

ファジィ制御とその適用動向

解 説

菅野 道夫*

Fuzzy Control and Its Applications

Michio SUGENO

1. はじめに

今年、1989年は日本のファジィ理論の研究にとって画期的な年である。第1に、ファジィ・プロジェクトが始まったこと、第2に、新学会が設立されたことである。通産省（電子政策課）は、昨年来、産官学の協力の下、ファジィ工学の研究推進のためのプロジェクトを準備してきたが、この4月に国際ファジィシステム研究所（横浜）を発足させた。略称を LIFE (Laboratory for International Fuzzy Engineering Research) というこの研究所は技術研究組合で、民間48社の出資金と通産省からの研究委託金で運営されるものである。研究開発費は約51億で、6年間に期間を限って、ファジィ制御、ファジィ情報処理、ファジィコンピューターの開発研究を行う計画である。

また、科学技術庁は5年間で10億の研究費を投下して、ファジィシステムの基礎研究をするプロジェクトを今年からスタートさせる計画である。LIFEのように独立のオフィスと研究員を擁するわけではないが、産官学のプロジェクトチームが編成されている。

日本の研究者によるファジィ理論の研究活動は世界に先駆けて行われ、1972年にはファジィシステム研究会が発足した。同年、計測自動制御学会が最初のシンポジウムを開催した。国際的には、1973年にIFACのシンポジウムで一つのテーマとしてとり上げられたのが最初である。その後、ファジィ理論研究のワーキンググループは、1976年にヨーロッパで、1980年に入って中国と北米で相次いで結成された。これらのワーキンググループが合同して、1984年にはIFSA（国際ファジィシステム学会）が組織されるに至った。日本のワーキンググループ自体はIFSA日本支部として再組織され、活発な活動を続けてきた。そして、会員数の増加と研究の高揚を受けて、この6月に日本ファジィ学会（略称、SOFT, Society for Fuzzy Theory and Systems）が発

足することとなった。

さて、ファジィ (Fuzzy) という英語の形容詞は、「けぼのような」という意味であり、派生的に「ぼやけた、はっきりしない」という意味がある。日本の研究者は「あいまい」と訳して使っている。「あいまい」という形容詞は不確かな有様を表すとても広い言葉であるが、悪い意味に使われることが多いため、最近では単にファジィというようになった。

ファジィ理論が登場したのは1965年のことで、発端はカリフォルニア大学のザデー (ZADEH) 教授によるファジィ集合という一編の論文にある。ザデーは制御理論の専門家であり、そのせいか、ファジィ理論はまず制御工学者に受け入れられ、応用も制御を中心に行われてきた。

ファジィ集合という概念を考えついた動機として、ザデーは「不適合性の原理」というものをあげている。この原理というのは、システムの複雑さが増し、ある限度を超えると、システムの挙動についての正確でかつ意義のある記述は不可能になるというものである。限度というのは、これを超すと、正確ということと役に立つという要請が互いに相容れないものになる複雑さのレベルのことである。正確さを期そうとすると、システムの変数とパラメーターの数が膨大なものとなり、システムの全体像が見えず、使い物にならないモデルになってしまう。一方、見通しのよい役に立つモデルというのは、量的関係を犠牲にした定性的なものであるだろう。

一つの例として、ザデーは2台の止まっている車の間に車を駐車させる問題をあげて説明している。この問題をふつうの制御理論の方法に従って、車の運動方程式を立てることによって解こうとしたら、駐車することはほとんど不可能である。地面やタイヤの状態などを正確に測定することはできず、運動方程式の構造はともかくとして、パラメーターを決定することはできない。しかし、自動車の運転教習所で、未経験の生徒は車の動きを記述する方程式を知ることなく、教官から口頭で教わるだけ

平成元年6月5日受付 (Received June 5, 1989) (依頼解説)

* 東京工業大学大学院総合理工学研究科教授 工博 (Graduate School of Science and Engineering at Nagatsuta, Tokyo Institute of Technology, 4259 Nagatsuta Midori-ku Yokohama 227)

Key words: fuzzy sets; fuzzy logic; fuzzy reasoning; fuzzy control.

