学 習 制 御 と 知 識[†]

菅 沼 義 昇*・伊 藤 正 美*

Learning Control and Knowledge

Yoshinori SUGANUMA* and Masami Ito*

This paper proposes a new method for knowledge acquisition, knowledge representation, generalization and reasoning in learning control. A learning controller to exploit an analogy between a current unsolved problem and a similar but previously solved problem to simplify its search for a solution is outlined. As an example, this controller is applied to control of multi-links systems and by simulation it is indicated that by using an analogy it can speed the search of input sequences to transfer initial states to goal states.

Key Words: machine learning, robot, learning control

1. はじめに

人間は、一つの問題について学習した後は、同じ問題はもちろんのこと、類似の問題も、その学習によって得られた知識を利用することによって、初めての場合一より容易に解決することができるようになる。これが可能であるためには、最初の問題を解決した際に得られた知識を、利用可能な状態で保存しておく必要がある。つまり、学習にとって、どのようにして知識を表現するのか、推論に利用できるようにどのようにして知識を一般化するのか、それらの知識を用いてどのようにして推論を行うのかなどが、核心的な問題となる。しかしながら、制御の分野において従来学習制御と呼ばれていたもののなかには、これらの点に対する考慮がほとんど払われておらず、単なる繰返し制御にすぎないものが多い^{1),2)}。他の学習制御に関連する研究としては、心理学・人工知能の分野における、人間の運動学習をモデル化しようとする立場からの研究が

存在する^{3),4)}。しかし、これらの結果も、実際のシステムに対する学習制御器の構成に直接利用するには、いまだ不十分な状態である。

筆者らはすでに、学習制御における知識表現について、一つの方法を提案した^{5),6)}。しかし、そこで採用した知識表現法などには問題点が多く残っており、十分な学習効果をあげることができなかった。現在明らかにになった主たる問題点は、以下に示すとおりである。

(1) 全体の知識構造が明確に記述されていない。

一般的知識と問題固有の知識がはっきりと分離されておらず、また、手続きのなかに埋め込まれている部分が多いため、プログラムの修正が非常に困難になっている。

(2) 知識を一般化処理した後も、かなりの数の知識を保存しなければならない。そのため、多くの記憶容量が必要になるとともに、知識の探索・一般化に多くの時間が費やされ、学習が進むにつれ問題を解決するために必要な時間が、かなりの速さで増加する。これは明らかに学習とは逆の現象である。

(3) 一般化された知識が、推論にとって有効な形になっていないとともに、推論方法にも問題がある。そのため、さきの経験によって得られた知識を有効に利用できず、十分な学習効果をあげることができない。また、ループに入ってしまう場合が多く発生し、特に、多入力多出力系に対しては、問題を解決するために膨大な時間がかかる。

そこで、本論文では、これらの問題点を考慮して修正された知識表現法、一般化手法、および推論方法などを提案する。

2. 学習と制御

機械自身に学習させるといっても、最初知識がまったくない状態では学習を行うことができない。学習分

[†] 第4回知識工学シンポジウムで発表(昭 61・3)

* 名古屋大学工学部 名古屋市千種区不老町

* Faculty of Engineering, Nagoya University, Nagoya

(Received October 31, 1985)

(Revised April 18, 1986)

野がいかなるものであっても、学習過程そのものを制御するメタレベルの知識が必要である。いったい、このすべての学習分野に共通する学習を可能にする知識とはどのようなものであろうか。たとえば、人間ではどのような知識として所有しているのか。非常に興味のある問題ではあるが、現在の段階ではそこまで一般的な問題は取り扱わない。ある対象を制御するという運動学習的な面において、制御対象がいかなるものであっても、それを学習的に制御するための知識に問題を限定する。もちろんこのときも、学習を行わせるためには、学習自身を制御する制御対象によらない一般的知識を前もって所有していなければならない。たとえば、ある範囲内では、線形的な内挿・外挿が有効である。速度を積分すると位置になるなどの知識は所有しており、学習の対象にはしない。本論文では、この一般的知識のもとで、各制御対象のシステムパラメータ（たとえば、リンクの長さ・質量などであるが、もし、あるリンク系の制御によって得られた知識を、他のリンク系の制御に利用しないならば、パラメータに対する知識は不用である）を知り、かつ、その入出力を観測することによって、試行錯誤を繰返しながら学習的に制御し、その過程で得られた制御対象（問題）固有の知識を記憶し、また、その知識を他の類似な問題にも利用できるように一般化する方法などについて述べる。

