日本ファジィ学会誌 Vol.4, No.2, pp.379-388(1992)

論文

デルタルールによるファジィ推論の自動チューニング手法と障害物回避への応用†

野村 博義* 林 勲* 若見 昇*

ファジィ推論を用いて制御対象を目標の仕様にあうように制御するには、推論ルールのチューニングが必要である。従来、このチューニングを自動的に行う手法がいくつか提案されている。しかし、これらの手法は、チューニングにかかる演算時間が長く、また、十分な汎化能力が得られないという課題を有していた。本論文では、ニューラルネットの学習則であるデルタルールを用いて自動的に推論ルールをチューニングする新たな手法を提案する。本手法は、前件部のメンバーシップ関数を推論ルールごとに独立に設定し、与えられた入出力データに適合するメンバーシップ関数のみをチューニングする。このため、短い学習時間で汎化能力の高い推論ルールの構築が可能である。ここでは、本手法のアルゴリズムを定式化し、簡単な数値例と移動ロボットの障害物回避問題に適用して本手法の有効性を示す。

キーワード：ファジィ推論、学習制御、ニューラルネット、障害物回避

1. はじめに

最近、ファジィ推論の機器制御への応用が活発に行われている[1]。ファジィ推論は、専門家の持つ定性的な知識を「If～Then…」形式の推論ルールとして表現することができる。しかし、ファジィ推論を用いて、制御対象を目標の仕様に合うように制御するには、推論ルールの調整が必要となる。この調整の作業をチューニングと呼んでいる。従来では、チューニングは試行錯誤の実験によって行なわれることが多く、そのため最適な推論ルールを構築するには長い開発時間が必要であった。特に、多入力を持つ対象を取り扱う場合には、チューニングの対象となるパラメータが多くなり、試行錯誤によるチューニングは非常に困難であった。

このような問題を解決する自動チューニング手法として、ニューラルネット駆動型ファジィ推論[2]や、逐次型ファジィモデリング[3),4)]などがある。特に逐次型ファジィモデリングでは、与えられた入出力データから自動的にその入出力関係を短時間で学習することができる。しかし、学習に用いていないデータに対する入出力関係を十分に表現できないという、汎化能力の向上に課題があった。

本論文では、逐次型ファジィモデリングを拡張し、より汎化能力を高めた学習型のファジィ推論モデルを提案する。提案するファジィ推論モデルでは、ニューラルネットの学習則であるデルタルール[5]を用いて、与えられた入出力データから前件部のメンバーシップ関数のパラメータと後件部の実数値を自動的にチューニングする。本手法では、前件部のメンバーシップ関数を推論ルールごとに独立にチューニングするので、与えられた学習用のデータに対して適合したメンバーシップ関数のみが調整される。これにより、汎化能力の高い推論ルールが短い学習時間で得られる。

† A Self-Tuning Method of Fuzzy Reasoning by Delta Rule and Its Application to a Moving Obstacle Avoidance
Hiroyoshi NOMURA, Isao HAYASHI and Noboru WAKAMI

* 松下電器産業株式会社 中央研究所
Central Research Laboratories Matsushita Electric Industrial Co., Ltd.

ここでは、まず、本手法の基礎となる逐次型ファジィモデリングについてその概略を述べ、提案する本手法の学習アルゴリズムを説明する。次に簡単な数値例に適用して本手法と他手法の比較を行う。また、本手法の有効性を示すため、移動ロボットの動的障害物回避問題に適用した例について述べる。

2．逐次型ファジィモデリング

市橋らの提案した逐次型ファジィモデリングの概要について説明する。

逐次型ファジィモデリングでの推論ルールは、次式のように表されている。

Rule i：

if x_1 is M_{1k} and ... and x_m is M_{mk}

then $y = p_{i0}+p_{i1}\cdot x_1+ \cdots +p_{im}\cdot x_m$ (1)

$(i=1,\cdots,n)$

ただし、iはルール番号、x_j(j=1,…,m)は入力変数、M_{jk}は入力変数x_jの定義域におけるk番目のメンバーシップ関数、p_{ij}は、i番目の推論ルールの後件部の線形関数の第j番目の係数を示している。ただし、添え字 k はルール番号iと入力変数番号jの関数として表現されている[3]。

