is so designed that it can control a large variety of plants including non-linear ones.

An example of controlled plant consisting of an automobile and a restricted course is shown and a result of the control experiment on this plant is given.

I. まえおき

この論文でいうパタン認識とは，与えられた一群のデータを一定の類に分類することである．このような分類が必要とされるのは，例えば，いくつかの症状によって病気の診断を下す場合である．この場合のデータは，“せきが出るが，リンパ腺ははれていない”，といった個々の症状の有無に関する一覧表である．このようなデータは，ある症状がある場合を1，ない場合を0で表現することにすれば，症状の数の長さを持つビットパタンとして表現することが出来る．他の形のデータもビットパタンとしての表現が可能である．例えば，数値の2進表現はとりもなおさずビットパタンであるし，紙に書かれた文字も適当なメッシュを切って各メッシュを文字の線が通っているかどうかを0と1で表現することによってビットパタンとして表わすことが出来る．このようなビットパタンを電子計算機などの機械を用いて，自動的に分類しようとする試みは，種々なされている．代表的な例として，ローゼンブラットのパーセプトロン[1]と，ウィドロウのアダライン[2]がある．例として，アダラインがいかにして『文字A』を認識するか見てみることにする．文字は，上述の方法で，ビットパタンとして表現されているものとする．ただアダラインの場合，計算の都合によってビットパタンは0と1でなく，−1と1を用いる．例えば，図-1に示される文字は，4×4のメッシュを用いて，長さ16のビットパタン（−1, 1, 1, −1, −1, 1, 1, −1, 1, 1, 1, 1, 1, −1, −1, 1）で表現される（4×4程度の荒いメッシュを用いたのでは，文字の形に関する細かい情報は表現されない．実際に文字を認識するためには，もっと細かいメッシュが必要である）．

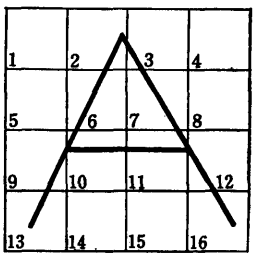


図-1　メッシュを用いた文字のビットパタン化

アダラインは，ビットパタン $\boldsymbol{p}$ ($=(p_1, p_2, \cdots, p_N)$) が与えられると，関数

$$d(\boldsymbol{p}) = \sum_{i=1}^{N} w_i p_i - \theta \qquad (1)$$

の値を計算する．そして $d(\boldsymbol{p})$ の値が正であるならば，このパタンを『文字A』と判定し，負であれば，これは『文字A』でないと判定する．もちろん，この結果が正しいためには，(1)

式の重みベクトル $\boldsymbol{w}$ $(=(w_1, w_2, \cdots, w_N))$ や閾値 θ の値はうまく選ばれている必要がある．他の文字，例えば，『文字 B』を読むためには，重みベクトルや閾値は，又別の値でなくてはならない．

(1) 式から明らかなように，p_i のいくつかの正負が反転してもそれが $d(\boldsymbol{p})$ の値の正負に影響しないかぎり，アダラインは同じ判定を与える．これは，ある程度くずれた字であってもアダラインが正しい判定を与える可能性のあることを意味している．

重みベクトルや閾値を適当な値に設定するには，『教育』が有効である．教育を行うにあたっては，『文字 A』のサンプルがいくつかと，『文字 A』でないもののサンプルがいくつか教育用のデータとして用意される．これらのデータに対するアダラインの判定の合否に応じて，重みベクトルや閾値を修正し，すべてに対して正解が得られるようにするのが『教育』である．この過程は，フローチャートを用いて図-2のように表わすことが出来る．

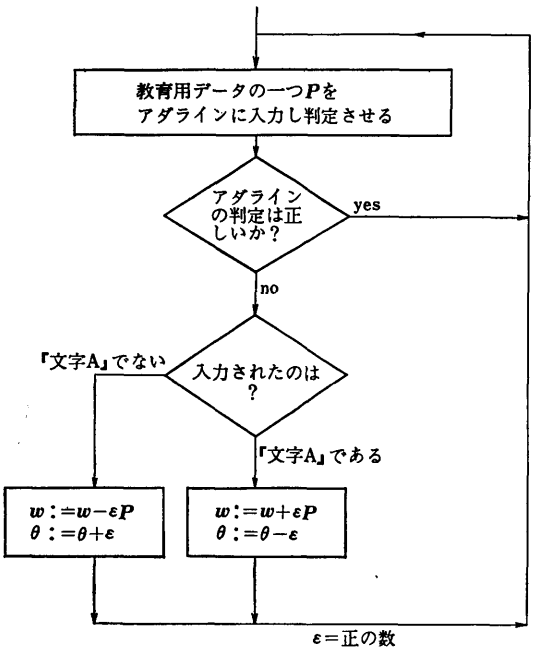


図-2 アダラインの学習

教育用データに関して，正しい判定を下す重みベクトルと閾値の組が一つでも存在すれば，図-2の過程は任意の初期値に対して，有限回で収束することが証明されている．この過程は，アダラインを使用する立場から見れば『教育』であるが，アダライン側から見れば，『学習』である．このように，正解が与えられていて，正解との一致や不一致に従って学習が行われるとき，これを教師つき学習という．