一般的知識および問題固有の知識の詳細について論ずる前に、本システム全体の制御戦略について述べておく。上に述べたように、このシステムで取り扱う制御対象はいかなるものであっても構わないが、説明の都合上、以下、制御対象として多入力多出力の力学系を念頭において説明する。制御の目的は、与えられた初期位置から目標位置へ、目標位置をオーバーシュートすることなしに、許容速度および許容入力範囲内で、なるべく早くもっていくこととする。問題は、この目的を達成するために必要な、入力（作用素）系列を求めることになる。しかし、このようなマクロ作用素を直接求めることは難しいため、一つの作用素を適用することによって達成できる副目標の集合に分解する。ただし、ここで決められる副目標は、現在所有する知識を基に推論した結果であり、知識が少ない状態では、すべての副目標が達成されたとしても、最終目標が達成できるとは限らない。そのようなときは、最終目標を達成できなかったときの情報を用いて、副目標の一部を変更し、再び最終目標を目指して試行を再開する。実際の副目標としては、制御周期 H 時間ごとに、目標とすべき速度として与えられる。そして、現

在の速度をこの副目標として与えられた速度に変換可能な作用素（入力）を、一般のおよび問題固有の知識を利用して、推論していくことになる。このことは、各時刻において、何回かの試行錯誤を繰返すことになり、もし、副目標を満足していなければ、時間を H だけ戻し、再試行を行うことを意味している。しかし、実際の制御においては、これは必ずしも可能ではない。このような場合においても、ある程度プログラムを修正することにより、基本的には本論文で述べる方法によって学習制御は可能であると思われる。なぜなら、制御対象に対する知識が、どのような順序であれ、得られればよいからである。しかし、希望する状態に戻すことの困難度、および、その状態で許される試行錯誤の範囲（たとえば、ある入力を入れてその出力を観測したいが、それが許されるか否かなどのこと）の減少に従って、学習に必要な時間は増加していくものと思われる。

3. 問題固有の知識

一般に、未知の制御対象を制御しようとしたとき、副目標を達成できる入力を、1 回で推論できるとは限らない。現在の知識を用いて推論した入力を対象に対して与え、その出力を観測し、そのデータを基にさらにつぎの入力を推論するということを、何回か繰返す必要がある。この過程で得られる知識、つまり、ある入力を与えると出力がどのようになるかを、そのままの形で表現したものが、この $K1$ と呼ぶ、最も基本的な知識である。制御対象の現在の位置を X 、速度を V 、入力を U 、 H 時間後の速度を VH とすると、つぎのように表わされる。

$$K1) (X, V, U) \rightarrow VH \quad (1)$$

これは、位置 X 、速度 V のもとで、入力 U を入れると速度が VH になることを表わしており、一つの問題を解決する際に得られたほとんどすべての知識を含む。

出力を観測して入力を変更するといっても、多入力多出力の場合、どの入力がどの出力に最も影響を与えるのか、また、どの程度入力を変化させれば出力に影響が現れるのかなどを、前もって知る必要がある。そこで、各時刻における最初の試行錯誤として、各入力を単独に微量だけ変化させ、その出力を観測する。そのとき、出力に十分な変化が現れない場合は、入力値を少しだけ増加させ、同じことを繰返す。この試行錯誤により、上に述べたような関係を得ることができ、それを、知識 $K2$ として保存する。位置を Xa 、速度を Va とし、入出力関係を Ra 、微小入力を DUa と