逐次型ファジィモデリングでは前件部のメンバーシップ関数M_{jk}に次式のような条件をつけている。

$$\sum_{k=1}^{r_j} M_{jk}(x_j)=1 \qquad \text{for} \quad \forall x_j \qquad (2)$$

ただし、r_j は入力変数x_jに対するメンバーシップ関数の総数である。

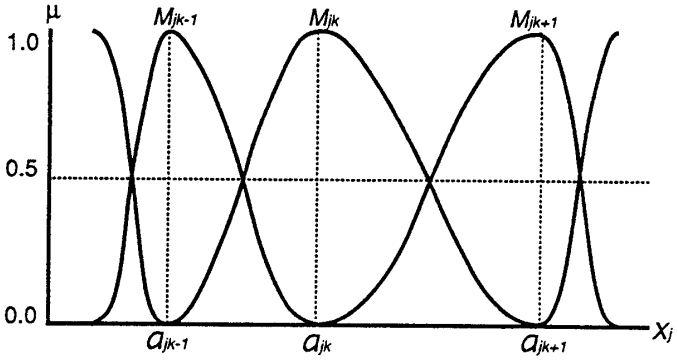


図1 逐次型ファジィモデリングに用いる前件部のメンバーシップ関数

(2)式の条件は、入力変数x_jに対するメンバーシップ関数の適合度の総和が1になることを表している。図1に前件部のメンバーシップ関数の一例を示す。ただし、メンバーシップ関数M_{jk}の中心をa_{jk}としている。(2)式の条件により図1では、隣あうメンバーシップ関数がメンバーシップ値0.5で交わっている。逐次型ファジィモデリングでは、このメンバーシップ関数の中心値a_{jk}と(1)式の後件部の線形関数の各係数$p_{i0},\cdots,p_{im}$を最急降下法を用いてチューニングしている。

逐次型ファジィモデリングは、バックプロパゲーション型のニューラルネット[5]と比較して高速な学習が可能であることが報告されている[3]。しかし、(2)式の条件を設定しているので、推論ルールごとに前件部のメンバーシップ関数を自由にチューニングできず、例えば、入出力データが一様に分布していない場合には、入出力関係を十分に表現する推論ルールを構築することが困難だった。

3．デルタルールを用いたファジィ推論

逐次型ファジィモデリングを拡張し、より汎化能力のある推論ルールを自動構築できる手法を新たに提案する。

3.1 アルゴリズムの概要

提案するファジィ推論モデルでは、後件部を実数値で表現した簡略ファジィ推論[6],[7]を用いる。

いま、入力を$x_1,\cdots,x_m$とし、出力をyとすると、簡略ファジィ推論の推論ルールは次式で表される。

Rule i：

if x_1 is A_{i1} and $\cdots$ and x_m is A_{im}

then y is w_i

$(i=1,\cdots,n)$ (3)

ここで、A_{ij}は前件部のメンバーシップ関数、w_iは後件部の実数値を表す。ここでの前件部のメンバーシップ関数A_{ij}は、推論ルールごとに独立に設定されており、逐次型ファジィモデリングのM_{jk}と異なり、添え字に推論ルール番号iを持つ。

簡略ファジィ推論の推論結果 y は次式で計算できる。

$$\mu_i = A_{i1}(x_1) \cdot A_{i2}(x_2) \cdot \cdots \cdot A_{im}(x_m) \quad (4)$$

$$y = \frac{\sum_{i=1}^{n} \mu_i \cdot w_i}{\sum_{i=1}^{n} \mu_i} \quad (5)$$

ここで、(4)式の μ_i は前件部の適合度を示す。

本手法での簡略ファジィ推論は図2のようなネットワーク構造で表すことができる。

図2では、入力変数を4個、推論ルールを2個として、(4)式および(5)式の演算を表すユニットを○で、メンバーシップ関数 A_{ij} と後件部の実数値 w_i を□で表している。出力層の演算ユニットは(5)式の重心演算を表し、中間層の演算ユニットは(4)式の適合度演算を表している。

入力層に入力 $x_1, \cdots, x_m$ が与えられた場合、中間層で各推論ルールに対する適合度 μ_i を計算し、出力層で重心演算を行い、出力値 y を導出する。

本手法では、ニューラルネットの学習則の一つであるデルタルールを用いて、□で表した後件部の実数値と前件部のメンバーシップ関数のパラメータをチューニングする。チューニングのアルゴリズムでは、まず、与えられた入出力データの出力値 y^r と出力 y の推論誤差を小さくするように、後件部の実数値 w_i を更新する。次に、更新された w_i と推論誤差から前件部のメンバーシップ関数