以上のようなパタン認識とその学習を，制御に応用することが可能である．例えば，室温がある一定の温度以上になったらクーラーのスイッチを入れ，その温度以下になったらスイッチを切るという場合は，室温をクーラーのスイッチを入れるべき範囲とそれ以外に分類することがそのまま制御につながる．一般に，K 個の異なる制御入力が可能な場合，制御の問題は，制御対象の状態を示す観測値を K 個の類に分類するパタン認識の問題に帰着される．

自動車のハンドルが一定の範囲内で任意の角度をとり得るように，連続量をとる制御入力の例が多い．そのような場合にも，適当な量子化によって有限個の制御入力だけを許すことにすれば，やはりこれもパタン認識の問題として扱うことが出来る．

パタン認識による制御の利点の一つは，パタン認識の学習をそのまま制御の学習に利用出来ることである．一般に自動制御を行う場合には，制御系のパラメータを適当な値に設定することが必要であるが，これが学習を用いることによって簡単に出来るからである．ウィドロウ [2] は，その例として，アダラインの拡張であるマダラインによる制御の学習について述べている．それは，図-3に示される倒立振子の安定化を目的とする制御である．

倒立振子は，直線上を動く台車に支点を持っている．台車は，スイッチの切換によって前進，

又は後退させることが出来る．台車をうまく動かしてやることによって振子が倒れないように保つことが出来るのである．マダラインには，振子の状態（振子の角度と角速度，及び台車の位置と速度）が一定時間間隔でサンプリングされて，ビットパタンに変換して入力される．学習期間中は，マダラインの出力は，『比較器』につながれていて，人間によるスイッチの切換と一致するかどうか比較される．振子の各状態に対する両者の出力が一致するようになるまで，図-2と同様の学習が行われる．ウィドロウによると，このような学習を数分間行った後は，マダラインによる振子の制御が可能であるという．

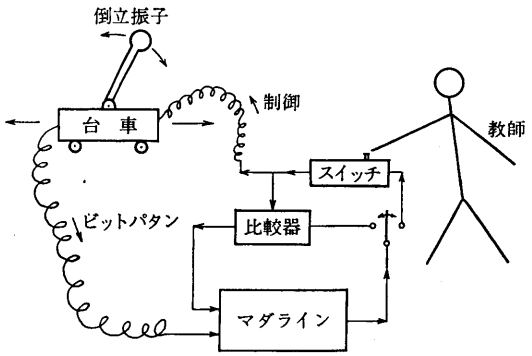


図-3　マダラインによる倒立振子制御の学習

これは，制御対象がある状態にあるときに，どのような行動をとるのが『正しい』のか，人間による制御が『正解』として与えられながら学習が進む教師付学習である．それでは，人間のまねをするという形ではなく，いわば制御を『独学』する機械あるいは，計算機プログラムは可能であろうか？　この問題に対する一つの答がゲインズとアンドレア [3] によって提案されている．彼らがステラと名付けた機械は，許される範囲内でいろいろな制御入力を試行制御として制御対象に与えて，その効果を試してみる．学習は制御対象のある状態（やはりビットパタンとして与えられる）に対して好ましい結果を与える入力が見つかるたびに，その状態と制御入力の対を記憶に加えるという形で進められる．試行錯誤的な学習法則ということが出来よう．

この論文は，筆者が開発したステラの改良版というべき，MAC (Multi-purpose Adaptive Controller) の構造とその振舞に関する報告である．ステラとの差は次の3点にある．

1. 学習を進めていく過程で，ある状態に対するある制御入力が好ましくないことが分る場合がある．MAC はこの時の制御対象の状態とその出力を記憶し，次に同じ状態が現れた時に，その出力を避けることが出来る．このような機能を持たないステラに比べて学習の早さの点で有利となる．これは又，学習期間中に起り得る損害を出来るかぎり小さくする働きもする．
2. MAC もステラと同様に自分自身の持つパラメータの値を修正しつつ学習するが，この修正法則はステラに比べてずっと単純である．
3. ステラは，一種の予測機能を持っており，各制御入力の結果を予測し，適当と判断される制御入力を試行制御として用いる．筆者の実験によると，この機能は計算時間と記憶容量を必要とする割に有効でない．むしろ有害な場合があり得る．MAC は試行制御をランダムに選び，構造がステラに比べて単純なものとなっている．

II. MAC

MAC が倒立振子の安定化を学習する場面を想定すると，振子と MAC の間の情報の流れは，図-4のようになる．

図-3と比較すると，マダラインと MAC の違いは明らかである．両者とも振子の状態を表現するビットパタンが入力されている点は同じであるが，図-3には，マダラインに『正しい制御』を示す人間と比較器が表れている．これに対して図-4には，『判定器』が置かれている．

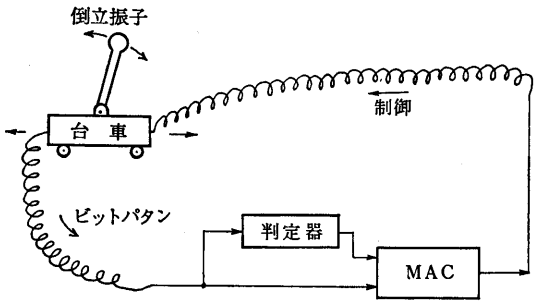


図-4 MACによる倒立振子の制御

判定器は，制御対象が望ましい状態にあるか否かの判定を MAC に伝える役割を持っている．この場合，振子が直立に近ければ 1，ある程度以上傾いていれば −1，その中間では 0 と判定する．MAC は判定を 1 に保つことを目標として，振子の各状態に対する適当な出力を次々に決定してゆく（これ以後，状態ということばは，それに対応するビットパタンの意味で使われる）．MAC の可能な出力は一つの整数値 (1, 2, ⋯, L) である．L は制御対象に応じて適当な値に設定しておく．