で、うまく駐車させることができる。生徒が教わることは「ハンドルを右に切りながら前進し、左へ戻して止まる。つぎに右に切りながら後退し、左へ戻す。失敗したらやり直す」ということである。

ファジ理論の最初の応用研究がなされたのは1974年のことで、スチームエンジン模型のファジ制御であった。ファジ制御というのは、制御におけるエキスパートシステムといってもよいもので、ファジ理論を用いて熟練オペレーターの制御経験・知識を表現し、計算機を用いて実行させようとするものである。実際、現在流行のエキスパートシステムのごく初期の成功例はファジ制御である。1980年にはセメントキルンの実プロセスに適用されるようになった。この応用は初め、デンマークで行われ、セメントキルン用のファジ制御器が商品化され、現在、ヨーロッパ、アメリカで使われている。

制御にファジ理論を応用する試みは、この後、日本で活発になり、その先鞭をつけたのは、富士電機による浄水場や下水処理場のファジ制御、それから、日立製作所による仙台市の地下鉄のファジ自動運転システムの開発であった。この地下鉄の制御は、ファジ理論が初めて市民生活に直接関連をもったという点で画期的なものであり、ファジという用語が一般の人々に知られるきっかけとなった。また、松下住設は、ファジ理論に基づいて制御する家庭用の温水給水器をこの4月に商品化したので、ファジ理論はいまや一般家庭の中にも浸透する時代になっている。

いくつかの実例を3節で紹介する。

2. ファジ集合と論理

ファジ理論とは、ファジネスという不確かさを扱う理論枠組のことであり、ファジ集合論、ファジ論理、ファジ測度論の総称である。

ファジ集合では言葉の意味や概念の定義に見られるような不確かさを扱う。

まず、ファジ集合論で扱われるファジネスという不確かさの様相と、確率論で扱われる、たとえばランダムネスという言葉に代表されるような、不確かさの違いを見てみよう。図1は「平熱」という概念の意味を、ファ

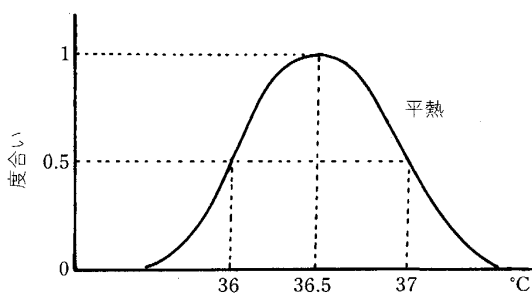


図1 「平熱」の意味

ジ集合論的にグラフに表したものである。横軸は体温を示し、縦軸は36°Cや37°Cがどの程度「平熱」と見なせるか、その度合いを示している。度合いが1なら完全に「平熱」と見なせ、0に近づくと「平熱」ではなくなる。ほとんどの自然言語は図に示すように意味はあいまいで0か1かというように2値論理的には表現できない。「平熱」と「非平熱」の境はあいまいで、ぼやけている。別の言い方をすると、「平熱」と「非平熱」は互いに矛盾概念ではなく、両者の中間的領域が存在する。すなわち、自然言語の世界は排中律が成立しない世界である。

確率論が扱う不確かさはいろいろあるが、最も一般的な不確かさは、「サイコロを投げて3の目が出る」「明日は雨が降る」など、事象の生起に関わるものである。これらは実験をすればはっきりしたり、時の経過を待てば確定するといった性質の不確かさである。これに対して、「平熱」とか「中年」などの言葉の意味に見られる不確かさは、実験や時の経過などによっても解消されない本質的なものである。

もう一つの例として、「およそ2」とか「およそ5」という概念を考えてみよう。これはファジ理論ではファジ数と呼ばれるものであり、「平熱」と同じように意味をグラフで表すと、たとえば図2のようになる。ここで「およそ2」を表すグラフは、たとえば

$$h(x) = e^{-(x-2)^2}$$

と書くことができる。

さて、 X をある集合とする。 X の部分で、境界が不明確であるような領域のことを X のファジ(部分)集合という。ファジ集合は必要に応じて、 A 、 B などと表し、 X のふつうの部分集合 E 、 F などと区別する。ファジ集合 A の境界のあいまいさは X の要素 x が A に含まれると見なせる度合い $h_A(x)$ によって表される。このとき、関数

$$h_A: X \rightarrow [0, 1]$$

は A のメンバーシップ関数と呼ばれる。ここで、 X は全体集合、あるいはファジ集合 A の台(集合)と呼ばれる。図1と図2はそれぞれ「平熱」「およそ2」「およそ5」と名付けられたファジ集合のメンバーシップ関数を表している。