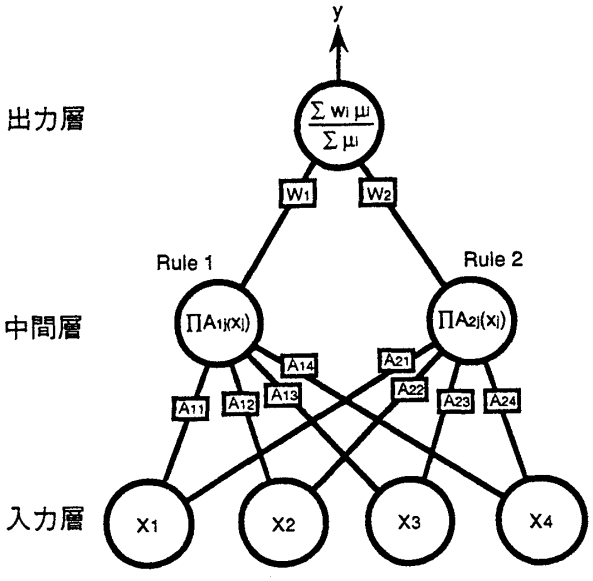


図2 本手法の構成図

A_{ij} のパラメータを更新する。このように学習のアルゴリズムは、入力データが与えられた場合の推論方向と逆方向に出力層から中間層へとパラメータをチューニングしていく。これは、バックプロパゲーション型のニューラルネットワークの学習過程と類似しているといえる。

次に、本手法と逐次型ファジィモデリングの違いについて議論する。

逐次型ファジィモデリングでは、前件部のメンバーシップ関数 M_{jk} が複数の推論ルールに共有されているので、あるデータに対して推論ルールの前件部のメンバーシップ関数をチューニングすると、そのデータに関係のない推論ルールのメンバーシップ関数も同時にチューニングされるという現象が起こる。また、(2)式の条件により、一つの入力変数に対して推論ルールの適合する範囲を3つ以上重複させることができず、与えられた入出力データの分布が偏っている場合には、チューニングされない推論ルールが存在することがある。これらより、逐次型ファジィモデリングでは学習に用いたデータと異なるデータを推論ルールに入力した場合に、真値に近い推論結果を出力できないことがある。いわゆる推論ルールの汎化能力が十分ではないと考える。

本手法では、逐次型ファジィモデリングでの(2)式の条件を除去し、(3)式のように前件部のメンバーシップ関数を推論ルールごとに独立に設定する。これらにより、入力データに対して適合した推論ルールのみをチューニングすることができる。したがって、自由度の高いチューニングが可能であり、学習に用いていないデータに対しても真値に近い値を出力することができる。したがって、本手法は高い汎化能力を持つといえる。

3.2 アルゴリズムの定式化

本手法で用いる前件部のメンバーシップ関数 A_{ij} を図3に示す。

ここで用いるメンバーシップ関数は二等辺三角形型であり、中心値 a_{ij} および幅 b_{ij} を用いて次式のように表現する。

$$A_{ij}(x_j)=\begin{cases}1-\dfrac{2\cdot|x_j-a_{ij}|}{b_{ij}} \\ \quad ;\ a_{ij}-\dfrac{b_{ij}}{2}<x_j<a_{ij}+\dfrac{b_{ij}}{2} \\ 0 \quad ;\ x_j\leq a_{ij}-\dfrac{b_{ij}}{2},\ x_j\geq a_{ij}+\dfrac{b_{ij}}{2}\end{cases} \tag{6}$$

ただし、$b_{ij}>0$ である。

本手法では、メンバーシップ関数の中心値 a_{ij}、幅 b_{ij}、および後件部の実数値 w_i を推論ルールのチューニングのためのパラメータとする。これらのパラメータのチューニングにはデルタルールを用いる。デルタルールとは、パーセプトロン[8]やバックプロパゲーション型のニューラルネット[5]の学習則であり、与えられた入出力データを用いて逐次的に学習を繰り返すことにより最適解を得る手法である。