MAC の概要

ある対象を制御するのに不可欠な要素は，制御入力の各々に対するその対象の反応の仕方，つまり状態相互の間の遷移法則に関する知識である．MAC は対象の状態をクラスという概念でとらえる．クラスは，状態遷移に関して等価と考えられる状態の集合である．このクラスの間の遷移法則を表現するために MAC はポインタ及びポインタ値を用いる．状態をクラスとしてまとめてはいるものの，各クラスが制御入力の各々に対してどう反応するかをすべて記憶するのは容易でない．MAC は，各クラス毎に一つの制御入力（これは MAC から見れば出力であるからクラス出力という）を固定する．この制御入力によってどのクラスへの遷移が起り得るかを表現するのがポインタであり，ポインタ値はその程度を表現する．かなりの無理をしてメモリの節約を図っているわけであるが，シミュレーションの結果を見ると，これでも学習制御が可能なことが分る．なお制御を行うには，対象のどのような状態を望むのかが明らかでなければならないが，MAC は望ましい状態の集合を目標クラスという概念でとらえる．以下，クラス，ポインタ，目標クラスなどを順次説明してゆく．

クラス

制御対象の各状態に対して適当な出力を定めるといっても，異なるすべてのビットパタンを個別に扱ってその各々に一つずつ出力を決定するのは非能率的である（長さ K のビットパタンの数は 2^K である）．MAC は，ある状態 $\boldsymbol{p}$ に対する出力を決定するときに，$\boldsymbol{p}$ 及びその近傍から成る状態の集合を『クラス』として登録する．このクラスに属する状態に対しては $\boldsymbol{p}$ に対すると同じ出力が適当とみなすのである．この出力を『クラス出力』と名づける．$\boldsymbol{p}$ はこのクラスの『代表パタン』といわれる．$\boldsymbol{p}$ によって代表されるクラス，つまり $\boldsymbol{p}$ の近傍は，非負の実数を成分とする『重みベクトル』$\boldsymbol{w}$ $(=(w_1, w_2, \cdots))$ を用いて次の式で定義される．

$$\{\boldsymbol{p} \text{ の近傍}\} = \{\boldsymbol{q} \mid \sum_{i=1}^{K} \phi(p_i, q_i)\, w_i < 1\} \tag{2}$$

ここで K はビットパタンの長さである．p_i 及び q_i はそれぞれビットパタン $\boldsymbol{p}$, $\boldsymbol{q}$ の第 i 番目の成分である．ϕ は，

$$\phi(x, y) = \begin{cases} 0 & (x = y \text{ の場合}) \\ 1 & (x \neq y \text{ の場合}) \end{cases} \tag{3}$$

で定義される関数である．

(2) 式は，$\boldsymbol{p}$ と $\boldsymbol{q}$ の違い方を各ビット毎に異なる重みをつけて積算し，その違い方の小さい

ものを $\boldsymbol{p}$ の近傍と見なすことを意味している．各クラスごとに重みベクトルをうまくとることによって，クラス出力が適当とみなされる範囲をクラスとしてまとめることが出来る．重みベクトルに関する MAC の学習法則はそうなるように構成されている．このようにクラスの範囲は学習の過程で変形されるが，代表パタンとクラス出力は一度決定されると変更されない．ただし役にたたないクラスは削除されることがある．

ポインタ

制御対象の状態の各々は無関係ではない．制御入力を加えると，対象の特性に応じた一定の遷移法則に従って状態の遷移が起る．状態がビットパタンとして量子化されているため及び外乱などの影響によってこの状態の遷移は一般に確率的なものとなるが，ある制御入力によって特定の状態から遷移し得る行先はそれほど多くない．クラス相互の間にも同じことが言える．各クラスは，そのクラス出力によって遷移し得るクラスを記憶することが出来る．これを『ポインタ』という．各クラスは最大5個までポインタを持つことが出来る．各ポインタには，その示す先に遷移する確率にあたる『ポインタ値』が付随する（これは0と1の間の値をとる）．これによってどのクラスからどのクラスへの関係が近いか，その近さはどれ位かということが表現される．図-5 に示すようにクラスの集合は，ポインタで結ばれたグラフ構造を持つことになる．図中の目標クラスは後で説明される．

ポインタ及びポインタ値も学習によって構成される．

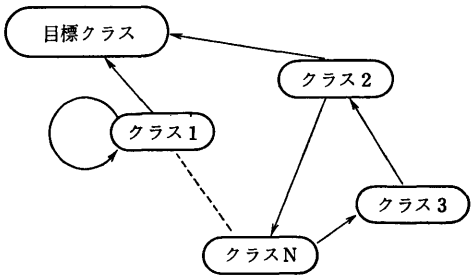


図-5　クラス及びポインタ

クラス値

制御対象の各状態は等価ではない．ある状態は判定器によって1と評価され，他のある状態は0あるいは−1と評価される．ある状態は適当な入力によって望ましい状態に遷移させることが出来るのに対して，他のある状態からは何段階かの遷移を経てはじめて望ましい状態に達するかもしれない．これに対応して各クラスにも評価が与えられている．これを『クラス値』という．クラスの一つに『目標クラス』と名づけられる特別なクラスがある．後述の作られ方で明らかなように，このクラスは判定器によって望ましいと規定される状態の集合に対応する．図-5 に見られるように，どのクラスもポインタによって直接的あるいは間接的に目標クラスと結ばれている（これは学習法則によって保証されている）．あるクラスから目標クラスに到る道筋にあるポインタのポインタ値の積をもってそのクラスのクラス値と定義する．ポインタ値は1以下の正の数であるから，目標クラスに致る道筋が長いほどこの値は小さくなる．道筋が複数個あってこの値がユニークに定まらない時がある．その時は，その最大値をクラス値とする．