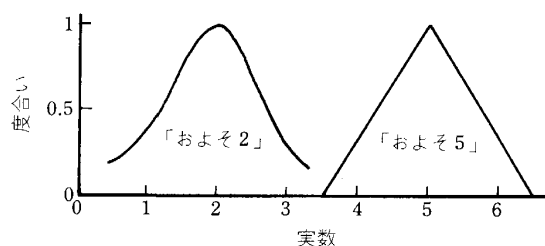


図2 ファジ数「およそ2」と「およそ5」

ふつうの集合はファジィ集合と区別するとき、クリस्प (Crisp) 集合と呼ばれ、 X のクリस्प (部分) 集合 E の定義関数とは、

$$\chi_E(x) = \begin{cases} 1 & x \in E \\ 0 & x \notin E \end{cases}$$

と定められる関数

$$\chi_E: X \rightarrow \{0, 1\}$$

のことである。メンバーシップ関数とは定義関数を拡張して、関数値として0と1だけではなく、その間の任意の値をとらせるようにしたものである。

ファジィ集合間の関係、演算などはすべてメンバーシップ関数によって間接的に定義される。和集合、共通集合、補集合はそれぞれ次のように定義される。

$$A \cup B \leftrightarrow h_{A \cup B}(x) = h_A(x) \vee h_B(x)$$

$$A \cap B \leftrightarrow h_{A \cap B}(x) = h_A(x) \wedge h_B(x)$$

$$A^c \leftrightarrow h_{A^c}(x) = 1 - h_A(x)$$

ただし、

$$a \vee b = \max \{a, b\}, \quad a \wedge b = \min \{a, b\}$$

を表す。

これらの演算をグラフで表すと、図3、図4のようになる。これらのファジィ集合演算はクリस्प集合演算の性質をほとんど保存する。

論理学は「2は偶数である」とか「ソクラテスは人間である」などの命題の真偽を問題にする学問である。いわゆる数学の論理というのは2値論理と呼ばれるものであり、真偽を真か偽かと二者択一的に決めることのできる命題だけを扱うものである。論理命題は一般に「 x は A である」のように、主語(x)-述語(A)の形式で表せるが、2値論理の命題では、述語は疑う余地なく明確に定義された概念であると仮定されている。ここでは命題

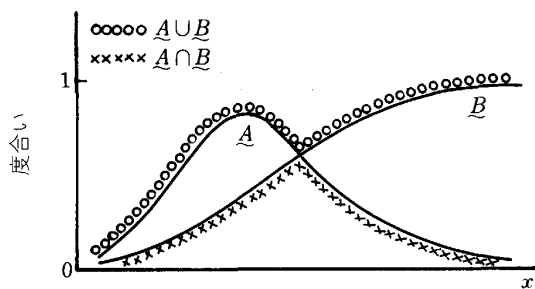


図3 和集合と補集合の演算

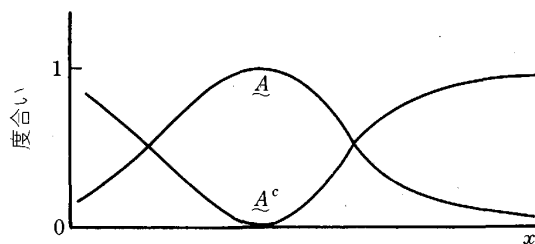


図4 補集合の演算

のもつ意味のあいまいさなどは問題にされることはない。

これに対して、ファジィ論理では「千は大きい数である」「彼女は若い」などのように、あいまいな述語をもつ命題が扱われる。述語の意味のあいまいさは、もちろんファジィ集合論的に表現される。2値論理と違ってファジィ論理では、「大きい数」とは、「若い」とは、などと命題の意味を問題にしているといつてよい。2値論理と区別してファジィ論理を特徴づけるもう一つのものとして、推論の形式があげられる。2値論理では、

(前提1) ソクラテスは人間である

(前提2) 人間は死ぬ

(結論) ソクラテスは死ぬ

のように、いわゆる3段論法により推論が行われる。前提1と前提2に含まれる共通項「人間」を媒介として「ソクラテスは死ぬ」という結論を推論する。結論の述語「死ぬ」は前提2の述語と同じである。

これに対して、ファジィ論理は

(前提1) 凍っている道は滑る

(前提2) 滑りやすい道は危険だ

(結論) 凍っている道はとても危険だ

というもので、前提1と2を媒介する共通項はなく、「滑る」「滑りやすい」という似ているものがあるにすぎない。また、結論の述語「とても危険だ」は前提2の述語「危険だ」と同じではない。このような推論は人間が日常的に行っているものであるが、2値論理では扱うことはできず、代わってファジィ論理はこれを可能にする。ここでの問題は、「滑りやすい」と「滑る」の意味の隔たりを測ることである。「滑りやすい」はもちろん「滑る」と「滑らない」の間であって、「滑る」に近い概念であると了解するわけである。

3. ファジィ制御の方法

3.1 ファジィ制御規則

「曲がり角に近づいたら徐々に減速し、曲がり角に入りかけたらハンドルをゆっくり右に切る。そして、曲がり終える少し前からハンドルを戻しつつ、徐々に加速する」。これは、車を右折させるためのファジィアルゴリズムである。ファジィアルゴリズムでは、

if x is A then y is B , if not then y is C and then

if x is D then y is E , if not then y is F and then...