いま、$m+1$ 次元の入出力データ $(x_1,\cdots x_m,y^r)$ が得られたとする。デルタルールの学習は次式の評価関数 E の最小化問題として定式化できることが知られている[5),9)]。

$$E=\frac{1}{2}(y-y^r)^2 \tag{7}$$

評価関数 E の値を減少させるためには、パラメータ a_{ij}、b_{ij}、w_i に対して、E が減少する方向ベクトルを計算すればよい。この方向ベクトルは $(-\partial E/\partial a_{ij}, -\partial E/\partial b_{ij}, -\partial E/\partial w_i)$ で表され、学習式は次式のようになる。

$$a_{ij}(t+1)=a_{ij}(t)-K_a\cdot\frac{\partial E}{\partial a_{ij}} \tag{8}$$

$$b_{ij}(t+1)=b_{ij}(t)-K_b\cdot\frac{\partial E}{\partial b_{ij}} \tag{9}$$

$$w_i(t+1)=w_i(t)-K_w\cdot\frac{\partial E}{\partial w_i} \tag{10}$$

ただし、K_a, K_b, K_w は学習に用いる係数であり、t は学習回数を示す。

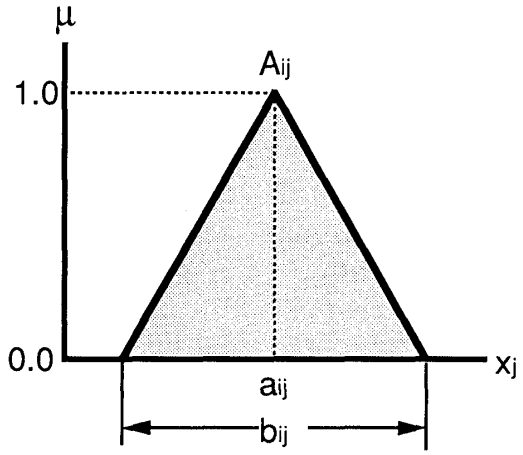


図 3 前件部のメンバーシップ関数

(4)～(7)式から $\partial E/\partial a_{ij}$, $\partial E/\partial b_{ij}$, $\partial E/\partial w_i$ を計算すると、次式のようになる。

$$\frac{\partial E}{\partial a_{ij}}=\frac{\mu_i}{\sum_{i=1}^{n}\mu_i}\cdot(y-y^r)\cdot(w_i-y)\cdot\mathrm{sgn}(x_j-a_{ij})\cdot\frac{2}{b_{ij}\cdot A_{ij}(x_j)} \tag{11}$$

$$\frac{\partial E}{\partial b_{ij}}=\frac{\mu_i}{\sum_{i=1}^{n}\mu_i}\cdot(y-y^r)\cdot(w_i-y)\cdot\frac{1-A_{ij}(x_j)}{A_{ij}(x_j)}\cdot\frac{1}{b_{ij}} \tag{12}$$

$$\frac{\partial E}{\partial w_i}=\frac{\mu_i}{\sum_{i=1}^{n}\mu_i}\cdot(y-y^r) \tag{13}$$

ただし、(11)式では $x_j\neq a_{ij}$ であり、sgn は次式の符号を表す関数を示している。

$$\mathrm{sgn}(z)=\begin{cases}-1 & ;\ z<0\\ 0 & ;\ z=0\\ 1 & ;\ z>0\end{cases} \tag{14}$$

したがって、(11)～(13)式をそれぞれ(8)～(10)式に代入することにより次式が得られる。

$$a_{ij}(t+1)=a_{ij}(t)-\frac{K_a\cdot\mu_i}{\sum_{i=1}^{n}\mu_i}\cdot(y-y^r)\cdot(w_i(t)-y)\cdot\mathrm{sgn}(x_j-a_{ij}(t))\cdot\frac{2}{b_{ij}(t)\cdot A_{ij}(x_j)} \tag{15}$$

$$b_{ij}(t+1)=b_{ij}(t)-\frac{K_b\cdot\mu_i}{\sum_{i=1}^{n}\mu_i}\cdot(y-y^r)\cdot(w_i(t)-y)\cdot\frac{1-A_{ij}(x_j)}{A_{ij}(x_j)}\cdot\frac{1}{b_{ij}(t)} \quad (16)$$

$$w_i(t+1)=w_i(t)-\frac{K_w\cdot\mu_i}{\sum_{i=1}^{n}\mu_i}\cdot(y-y^r) \quad (17)$$