補助クラス

I 節で触れたように，MAC はある状態に対して好ましくない制御入力を記憶することによって学習の能率をあげている．このためのメモリが『補助クラス』及びそれに付随する『禁止出力』である．補助クラスもクラスと同様に代表パタンと重みベクトルで規定される集合である．状態 $\boldsymbol{p}$ を含む補助クラスが1個以上ある時，付随する禁止出力の集合を『$\boldsymbol{p}$ に対する禁止出力リスト』という．

MAC の出力決定法則

MAC の出力決定方式をフローチャートの形にすると，図-6 のようになる．図中破線にかこまれた部分は，クラス選択手続として図-8 のフローチャートでも用いられる．

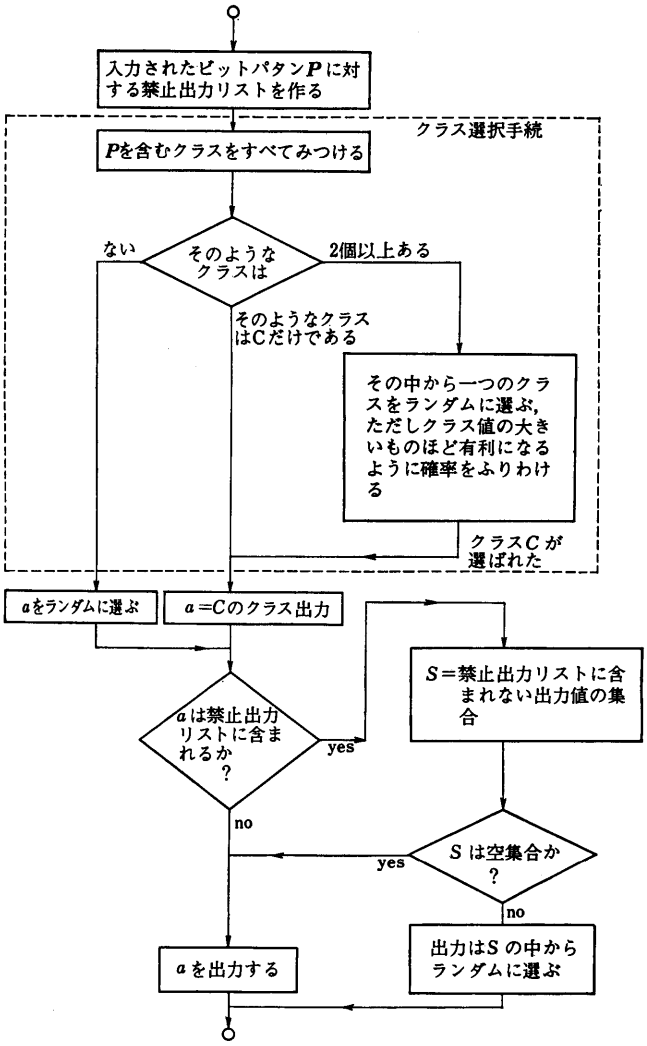


図-6 MAC の出力決定法則

目標クラス

学習の始まる時点での MAC は全くの白紙状態で，クラスなどを登録するためのテーブルはすべて空白である．ビットパタンが与えられても分類すべきクラスがないので，図-6 に明らかなように出力は確率的に選択される．判定器からの入力が 1 になった時に，はじめて一つのクラスが作られる．この最初に作られるクラスは特別なクラスで特に目標クラスと名づけられる．このクラスの代表パタンとしては，その時に入力されたビットパタンが採用される．重みベクトルの初期値としては各要素に 0.5 が与えられる．これは，代表パタンと 1 ケ所だけ異なる状態はこの目標クラスに含むが 2 ケ所以上異なる状態は含ませないことを意味する．このク

ラスは例外的にクラス値とクラス出力が定義されない．従って学習の場合は別として，出力決定にあたってはこのクラスは無視される．目標クラスに関する学習は図-7のフローチャートに示される．この部分は教師付学習の形になっている．