のように、条件付き指示が直列的につながるが、ふつうのファジィ制御では if~then~ 型の制御規則が並列的に置かれ、同時に働く場合が多い。自動車の運転制御や移動ロボットの制御などの場合には、直列型と並列型の併用も考えられるだろう。

さて、ファジィ制御ではつぎのようなファジィ制御規

則が用いられる。

(R1) if x is A_1 and y is B_1 then z is C_1

(R2) if x is A_2 and y is B_2 then z is C_2

ここで、制御規則 R1 と R2 は論理的には OR 結合され、規則の適用条件に合えば並列的に働くものである。A, B, Cなどは、規則の中の「大きい」「小さい」「およそゼロ」などの言葉のあいまいさを表現するファジィ集合である。簡単のためにファジィ集合 A, Bなどのメンバーシップ関数を $A(x)$, $B(y)$ などと表すことにする。

制御規則の if の部分は前件部 (条件部) then の部分は後件部 (結論部) と呼ばれる。前件部の x , y は前件部変数または入力変数、後件部の z は後件部変数または出力変数と呼ばれる。ファジィ集合 A, B, Cなどはいろいろな値をとり得るので、ファジィ変数、さらには言語変数と呼ばれることもある。ふつうのプラント制御の場合は、後件部変数は制御のための操作量 (プラントへの入力) であり、前件部変数は操作量を決めるのに役立つ制御器への入力情報である。

ファジィ制御の例として、1入力1出力プラントの定置制御を考えてみる。操作量を u 、1 サンプル間の操作量の変化分を Δu 、設定値からの出力の偏差を e 、偏差の変化分を Δe とする。あるサンプリング時刻で、これまで入力されていた操作量 u を変化させる量としての Δu を後件部変数に選ぶとすると、制御のための常識的な情報は e と Δe である。そこで、たとえば制御規則は

(R1') if e is PB and Δe is ZO then Δu is PB

(R2') if e is PB and Δe is NS then Δu is PM

のようになる。ここで、P は Positive, N は Negative, B は Big, M は Medium, S は Small, ZO はおよそ Zero を表す。そこで R1' は「もし偏差が正で大きく、偏差の変化分がほとんどゼロなら、操作量を正で大きく増やせ」という言語的規則を意味する。

ファジィ変数がとる値 PB, PM, NSなどのメンバーシップ関数は、たとえば左右対象のつり鐘型 (図5(a)) と三角型 (図5(b)) のものが選ばれることが多いが、ときには、台形型 (図5(c)) や非対称形にするなど工夫をこらすこともある。一般に、制御規則の数が多い場合はメンバーシップ関数の形は標準的なものでよいが、数が少ない場合は形をうまく選ぶことが必要となる。

上に示した制御規則の形式は最も一般的なもので、他にいろいろな変形がある。ただし、前件部にファジィ変数を用いることは共通である。簡単な形式として、次のような後件部にファジィ値の代わりに直接数値を書くものが考えられる。

(R3) if x is A_1 and y is B_1 then $z = z_1$

(R4) if x is A_2 and y is B_2 then $z = z_2$

この形式を複雑にすると、次のようになる。

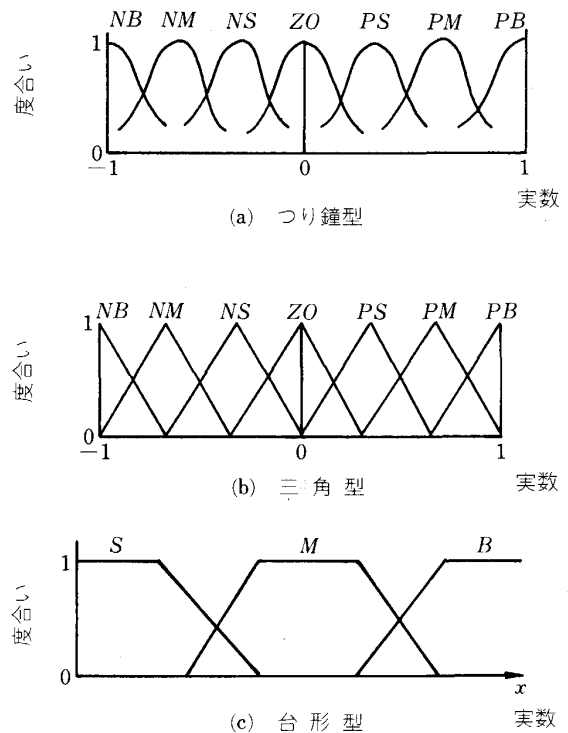


図 5 メンバーシップ関数の型

(R5) if x is A_1 and y is B_1 then $z = f_1(x, y)$

(R6) if x is A_2 and y is B_2 then $z = f_2(x, y)$

この例では、後件部には z がとる数値やファジィ値ではなく、 z を x , y から決める関数が書かれている。

3.2 ファジィ推論

ファジィ推論とは、前件部変数 x , y についての入力情報を得て、後件部変数 z がとるべき値を推論するものであるが、ふつうは複数個の規則を用いて、総合的に推論が行われる。ただし、ファジィアルゴリズムを単に実行するような場合は、規則の数は一つと考えられる。