すると、知識 $K2$ は、つぎのように二つの知識として表わされる。

$$K2) (Xa, Va) \rightarrow Ra \quad (2)$$

$$K2) (Xa, Va) \rightarrow DUa \quad (3)$$

入出力関係とは、 H 時間後の各速度に、何番目の入力 が最も強く、また、速度の増加または減少のどちらの 方向に、影響を与えるかを表わし、微小入力とは、 H 時間後の速度に、0 入力のとき比較して、ある値以上 の差を生じさせる入力値を表わしている。

では、これらの知識を用いてつぎに与えるべき入力 をどのようにして推論していけばよいであろうか。知識 $K1$ が存在せず、知識 $K2$ だけのときは、修正したい出力に最も影響を与える入力を、出力が改善されると 思われる方向（上に述べた入出力関係によって知ること ができる）に、やはり知識 $K2$ によって与えられた微量 量だけ変化させる。このようなことを繰返すことによって、知識 $K1$ が増加し、以下に述べる推論方法を利用 できるようになる。知識 $K1$ を用いた推論方法を説明す るために、ここでは 2 入力 2 出力系を例にとる。いま、修正したい 2 番目の速度に最も影響を与えるのは、1 番目の入力であるとする。推論を行いやすいように、実際の知識 $K1$ は (1) の形と多少異なった形で保存されている。現在の位置 X ・速度 V ・2 番目の入力が $U21$ のとき、1 番目の入力が $U11$ 、 $U12$ のおのおの場合に対する結果 ($VH11$, $VH21$), ($VH12$, $VH22$) が得られているものとする。このとき、知識 $K1$ は、つぎのように表わされる。

$$(X, V, U21, U11) \rightarrow (VH11, VH21)$$

$$U12) \rightarrow (VH12, VH22)$$

このように、知識 $K1$ は、位置、速度、および、他の すべての入力を一定とし、ある一つの入力だけを変化 させた結果をまとめて保存している。そのため、同じ 情報を二重に保有するような場合が発生する。たとえば、1 番目の入力が $U11$ 、2 番目の入力が $U21$ のときの結果は、上に述べたような形で保存されていると ともに、1 番目の入力を一定 ($U11$) として、2 番目の入力を $U21$ 以外の値 $U22$ に変化させた結果 ($VH1$, $VH2$) が存在するならば、つぎのような知識も保存 されている。

$$(X, V, U11, U21) \rightarrow (VH11, VH21)$$

$$U22) \rightarrow (VH1, VH2)$$

このような知識が保存されていれば、2 番目の速度 $VH2$ をある値に一致させるためには、1 番目の入力 $U1$ をいくつにすればよいかは、線形内挿・外挿を用いて容易に推論できる。ただし、 $U1$ を変化させれば、1 番目の出力 $VH1$ も変化する。 $VH1$ も副目標

に近づくように変化するときは問題ないが、その逆のときは、得られた $U1$ をそのまま推論値として採用するわけにはいかない。 $VH1$ のほうが $VH2$ より早く副目標から遠ざかるようなときは推論できないことになる。そうでなければ、 $VH1$ の変化が、ある値以内に収まる範囲で $U1$ を修正するものとする。

以上述べた知識のままでは、システムパラメータ・位置などが、経験した値に完全に一致しない限り、この知識を利用することができない。そのため、一つの問題を解決することによって得られた知識は、その問題に限って有効となり、各問題を解決することにすべての知識を保存しておかなければならず、必要な記憶容量は膨大なものとなる。しかし、これらの知識のなかには、他の類似な問題にも利用できる共通な知識が含まれているはずである。また、上に述べた推論方法では、特に多入力多出力系において、その入出力間の干渉が強い場合は、有効な推論を行うことができず、ループに入ってしまうようなことが発生し、非常に長い計算時間が必要になる。そこで、得られた知識を、他の問題でも利用でき、知識の探索に要する時間を少なくするため、得られた情報をなるべく失わない範囲で少ない数に、かつ、後に述べる推論方法に有効に利用できるよう形に、一般化する必要がある。