学習が進むと，判定器によって『良い』と判断される状態の集合と目標クラスがほとんど一致するようになる．

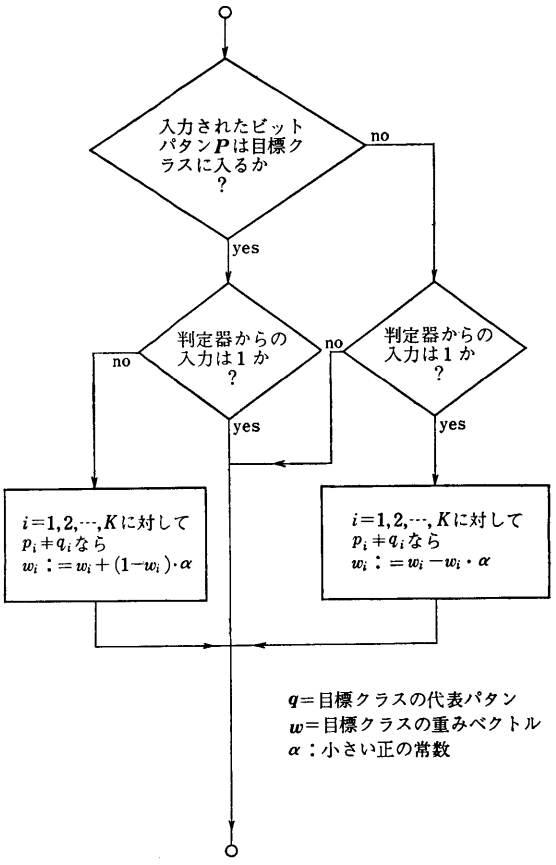


図-7　目標クラスに関する学習

クラスの登録

目標クラスが出来ると，これを基点として他のクラスを作ることが出来るようになる．新しいクラスが作られるのは，次の条件が満足された場合である．

『一周期前の状態 $\boldsymbol{p}$ はどのクラスにも属さなかったが $\boldsymbol{p}$ に対して出力 a を出した結果，クラス C に属する状態に遷移した』

この場合 $\boldsymbol{p}$ と a が新しいクラス C' の代表パタン及びクラス出力として登録される．なおこの時にクラス C がクラス C' のポインタとして登録される．C' の重みベクトルの各成分には初期値0.5が与えられる．ポインタ値の初期値も同様である．

ポインタの追加登録

クラスの構成のされ方，及びポインタの作られ方を考えると，クラス出力による状態遷移の結果はポインタの示すクラスに入るのが最もありそうなことである．しかし別のクラスに入ってしまう事がある．そのような時にMACはポインタをもう一つ追加登録する．ポインタは最大5つまで許される．既に5つのポインタが作られている場合には，ある程度以上小さいポインタ値を持つポインタがあれば，そのポインタが消されて登録がなされるが，そうでない場合は，登録はあきらめられる．2つのポインタが同じ値を持ちどちらかを消さなければならない時には，メモリー中の番地の若い方が消される．

重みベクトル及びポインタ値に関する学習法則

制御入力 a によって $\boldsymbol{q}\rightarrow\boldsymbol{p}$ の状態遷移が起ったものとする．この時 $\boldsymbol{q}$ がクラスDに含まれかつ a がDのクラス出力に等しければ，Dの重みベクトル及びポインタ値が修正される．フローチャートの形で表わせば図-8のようになる．

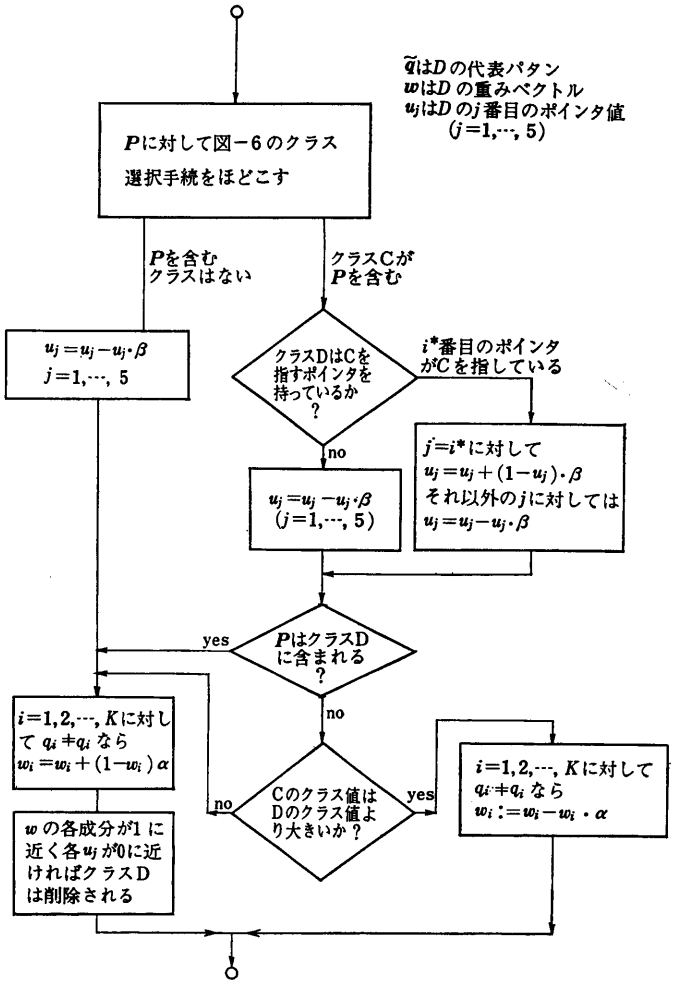


図-8 重みベクトル及びポインタ値に関する学習法則

補助クラスの登録

制御入力 a によって $p\to q$ の遷移が起ったとする．この時，q に対する判定が −1 であれば，p を代表パタンとする補助クラスが登録される．もちろんこのクラスに対する禁止出力は a である．q に対する判定が −1 でない場合にも同じ登録がなされる場合がある．それは，q を含む補助クラスがいくつもあって，すべての出力が禁止されている場合である．

補助クラスに関する学習

制御入力 a によって $p\to q$ の状態遷移が起った場合，図-9 のフローチャートに従ってそのクラスの重みベクトルが修正される．

III. MAC による制御学習の例

これまでの説明で明らかなように，MAC は制御対象が非線型な動作をするものであっても，制御することが可能である．以下は，非線形システムの制御の学習に MAC が成功した例であ