推論の手続きはどの形式の規則を用いても、ほとんど同じで、次の三つのステップからなる。

(1) 規則の前件部の適合度を入力情報について計算する。

(2) 各規則の推論結果を求める。

(3) 各規則の推論結果を適合度を重みとして、重み付き平均する。

規則が R1, R2 である場合について考えてみる。プラント制御では、入力情報はふつう $x = x^0$, $y = y^0$ のように数値で与えられることが多い。このとき、入力 x^0 , y^0 に規則の前件部に条件がどの程度適合するか、その度合いを w_1 , w_2 として

$$w_1 = A_1(x^0) * B_1(y^0)$$

$$w_2 = A_2(x^0) * B_2(y^0)$$

のように定める。ただし、 $A_1(x^0)$ はファジィ集合 A_1 の $x = x^0$ におけるメンバーシップ関数の値である。ま

た、*は Min 演算またはかけ算である。もし、入力情報がファジィで、 x is A° , y is B° と与えられたときは、 $A_1(x^\circ)$ の代わりに

$$\max_x [A_1(x) \wedge A^\circ(x)]$$

を用いればよい。 $B_1(y^\circ)$ についても同様である。

つぎに、R1, R2 による推論結果をそれぞれ $w_1 * C_1(z)$, $w_2 * C_2(z)$

とする。二つの規則を用いた推論結果を z is \hat{C} であるとして、そのメンバーシップ関数 $\hat{C}(z)$ を

$$\hat{C}(z) = \max [w_1 * C_1(z), w_2 * C_2(z)]$$

によって定める。最後に

$$z^\circ = \int \hat{C}(z) z dz / \int \hat{C}(z) dz$$

によって、 z の値 z° を推論する。いわゆるファジィ推論とは z is \hat{C} を求めるところまでであるが、制御の場合、プラントの操作量は数値でない困るので、このようにする。 z° は関数 $\hat{C}(z)$ の重心に相当し、 z° を求めることはファジィ推論結果の非ファジィ化、または解釈と呼ばれている。また、 $\hat{C}(z)$ を構成し、重心を求める過程は形式的に R1, R2 による推論結果の重み付き平均をとったことに相当している。

R3, R4 また R5, R6 の形式を用いるときは、入力 x° , y° について、ステップ(2)の z の値は直接求められる (R3, R4 のときは入力によらず一定) ので、ステップ(3)は文字どおり、重み付き平均をとることになる。

さて、推論法はどの形式の規則でもほとんど同じであると述べたが、ステップ(2)(3)の代わりに、次の(2')としてもよい。

(2')適合度が一番大きい規則による推論結果を採用する。

この場合は、複数個の規則が同時に働くことにはならないが、論理的判断が重要な状況では有効な方法となる。

3.3 ファジィ制御器の設計

ファジィ制御における制御器設計の考え方は、古典制御の方法に近いものであり、評価関数を用いることはせず、多様な制御の目的を考えつつ、いわばエキスパートシステムの考えに基づいて制御規則を作成する。

ファジィ制御器を設計するという事は、ファジィ制御規則を作るということである。具体的に決めるべき項目をまとめると、まず、制御規則は前件部と後件部の二つの部分に分けられ、前件部に関して決めるべきことが三つある。はじめに、制御のための入力情報である前件部変数 x , y などを決めること、次に、条件の形式を決めること、最後に、ファジィ集合のメンバーシップ関数を決めることである。はじめの二つ、前件部変数と条件の形式に関する部分は制御規則の(前件部の)構造と呼ばれる。メンバーシップ関数の部分は制御規則のパラメーターである。

ルールベースによる一般のエキスパートシステムと

違って、ファジィ制御の場合は、前件部変数を見出すことはさほど難しいことではない。しかし、視覚情報を使ってロボットを制御するなどの場合は、何を入力情報とすればよいかも大きな問題となる。最大の問題は、条件の形式という制御規則の構造を決定することである。条件の形式というのは、見方を変えると、入力空間をどのようにファジィに分割するかというファジィ分割の仕方のことである。たとえば、 x - y 平面を① x が Small で y が Small, ② x が Big で y が Small, ③ y が Big (x は何でもよい)、の三つの領域に分けることが規則の条件の形式を決めることに相当する。