まず最初に、知識 $K2$ の一般化について述べる。それは、条件部の値を範囲に変更し、かつ、結果のあいまいさを増加させることによって行われる。この一般化された結果を知識 $K3$ とし、この時点で知識 $K2$ は削除される。いま、同じシステムパラメータ Pa のもとで、(3) の知識に加え、つぎのような知識が得られたとする。

$$(Xa, Vb) \rightarrow DUb$$

DUa と DUb との差はある値 (G) 以下であったとする。また、システムパラメータ・位置がこの知識と一致し、速度が Va と Vb との間に入り、かつ、別の結果 (DUa と G 以上の差がある結果) が得られるような知識は存在しないものとする。このとき、つぎのように一般化される。

$$K3) (Pa, Xa, Va - Vb) \rightarrow DUa \quad (4)$$

この知識は、システムパラメータが Pa 、位置が Xa 、速度が Va から Vb の間に入ったとき、微小入力は DUa となるということを示している。ここで、記号「-」は、必ずしもベクトル的大小関係を表わしていない。各要素ごとに、対応する二つの値の間に入っていることを示す。この一般化は、システムパラメータなどに対しても行われ、最も一般化された形としてはつぎのようになる。

$$K3) (Pa - Pb, Xa - Xb, Va - Vb) \rightarrow DUa$$

(5)

G の値の決め方は、この一般化の方法にとって本質的なことであり、かなり難しい問題である。 G を大きくすれば、広い区間に対して一般化されるため、知識 $K3$ として保存すべき数が減少し、かつ、知識 $K3$ が利用される頻度が増加するが、推論誤差が大きくなる。逆に、 G を小さくすると、知識 $K3$ の数が増加し、知識 $K3$ が利用される頻度は減少するが、推論誤差は小さくなる。この現象は、一見矛盾しているように感じられるが、極端な場合を考えてみれば当然なことである。たとえば、 $G=0$ の場合を考えてみる。このとき、一般化はほとんど行われず、知識 $K3$ は(3)の形のまま保存されることになる。そのため、知識 $K3$ の数は増加するが、条件部の値に完全に一致しない限りこの知識は利用されないため、利用回数は減少する。しかし、利用されたときは、正確な推論値を与えることになる。学習効果をあげるためには、一般化された知識をより頻繁に利用し、知識がない状態での処理の回数を減らす必要がある。しかし、これらの知識を用いた推論誤差があまり大きければ、かえって、学習効果を減少させることになりかねない。一般に、 G の値は、制御目的、特に、許容誤差などに大きく依存する。この G 自身を学習させることも、基本的には可能であるが、現在の段階では、上に述べたような点を十分考慮して、本システムのユーザが試行錯誤的に決めてやることになる。

このような一般化を行うとき、一つの大きな問題が発生する。それは、一般化された知識が必ずしも正しくないことである。そのようなときは、つぎのようにして、知識の修正を行う。いま、(5)の形の知識が存在するとき、システムパラメータ、位置、および、速度が、おのおの(5)の条件部に与えられた範囲に入るが、結果 DUc が、 DUa と G 以上の差があるような知識が得られたとする。このとき、知識(5)は、つぎのような二つの知識に分解される。

$$(Pa, Xa, Va) \rightarrow DUa$$

$$(Pb, Xb, Vb) \rightarrow DUb$$

この修正により、知識の一部を失う場合が発生するが、さらに学習を行うことにより、新しい知識によって補われることになる。

つぎに、主として知識 $K1$ から得られる知識の一般化について述べる。まず、それをどのような形で保存すべきかを明らかにするために、一般化された知識を用いた推論方法から述べていく。再び、2入力2出力系を例にとって説明する。いま、他の入力を固定して