る．

MAC による自動車の制御

ここでとりあげるのは，非常に単純化された形ではあるが，自動車の制御の学習である．この自動車には，ハンドルがついていて，直進の他に，右又は左にカーブを切ることが出来る．ただし，ハンドルを切る角度は直進の他に右に 15° 左に 15° のいずれかに限られる．アクセルやブレーキの操作にあたる制御は考えずに，一定のスピードで走るものとする．つまり，この場合制御入力として可能な値は3値である．MAC の出力 1, 2 及び 3 がそれぞれ直進，右折，左折に対応し，自動車は，MAC の出力に応じて方向を変えその方向に一定の長さだけ前進する．この自動車は，図-10 に示されるようなコースに置かれている．自動車はこのコースの中では，内側又は外側の壁に触れないかぎり，MAC の出力に応じて自由に走ることが出来る．壁に衝突した時は特別で，ハンドルの操作に応じて，車の向きは変るが前進はしない．例えば図-10 の q 点がそうで，自動車はぶつかった点で左に向きを変えて行き止りの状態から脱出している．この図に描かれている矢印は，自動車の軌跡を示す．コースは右廻りにも左廻りにも廻ることが出来る．この例の場合の学習の目標は，壁に触れることなくコースに沿って自動車が走るようになることである．

この学習が完成するのは，要するに，図-11 の A, B, C の各々のような状態のそれぞれに対して MAC がハンドルを正しく，直進，左折，右折と切ることを覚えた場合である．それには，今自動車がAのような状態にあるのか，あるいはBかCなの

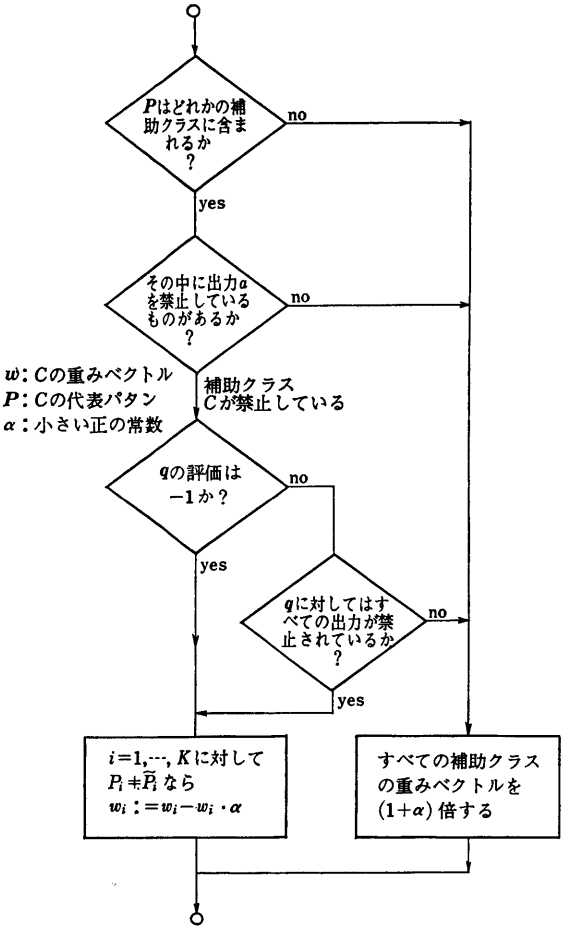


図-9　補助クラスに関する学習

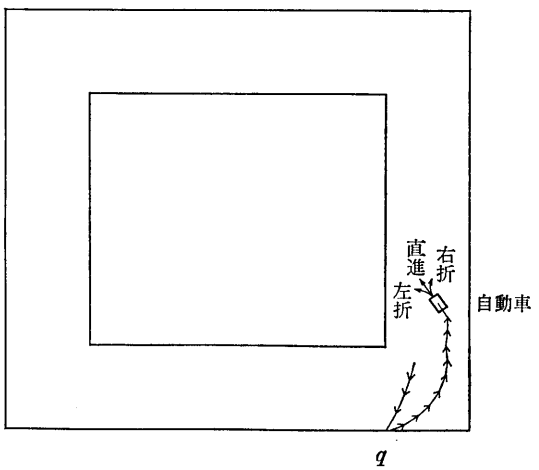


図-10　自動車とコース

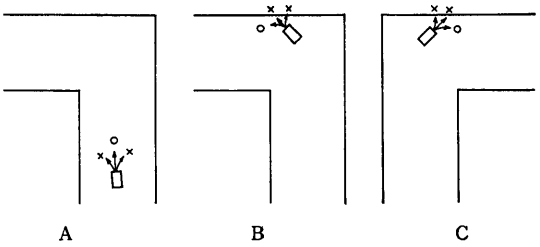


図-11 自動車の状態と正しい制御

かということがMACに与えられるパタンに表現されていなくてはならない．これは例えば図-12のように，コースに格子を切って，そのどの枡目に今自動車がいるかという情報と，その時の自動車の向きによって表現することが出来る．これとは別の表現方法を考えることも出来る

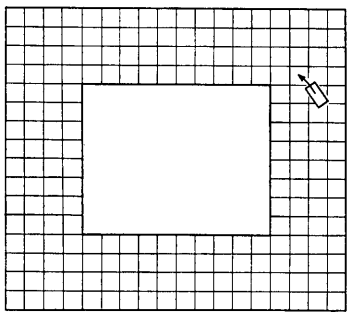


図-12 自動車の状態の表現 (I)