後件部については、制御器の出力としての後件部変数と後件部のファジィ変数のパラメーターを決めることが問題になる。後件部変数も、ふつうはプラントへの入力である操作量が選ばれるので、おのずから決まることになるが、必ずしも物理量とする必要はない。たとえば、ファジィ制御器が階層構造をしているとき、上位の規則の後件部変数は下位の規則におけるパラメーターであるという場合も考えられる。

これまでに試みられてきた設計法はつぎの四つの種類に分けられる。ただし、一つの方法で制御規則の構造とパラメーターのすべてを決めることは難しい。シミュレーターの利用、あるいは制御対象の数学モデルを使った計算機シミュレーションが可能な場合は、これにより、たとえば規則のパラメーターの決定や調整を行うことができる。しかし、これは当然のことであるから、独立した設計法とは考えないことにする。

(1) エキスパートの経験・知識

熟練オペレーターの経験や制御工学者の専門知識によって、制御規則を作る方法で、プラントの特徴についてのおよその知識を利用している。経験則は定性的な言葉で表現されないと、ファジィ制御規則の形に論理化できない。制御の場合ほとくに、経験や知識は技能として蓄えられていることが多いので、言葉でノウ・ハウを言い表すのは難しいことがある。

この方法はとくに制御規則の構造を決めるのに役立つが、構造の中でも、どのような情報を用いて、操作量を決めているかなどは、オペレーターにインタビューすることによって容易に分かることがある。また、条件の形式についても、たとえば、入力情報の x については、Small, Medium, Big と三つの場合に分けている。 y については Small, Big の二つに分けて考えている、などのことを聞くことができる。ただし、この方法では制御規則のパラメーターは決定できないので、他の方法を併用する必要がある。

(2) 人間オペレーターの操作モデル

ファジィ制御の考え方はエキスパートシステムの考え方に似ているが、前に述べたように、熟練オペレーターの経験を言葉で表現させるのは往々にして難しい。また、

言語的規則が得られたとしても、余りに大ざっぱで、使用に耐えない場合もある。

このようなとき、有力な設計法として、オペレーターの操作のモデリングの方法がある。これはオペレーターが使う情報と彼の操作出力との間の入出力モデルを作るものである。モデルとして、if~then~形式を選べば、そのまま制御規則として使うことができる。具体的に行うことは、オペレーターが操作する場合の入出力データを用いて、モデルの構造とパラメーターを決めることである。ただし、入出力データのみからモデルの構造、すなわち制御規則の前件部の構造を決めるのは難しい。構造については、たとえば(1)の方法であらかじめ見当をつけておき、ここではパラメーターを決定することを主としたほうがよい。

この方法で一つの問題になるのはデータの入手である。物理的に計測できるデータは問題ないが、オペレーターが目で判断しているプラントの状態などは、現場でオペレーターに記録してもらうなどしなければならぬ。

(3) 学習と適応

この方法は(1)(2)のようにエキスパートが存在しない場合の設計法である。学習の対象となるのは、制御規則の条件の形式とパラメーターである。制御規則の数がゼロの状態から始めて、自動学習により、制御規則を生成する研究もあるが、あまり実用的ではない。現実的な学習はオフラインで、制御規則を逐次的に改良していく程度のものであろう。パラメーター学習はもちろん一番簡単だが、構造についても、 x を Small と Big に分割していたものを、Medium の場合も追加するといったぐらいのことは十分可能である。いずれにせよ、学習するには実システムを用いた実験が可能か、もしくは制御対象のシミュレーターが必要となる。ふつうの工業プラントではこのようなことはほとんど考えられないが、ロボットなどの制御の場合は、この方法は有益な設計法となるだろう。

一方、適応というのはオンラインで用いる方法で、あらかじめ制御対象の数式モデルを作り、これに基づいて制御器を設計しておき、実システムとモデルとの違いをオンライン的に検出し、制御規則を修正あるいは生成するというものである。モデルはかなり近似の荒いものでよい。適応の論理はそれ自体 if~then~形式で表現することができる。この場合、適応規則の前件部変数は実システムを近似するモデルのパラメーター(たとえば、プロセスゲイン、遅れ時間)とか、実システムの変化する環境を特徴づける変数であり、適応規則の後件部変数は制御規則のパラメーターとするのが自然である。学習と適応を比べると適応のほうが簡単で応用範囲は広い。問題は適応規則のための入力情報を検出することである。

(4) プラントのファジィモデル

エキスパートシステムの考え方はエキスパートの真似をしようということなので、エキスパートを超えることはできない。エキスパートがいない状態では(1)(2)の方法は適用できず、また(3)の方法も、学習のための環境が作れないとか、また適応のための情報が入手できないなどの理由で使えない場合がある。