一つの入力だけを変化させた場合、入力と H 時間後の速度は、ある範囲で、近似的に線形的な関係があるものとすると、それらの関係はつぎのように表わされる。

入力 1 だけを変化させたときの速度 1 :

$$A11 * U1 + B11$$

入力 1 だけを変化させたときの速度 2 :

$$A21 * U1 + B21$$

入力 2 だけを変化させたときの速度 1 :

$$A12 * U2 + B12$$

入力 2 だけを変化させたときの速度 2 :

$$A22 * U2 + B22$$

A_{ij} , B_{ij} が既知であれば、 H 時間後の速度を ($V1$, $V2$) は、知識 $K1$ より容易に求めることができ、また、これにするための入力 (X , Y) は、つぎの式を解くことによって得られる。

$$A11 * X + B11 + A12 * Y + B12 = V1$$

$$A21 * X + B21 + A22 * Y + B22 = V2 \quad (6)$$

したがって、知識としては、各状態における A_{ij} , B_{ij} を所有していればよいことになる。しかし、知識 $K3$ によって採用した一般化の方法を用いても、 B_{ij} の値はそのときの状態により本質的に変化するものであるため、有効な一般化を行うことができない。そこで、まず、 A_{ij} に対する知識だけを考える。 i 番目の入力の変化に対する各速度の変化の割合(傾き)を A_i とすると、この知識はつぎのように表わされる。

$$K4) (Pa, Xa, Va) \rightarrow A_i \quad (7)$$

この知識に対する一般化および修正は、知識 $K3$ とまったく同じ方法で行われる。しかし、この知識だけでは、目標とする速度を達成するための入力を推論することができない。いま、入力 ($UE1$, $UE2$) を入れたときの速度 ($VE1$, $VE2$) が、既知であるものとする。このとき、つぎの関係が得られる。

$$A11 * UE1 + B11 + A12 * UE2 + B12 = VE1$$

$$A21 * UE1 + B21 + A22 * UE2 + B22 = VE2$$

(8)

このときは、(6)式と、(8)式を用いれば、(X , Y) を求めることができる。そこで、すべての入力を 0 としたときの、 H 時間後の速度と現在の速度との差を DV とし、つぎの形で保存する。

$$K4) (Pa, Xa, Va) \rightarrow DV \quad (9)$$

また、この知識の一般化および修正も、知識 $K3$ と同じ方法で行うものとする。このようにすると、知識(9)を用いて、 VE (つまり B_{ij}) の推論値を得ることができ、その値と知識(7)および(6)式から、副目標を達成するために必要な入力の推論値 (X , Y) を計

算することができる。さらに、ある時刻における2回目以降の推論は、**UE** および **VE** の値がすでにわかっているため、知識 **K4** を用いることなしに、入力をより正確に推論することができる。もちろん、上に述べた一般化された知識が得られた後に、知識 **K1** は削除される。

最後に、一つの問題を解決したときの最終結果を表現した知識について述べる。この知識は、与えられた初期位置から目標値へ到達するための入力系列および各時刻における位置・速度の情報を含んでいる。各制御時点における位置を X_i 、速度を V_i 、入力を U_i とし、目標値を OX とすると、この知識はつぎのように表わされる。

$$K5) (Pa) \rightarrow (X1, V1, U1) \rightarrow \dots \rightarrow (X_i, V_i, U_i) \rightarrow \dots \rightarrow OX \quad (10)$$