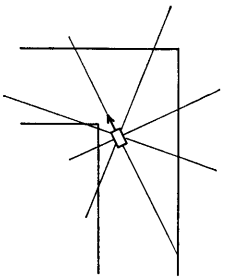


図-13 自動車の状態の表現 (II)

それは，図-13に示されるように，自動車の前後左右及び斜め方向それぞれどの位の距離の所に壁があるかを表現してやる方法である．これを前者の表現方法Iに対して表現方法IIということにする．表現方法Iが自動車のコースに対する大局的な位置を記述するのに対し，表現方法IIは，局所的な位置を記述するということが出来る．Iの方法を採用するとすると，図-14のAとBの場合は異なるパタンで表現されることになり，MACの学習が完成して，Aの場合にうまくコースを廻れるようになってもBの場合には適用できないことを意味する．これに対して，表現IIによって制御を学習したMACは，Aの場合の制御を学習すれば当然B

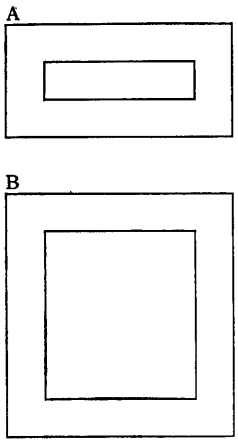


図-14 2つの異なるコース

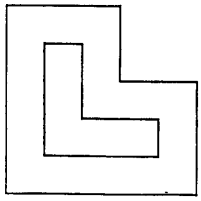


図-15 むずかしいコース

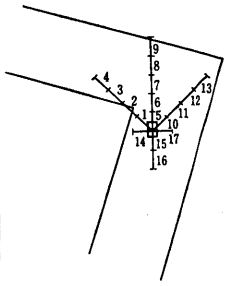


図-16 実験に用いられた自動車の状態表現

の場合にも制御することが出来るし，図-15 に示されるようなコースを与えられてもうまく自動車を運転できるはずである．実験は表現 II を用いて行われ，具体的には，図-16 に示される 17 本の線分（1から 17 まで番号づけられている）の各々が壁に触れているか否かを表わす，17 次元のベクトルによって自動車の状態が表現された．図の例の場合，自動車の状態は，パタン (0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0) で表現されることになる．

学習が行われるためには，判定器が必要である．この自動車の場合に用いられた判定器は，上記のように作られたパタンの 5，6，7 及び 8 番目の要表が 0 であれば 1，5 番目の要素が 1 であれば −1，それ以外の場合には 0 を出した．つまり，前方に余裕がある状態を良しとし，行止りの状態を不可とする判定を下すわけである．常に前方に余裕があるように制御をすることがすなわち，壁に衝突せずにコースを廻ることであるから，これは，制御の最終的な目標をうまく表現しているといえる．

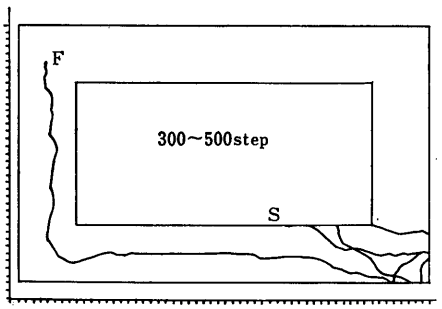


図-17　学習の初期

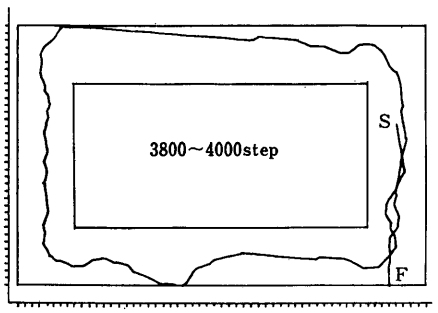


図-18　学習の中期

この系の制御を学習した結果が図の 17，18 及び 19 である．図-17 は学習の初期の段階の自動車の軌跡を示し，でたらめな制御を行い，その結果として，何度も壁に衝突している様子が見られる．図-18 は学習がほぼ完成した状態を示し，ほぼうまくコースを廻っているが，まだ何回か壁に触れている．図-19 では一度も壁に触れていない．この時 MAC の記憶しているクラスの代表パタン及び対応するクラス出力値の一部（この時 MAC は全部で 50 のクラスを構成している）を表にすれば表-1 のようになる．図-19 で最後にコースの角を曲る部分（ステップ 10450～10478）の MAC への入力パタン，その時の MAC によるパタンの分類及び出力を表-2 に示す．この間の軌跡を描けば図-20 のようになる．表-2 に示されるように，直進して来た自動車の前方に壁のあることが，ステップ 10452 で検出され右にカーブしはじめる（途中何回かは逆に左にハンドルを切ってはいるが）．ステップ 10450 に見られるように，入力パタンが完全に代表パタンと一致しなくてもそのクラス（この場合はクラス 7）に分類されている．クラス 7 に入るパタンか否かの分類に入力パタンの第 16 番目の要素は重要でないのである．もう一つ注意すべきことは，ステップ 10469 と 10470 に見られるように，同じ入力パタンであるにもかかわらず，異なる分類がされる場合があることである．この場合のクラスの選択は確率的になされている．このようなときには，その制御の結果によってどちらかの分類が有利になるよ