この方法はプラントのファジィモデルから理論的にファジィ制御規則を導こうとするものである。このためにはプラントのファジィモデルを作らねばならないが、ファジィモデルはやはり if~then~形式でプラントの動特性を記述したものが望ましい。制御規則と同じ形式で記述することにより、制御規則を導くのが容易になる。最も単純な考え方は、プラントのファジィモデルにファジィ制御器をつないだとき、全体が望ましい挙動を示すように、制御規則を作るというものである。記述力が大きいという点から、後件部が関数になっている R5, R6 がモデルの形式として適している。

問題はファジィモデルの同定である。同定とはプラントの入出力データを用いて if~then~形式の構造とパラメーターを決めることである。

4. ファジィ制御の実例

ファジィ制御は、制御におけるファジィエキスパートシステムともいえるものであるが、実用化が進まない通常のエキスパートシステムと比べて、とくに実用化例が多い分野である。1988年の12月現在で、実用化を目指すファジィ制御の国内のプロジェクトは約80あり、その内約40はすでに実用化されているものである。

以下ではファジィ制御の主な実用化例を紹介する。

4.1 セメントキルンの制御⁶⁾

デンマークの F. L. Smidth 社は、世界に先駆けて1981年セメントキルン用のファジィ・コントローラーを開発し、実運転に入った。

セメントキルンというのは、砂、粘土、生石灰からセメントクリンカーを製造するプラントである。セメントはクリンカーを細かく砕くことによって得られる。セメントキルンの制御は(1)排ガス中の O_2 , NO_x , CO の割合、(2)排ガスの温度、(3)キルンの回転トルク、(4)クリンカーの密度、から燃料と排ガスダンパーの開度の二つのプラント入力を操作するものである。セメントキルンというのは非線形性があり、また時間と共に特性が変わるプラントであるため、モデルを作るのが難しく、熟練したオペレーターの操作に頼っているプラントである。F. L. Smidth 社のファジィコントローラーは、その後いくつかのセメントキルンに実装されている。中でも、アメリカのオレゴン・ポートランドセメント会社の例では、オペレーターによる従来の制御に比べ、4%位のエネルギーの節約が達成され、炉壁レンガの寿命が50%延びている。

日本でも、1987年に大阪セメントが独自にファジィ制御を適用し、従来困難とされていたセメントキルンの運転の全自動化に成功した。

4.2 地下鉄の制御⁷⁾

従来の列車の運転制御方式は、いわゆるPID制御と呼ばれるもので、あらかじめ設定された駅間の走行速度パターンとのずれを小さくしようとするものであった。しかし、これは乗客の乗り心地の良さなどを考慮した制御方法ではない。熟練した運転者は走行速度パターンとのずれなどではなく、乗り心地、電力消費、停止精度などさまざまな運転指標を頭に入れて、巧妙に運転している。日立製作所は、運転者のこれらのノウ・ハウをファジィ制御規則の形で抽出し、仙台市の地下鉄の自動運転装置に応用した。

熟練した運転者は、「もしブレーキを少し強くしたとき、乗り心地よく、正確に停止できるならば、ブレーキを少し強くする」というようなロジックで数秒先の列車の状態を予測しつつ、巧妙に制御しているといわれる。ファジィ制御の方法によれば、このような経験的制御規則を容易に表現できる。1987年夏に実用化された自動運転装置は、24個の制御規則をもち、1sに10回の割合で制御指令を与えている。

従来の制御方法と比較した結果、ファジィ制御の方がプラットホームでの停止精度および乗り心地とも3倍くらい向上することが分かり、熟練運転者も違和感なく、ファジィ制御による自動運転に任せられるとのことである。

4.3 ガラス熔融炉の制御⁸⁾

日本電気硝子はガラス熔融炉の温度制御にファジィ制御を適用している。炉はSiO₂やAl₂O₃を含むガラス原料を1400~1600°Cで熔融して、ガラスを製造する反応槽である。原料を投入してからガラスができあがるまで、十数時間もかかる上、炉の数式モデルを作ることが困難なので、熟練したオペレーターの運転に頼っているのが現状である。

日本電気硝子はガラスの専門家とオペレーターの知見から、炉の温度を一定にするためのファジィ制御規則を作り、すでに実用化している。ファジィ制御を熟練したオペレーターによる制御と比較した結果、温度の管理精度が2倍以上も改善されることが分かった。

4.4 シールドトンネル掘削機の制御⁹⁾

東京電力と奥村組はトンネル掘削のためにファジィ制御を応用し、シールドトンネル掘削機の自動化を可能にした。シールドトンネル掘削機とは、地表に影響を与えず、直径5mくらいのトンネルを掘りながらコンクリートでシールドする機械で、従来、掘削速度と掘削方向の制御は自動化できず、熟練オペレーターに任せられてきた。東京電力はオペレーターのノウ・ハウをファジィ制御規則の形式で抽出し、奥村組と協力し、機械の自動化