現在のところ、この知識に対しては一般化がなされず、このままの形で保存される。

4. 制御における一般的知識

前章で述べた知識は、多くの問題に共通する部分をもってはいいても、基本的には各問題固有の知識であり、本システムにおいて、学習における知識獲得の対象となる知識である。この章では、前章で述べた問題固有の知識の獲得過程を制御し、さらに得られた知識を用いて実際の対象を制御していくために必要な一般的知識について述べる。一般的といっても、制御問題に限定しているため、すべての制御対象に共通する一般的知識、たとえば、現在速度が正であれば、位置に対応する値は増加するなどに関する知識はすべてに所有しているものとする。もちろん、このように限定しても、いかなる知識を、どのような形で所有すべきかを決めることは、非常に困難な問題である。本システムでは、これら全体の流れを制御し、推論を行うための知識は、47個の知識から構成されており、その主なものは以下に述べるとおりである。ただし、ここで、**K1**, **K2** などと表現された知識は、前章で述べた学習によって得られた問題固有の知識である。

M1) もし、知識 **K5** によって、現在の位置から最終目標位置へもっていくことができる入力系列を推論できるならば、以後の入力としてその値を採用する（マクロ作用素の適用）。

M2) もし、知識 **K4** を用いて、 H 時間後の速度が推論できるならば、つぎの制御時点までに、最大どれだけ減速できるかを推論する。

—この値で減速していても、最終目標値をオーバーシュートしてしまうと思われるときは、つ

ぎの制御時点に対する副目標として、現在の速度と逆方向の最大速度を与える。

—速度の変化率は一定として、つぎの制御時点で最終目標値に到達するような速度を、現在の速度を基に推論する（ただし、上限は、最大許容速度および最大許容量入力で達成できると思われる値によって押さえられる）。もし、この速度が達成できたとして、以後減速すれば、最終目標値をオーバーシュートしないと思われるときは、この値を副目標として設定する。そうでなければ、上の条件を満足するように、副目標を調整する。

M3) もし、知識 **K4** を用いて、 H 時間後の速度が推論できないならば、速度の変化率は一定として、つぎの制御時点で最終目標値に到達するような速度を現在の速度を基に推論し、その値をつぎの制御時点に対する副目標とする（ただし、上限は最大許容速度で押さえられる）。

M4) もし、最終目標値をオーバーシュートしてしまったときは、時間を H だけ戻し、その時間以降の副目標を修正する（たとえば、速度が大きすぎるときは、 $1/2$ にする）。この修正が不可能なときは、時間をさらに H だけ戻し、副目標の修正を行う。

M5) もし、最終目標に達していれば、得られた **K1**, **K2** などの知識を一般化して、**K3**, **K4**, **K5** などの知識として保存し、終了する。

M6) もし、副目標を達成していれば、時間を H だけ進め、そうでなければ、入力を推論しなおす。この現在の入力を与えることによって得られた結果は知識 **K1** として保存される。

M7) もし、知識 **K2** または **K3** によって、入出力間の基本的な関係を推論できるときは、その推論を行う。この基本的関係とは、以下に示すものである。

— H 時間後の各出力（速度）にどの入力がか最も大きな影響を与えるか

—またそれは、速度の増加または減少のどちらの方向に与えるのか

—どの程度入力を変化させると速度にある値以上の変化が現れるのか

M8) もし、知識 **K2** または **K3** によって、入出力間の基本的な関係を推論できないときは、各入力を、単独に、微小量だけ順に変化させ、それらの関係を得、知識 **K2** として保存する。

M9) もし、知識 **K4** により、 H 時間後の速度が推

論できるならば、副目標を達成するために必要な入力を推論する。

M10) もし、副目標を達成していないある速度に最も影響を与える入力 ui の変化に対応する知識が、知識 $K1$ の中に存在するときは、その速度が副目標に近づくように ui を調整する。