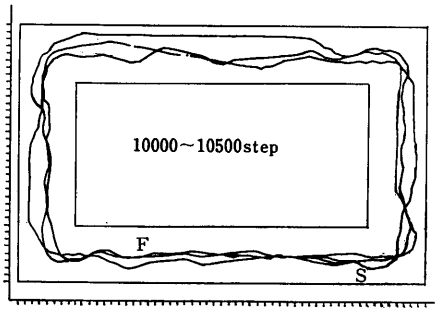


図-19　学習の完成

表-1

クラス	代表パタン				クラス出力値
2	0111	00000	0111	0000	1
7	0111	00000	0011	1010	1
12	0111	00001	0011	0000	2
19	0111	00011	0010	0000	2
20	0011	00011	0100	0000	3
23	0111	00111	0101	0000	1
31	0111	00111	0011	0000	2
33	0111	00011	0001	0000	3
34	0011	00111	0001	0000	3
35	0111	01111	0001	0000	2
36	0001	00011	0111	0000	3
37	0111	00111	0001	0010	2
43	0111	01111	0000	0000	2
46	1111	01111	0011	0000	2

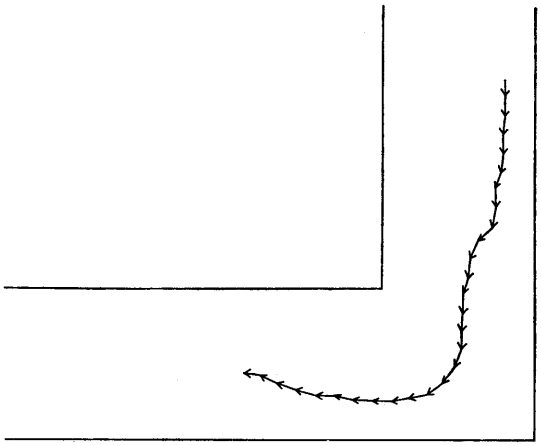
表-2

ステップ	入力パタン				分類	出力
10450	0111	00000	0011	1000	7	1
10451	0111	00000	0011	1000	2	1
10452	0111	00001	0011	1000	7	1
10453	0111	00001	0011	1000	7	1
10454	0111	00001	0010	1000	12	2
10455	0111	00001	0011	1010	7	1
10456	0111	00011	0011	0000	33	3
10457	0111	00011	0000	0000	19	2
10458	0111	00011	0010	0000	19	2
10459	0001	00011	0111	0010	36	3
10460	0011	00011	0100	0000	20	3
10461	0111	00111	0001	0000	37	2
10462	0011	00111	0000	0000	34	3
10463	0111	00111	0011	0000	31	2
10464	0011	01111	0001	0000	34	3
10465	0111	01111	0011	0000	46	2
10466	0111	01111	0011	0000	35	2
10467	0111	01111	0000	0000	43	2
10468	0111	01111	0000	0000	43	2
10469	0111	00111	0001	0000	23	1
10470	0111	00111	0001	0000	35	2
10471	0111	00000	0011	0000	7	1
10472	0111	00000	0011	0000	2	1
10473	0111	00000	0011	0000	7	1
10474	0111	00000	0011	0000	17	1
10475	0111	00000	0011	0000	12	2
10476	0011	00011	0111	0000	10	1
10477	0001	00011	0111	0000	13	3
10478	0111	00000	0111	0000	17	1

うに，さらに学習が進むわけである．

IV　ま　と　め

学習制御系とは，未知の対象に適応して適切な制御を行えるようになる制御系のことである．パタン認識の手法の一つの応用として，学習制御系 MAC を構成し，シミュレーションによる実験を行った．MAC にとって制御対象が線形であるか否かは無関係である．普通の線形制御の不可能な分野に適用できることが大きな利点である．前節の例は非線形システムの制御の一例である．又，この例では，自動車が直進して，コースの曲り角に達した時に，これをうまく廻るという制御が行われたわけであるが，これは，制御の過程で，通常とは異なるしかし何度か出会ったことのある情況（この場合はコースの曲り角.／）が現れ，これにうまく対処しなくてはならない場合にも MAC による制御が可能なことを示している．つまり制御対象がいくつか異なる動作モードを持っていて時々他のモードに移るという場合に対処できるということである．



10450ステップから10478ステップの
自動車の軌跡

図-20

一方 MAC の当面する困難としては，制御対象が複雑になった場合に非常に大きなメモリーが必要となることがあげられる．しかし，MAC における演算の大部分は，本質的にパラレル演算であるから，パラレル処理の機能を持つ計算機を用いることが出来れば，演算時間という面の困難は解決出来ると思われる．

参　考　文　献

[1] Rosenblatt, Frank "A Comparison of Several Perceptron Models", in Yovits, M.C., Jacobi, G.T., and Goldstein, G.D., eds. Self-Organizing Systems, Washington, D.C. Spartan Books, 1962.

[2] Widrow, Bernard "Pattern Recognition and Adaptive Control", IEEE, Appl. & Ind. (Sep, 1964), pp. 269–277

[3] Gains, B.R. and Andreae, J.H., Proc. 3rd IFAC Congress, LONDON, 1966.

訂　正

図-8　$q_i \mp q_i \rightarrow q_i \mp \bar{q}_i$　（二箇所）

図-9　$P:C$ の代表パタン → $\bar{P}:C$ の代表パタン