に成功した。この掘削機は現在、東京都営地下鉄のトンネルを掘るのに使われており、熟練オペレーターより良好な性能を示している。

4.5 自動車の速度制御⁵⁾

日産自動車は自動車の自動速度制御にファジィ制御を用いている。これは運転者がアクセルとブレーキを操作することなく、一定速度で自動車を走行させる装置であり、従来はPID制御で行われていた。日産自動車はファジィ制御に学習機能を組み込み、性能を大幅に向上させることに成功した。たとえば、上り坂にかかる時、車速は落ち、始めは徐々に加速して元の速度を回復するが、数回坂道を走行すると、この装置は学習により、加速の度合いを自動的に改善し、素早く元の速度を回復するようになる。

4.6 混合給湯器の制御¹⁰⁾

混合給湯器とは使用者が希望の湯温を指定し、蛇口を開くと、高温の湯と水を混合し、要求された温度と流量の湯を供給する機器である。従来PI制御により、制御が行われてきたが、オーバーシュート、オフセットの点で難点があった。松下住設機器は21個の制御規則からなるファジィコントローラーを開発し、従来の性能を飛躍的に改善することに成功した。たとえば、42°Cと指定すると、従来機器では一時的に約50°Cくらい(8°Cのオーバーシュート)の湯が出て、火傷しそうになるが、ファジィ制御による機器では高々43°Cくらいになるだけですむ。

4.7 鉄鋼プロセスの制御⁵⁾

新日本製鉄は鉄鋼プロセスにおける冷延工程にファジィ制御を適用した。薄板鋼板を圧延により製造する冷延工程では、従来から圧延理論によるセットアップモデルを用いて圧延制御を行っている。これは製品の仕様に基づいて、圧延機にかかる圧延荷重をあらかじめ推定し、セットするものである。圧延荷重は材料にかかる張力によって変える必要があるが、張力以外のさまざまな圧延材の特性に依存する。これらの中には測定が難しいものもあり、操業条件がたびたび変わるなどのため、理論どおりにはいかないのが普通で、熟練オペレーターの介入が必要とされている。

この工程にファジィ制御を適用した結果、従来方式に比べて、圧延荷重の推定誤差を大幅に減少できることが分かった。さらに、学習機能を取り入れることにより、材料の変化にも十分に対応できることが分かった。

鉄鋼関係では、この他にもNKKによる高炉制御⁵⁾が実用化され、たいへん良い成績を収めている。

5. 今後の課題

ファジィ制御の重要な課題はいくつか考えられるが、まず、制御対象のいわゆるファジィモデルとファジィ制御器をつなげたファジィ制御システムの解析理論の開発

が必要である。解析というのは、たとえば、安定性解析である。ファジィ制御システムは本質的に非線形であり、またシステムの記述言語（複数個の if-then 規則）が従来の言語（伝達関数、微分方程式）と著しく異なるため、理論的解析はとても難しい。

つぎに、産業応用の面では、信頼性のある学習制御方式や対話型制御方式の開発が必要とされよう。ファジィ制御器のメンテナンスの方法と考えてもよい。

今ひとつ、重要なのは画像情報、言語情報の利用も可能にする知的ファジィ制御方式の開発であろう。知能化というのは、現代のキーワードであるが、実用化は難しい。ファジィ制御は、従来のような記号論理ではなく、ファジィ性を取り入れた論理を用いているので、いわゆる“意味処理”がしやすく、知能化を図れる可能性がある。

文 献

- 1) L. A. ZADEH: *Information and Control*, **12** (1968), p. 94
- 2) E. H. MAMDANI: *Proc. IEE*, **121** (1974), p. 1585
- 3) 菅野道夫: ファジィ制御 (1988) [日刊工業新聞社]
- 4) ファジィシステム入門 (寺野寿郎, 浅居喜代治, 菅野道夫編) (1987) [オーム社]
- 5) 応用ファジィシステム入門 (寺野寿郎, 浅居喜代治, 菅野道夫編) (1989) [オーム社]
- 6) L. P. HOLMBLAD and J. J. OSTERGAARD: *Control of a Cement Kiln by Fuzzy Logic, in Fuzzy Information and Decision Processes*, ed. by M. M. GUPTA and E. SANCHEZ, North-Holland (1982)
- 7) 安信誠二: システム制御, **32** (1988), p. 629
- 8) S. AOKI, S. KAWACHI and M. SUGENO: *Proc. of 3rd IFSA Congress, Seattle* (1989), p. 212
- 9) 桑原 洋, 原田光男: 計測と制御, **27** (1988), p. 70
- 10) K. UEKI and H. FUJIEDA: *Proc. of 3rd IFSA Congress, Seattle* (1989), p. 216