M11) もし、今回の結果が前回の結果より改善されたときは、入力をその方向に、副目標になるべく近づくように変化させる。この推論は、以下に述べるようにして行われる。いま、前回の入力を $U1$ 、速度を $V1$ 、今回の入力を $U2$ 、速度を $V2$ とし、副目標を VO としたとき、まず、つぎの値を最小にする X を決める (ただし、速度がベクトルのときは、各要素に対する和をとったものになる)。

$$F = \{V1 + \alpha * (V2 - V1) - VO\} * * 2 \quad (11)$$

このとき、新しい入力に対する推論値 U は、つぎの式によって求めることができる。

$$U = U1 + \alpha * (U2 - U1) \quad (12)$$

M12) もし、入力を推論するために必要な問題固有の知識が存在しないときは、修正したい速度に最も影響を与える入力を、副目標に近づくと思われる方向に微小量だけ変化させる。

これらの知識はすべて、条件・処理対の、一種のプロダクションシステムの形をしている。実際の形は、個個の条件を数値により抽象化し、それらの AND 結合された条件部分と、その条件が満足されたとき行うべき手続きの名前を、ポインタで結合したリスト構造をしており、実際に学習を実行するプログラムとは独立なプログラムによって、会話的に作成される。学習を実行するプログラムは、記憶装置に保存されたこの知識を用いて、現在の状態がこの知識のどの条件部に一致するかを判断し、その処理部に示された手続きを起動することになる。

上に述べた知識は、2 種類の知識からなっている。一つは、学習過程自身を制御するための知識である。たとえば、 $M4, M5, M6$ などがこれに相当し、また、ここにあげていない知識のほとんどがこの種の知識で

ある。この知識は、目標を達成したか否かを判断したり、時間を進めたり、プログラムの実行を制御するためのものである。他の一つは、得られた知識をもとに推論を行うためのものである。得られた知識の量により、適用できる推論方法が異なるため、いくつかの知識からなっているが、大きく分けて三つのタイプの知識に分類される。まず第一のタイプは、問題固有の知識がまったくないか、または、非常に少ない場合に利用される知識である。たとえば、知識 $M3, M8, M10, M11, M12$ などがこのなかに含まれる。このタイプの知識は、経験のないときは非常に多く用いられるが、多くの問題を解決していくに従い、急速に利用回数が減少していくべき知識である。つぎのタイプは、知識 $M2, M7, M9$ などによって表わされる知識である。このタイプの知識は、最初は利用されず、ある程度経験を得ることによって徐々に利用され、学習が完了するに従い、再び利用回数が減少していくべき性質をもっている。なぜなら、学習の完了するに従い、タイプ 3 の知識 $M1$ の利用頻度が増し、与えられた目標を、ほとんど考えることなしに達成できるようにならなければならないからである。まさに、この時点が、熟練したといえる時期である。次章では、今までに述べたシステムが、十分な学習効果をあげることができるか否かを、多リンク系の制御を例にとり調べてみる。

5. シミュレーション

今まで述べたシステムを用いて、Fig. 1 に示すような 2 リンク系の制御を行ってみる。本論文の学習制御システムは、リンク系だけを対象としたものではないが、以下に述べるように、リンク系を制御することは、その非線形性および入出力間の干渉などの点から、学習制御対象として非常に困難なものとなるため、この例を採用した。各リンクの長さは $0.1m$ 、質

Fig. 1 Object system

Fig. 2 An example of control

Fig. 3 Learning curve (logarithm)

Fig. 4 Learning curve

量は1番目のリンクが10 kg, 2番目のリンクが5 kgとする。目的は、与えられた初期位置から目標位置まで、目標位置をオーバーシュートすることなしに、なるべく早くもっていくこととする。ただし、最大許容入力力は100 Nm, 最大許容角速度は2000度/秒, 最大許容角度誤差は1度, および最大許容角速度誤差は10度/秒とする。最終目標位置はすべて $\theta_1 = \theta_2 = 0$ 度とし、初期位置として、 θ_1 および θ_2 を、おのおの、 $[0 \text{ 度}, 180 \text{ 度}]$, $[-180 \text{ 度}, 180 \text{ 度}]$ 区間の一樣乱数から選び、30ケースのシミュレーションを行ってみた。一つの例として、初期位置が $\theta_1 = 84$ 度, $\theta_2 = 149$ 度の場合における10 msごとのスティック線図をFig. 2に示す。各問題を解決するために必要な時間(工数)として、学習しながら問題を解決したときにかかったCPU時間(PC-9800でBASICコンパイラを用いた時の値)と、まったく問題固有の知識がない位置で解決したときにかかったCPU時間との比(%)を採用した(実際は、CPU時間そのものを用いてもほとんど同様な結果が得られる)。このとき、横軸に問題の数の対数値、縦軸に累積平均工数(n 個の問題を解くためににかかった全工数を、 n で割った値)の対数値をとった学習曲線はFig. 3に示すとおりであり、習熟率($2n$ 個の累積平均工数/ n 個の累積

Fig. 5 Utilization of type 1 knowledge

Fig. 6 Utilization of type 2 knowledge

平均工数)は約74%になる。また、参考のため、横軸に問題の数、縦軸に工数をとった結果をFig. 4に示す。これらの結果より、このシステムがかなりの学習能力をもっていることがわかる。つぎに、各知識の利用状況を見てみる。タイプ1の知識の利用回数をFig. 5、タイプ2の知識の利用回数をFig. 6に示す。明らかに、これらの知識の利用状況は前章で述べたような傾向を示している。ただし、問題の数も少ないこともあって、いまだ熟練の域には達しておらず、タイプ3の知識については30個の問題で、1回利用されただけであった。

6. まとめ

一般に知識と問題固有の知識をはっきりと分離したことにより、プログラムの構造が明確になった。また、前回は、各問題を解くためににかかったCPU時間の面では、学習効果がまったく見られないどころか、学習とは逆の現象が発生し、試行回数の面で、わずかな学習効果をあげることができたにすぎなかった。しかし、本論文では、CPU時間の面でも、文献6)に比べて格段の学習効果をあげることができた。特に、2入力2出力系に対しては、文献6)では大型計算機を用いてもその実行がかなり困難であったが、本論文で

は、パソコンによっても十分実行可能となった。この理由は、主として、知識表現方法、一般化方法および推論方法を変更したことにより、的確な推論が行われることになったとともに、知識の増加をかなり押えることができた（前章で述べた例では、30 個の問題を解決した後保存されている知識の数は、約 500 個であった）ことによる。ただし、タイプ 3 の知識表現などの問題もあり、このままのシステムでは、完全な熟練の状態に達することは難しいものと思われるが、今後さらに研究を進めることにより、実際のシステムの制御に対して、以上述べたような方法を適用することは十分可能なことと思われる。

最後に、ノイズが存在する場合について簡単に述べる。ノイズが少ないときは、その制御目的にもよるが、本論文の方法で十分対応できるものと思われる。ただし、ノイズが多くなったときは、学習システム

は、ノイズが存在することを知り、かつ、その場合に対する処理に関する知識を所有する必要がある。

参 考 文 献

- 1) 川村, 宮崎, 有本: 学習制御方式のシステム論的考察, 計測自動制御学会論文集, **21-5**, 23/28 (1985)
- 2) 山崎, 菅野: 自動学習ファジーコントローラ, 計測自動制御学会論文集, **20-8**, 50/56 (1984)
- 3) Y. Anzai: Cognitive Control of Real-Time Event-Driven Systems, Cognitive Science, **8**, 221/254 (1984)
- 4) I. P. Goldstein and E. Grimson: Annotated Production Systems: A Model for Skill Acquisition, Proceedings of the 5th International Joint Conference on Artificial Intelligence, 311/317 (1977)
- 5) 菅沼, 伊藤: 熟練の獲得 (ロボットへの応用), 第 3 回知識工学シンポジウム資料, 51/56 (1985)
- 6) 菅沼, 伊藤: 制御における熟練の獲得と知識表現—多リンク系への応用—, 計測自動制御学会論文集, **22-1**, 84/91 (1986)