(11)式において$x_j=a_{ij}$の場合、評価関数Eはa_{ij}で微分可能でない。したがって、$x_j=a_{ij}$のときは(15)式の学習を適用しないようにする必要がある。しかし、$x_j=a_{ij}$を(15)式に代入すると、$a_{ij}(t+1)=a_{ij}(t)$となり、実質的に学習は行われない。このことより、$x_j=a_{ij}$のときにも(15)式の適用が可能である。

(15)～(17)式は評価関数Eが最も小さくなる方向を探索しながらパラメータa_{ij}、b_{ij}およびw_iを変化させることを意味している。したがって、(15)～(17)式の演算を繰り返し用いることにより、評価関数Eの値を極小にするような推論ルールが求まることになる。

3.3 チューニング手順

デルタルールによる自動チューニング手法を用いて、推論ルールを逐次的に学習する手順について次に述べる。

[Step a 1]

推論ルールの初期設定を行う。前件部のメンバーシップ関数A_{ij}の中心値a_{ij}および幅b_{ij}、後件部の実数値w_iの初期値を決める。

中心値a_{ij}の初期値は入力変数x_jの定義域を等分割するように設定する。また、幅b_{ij}は各入力変数での隣りあうメンバーシップ関数の中心値の間隔よりも大きくし、メンバーシップ関数が十分に重なり合うようにする。

[Step a 2]

入出力データ$(x_1,\cdots,x_m,y^r)$を入力する。

[Step a 3]

(4)～(6)式を用いて、入力データ$x_1,\cdots,x_m$に対してファジィ推論を行い、各推論ルールの適合度μ_iと推論結果yを求める。

[Step a 4]

推論結果y、適合度μ_i、出力データy^rを用いて(17)式により後件部の実数値w_iを更新する。

[Step a 5]

[Step a 3]と同様に、再度ファジィ推論を行う。

[Step a 6]

推論結果y、適合度μ_i、出力データy^r、および[Step a 4]で更新した後件部の実数値w_iを用いて、(15)および(16)式により前件部のメンバーシップ関数の中心値a_{ij}、幅b_{ij}を更新し、[Step a 2]に戻る。

4. 数値例

次の数式で表される3種類の2入力1出力の非線形システムの同定問題に本手法を適用し、その有効性を示す。

System 1：

$$y=(2\cdot x_1+4\cdot x_2^2+0.1)^2 \quad (18)$$

System 2：

$$y=4\cdot\sin(\pi\cdot x_1)+2\cdot\cos(\pi\cdot x_2) \quad (19)$$

System 3：

$$y=(3\cdot\exp(3\cdot x_1)+2\cdot\exp(-4\cdot x_2))^{-0.5} \quad (20)$$

(18)～(20)式の入力(x_1, x_2)を乱数により$[-1,1]$の範囲で変化させ、各システムごとにそれぞれ、同定用の入出力データ$(x_1^p,\cdots,x_m^p,y^{rp})$，$p=1,\cdots,P$と評価用の入出力データ$(x_1^q,\cdots,x_m^q,y^{rq})$，$q=1,\cdots,Q$を作成する。ただし、出力データは$[0,1]$の範囲で正規化する。ここで、同定用の入出力データを同定用データ、評価用の入出力データを評価用データと呼ぶ。各データ数は、$P=20$，$Q=20$とした。

この数値例では、すべての入出力データが学習開始までに得られているので、3.3節で述べた逐次型の学習をそのまま使うことはできない。したがって、この数値例では3.3節で述べた手順を一部変更した手順にしたがって自動チューニングを行った。その学習手順を以下に示す。

[Step b 1]

[Step a 1]と同様に推論ルールの初期設定を行い、データ番号 p を 1 に初期化する。

[Step b 2]

p 番目の入出力データ $(x_1^p, \cdots, x_m^p, y^{rp})$ を読み込む。

[Step b 3]

(4)～(6)式を用いて、入力データ $x_1^p, \cdots, x_m^p$ に対してファジィ推論を行い、各推論ルールの適合度 μ_i^p と推論結果 y^p を求める。

[Step b 4]

推論結果 y^p、適合度 μ_i^p、出力データ y^{rp} を用いて(17)式により後件部の実数値 w_i を更新する。

[Step b 5]

[Step b 3]と同様に、再度ファジィ推論を行う。

[Step b 6]

推論結果 y^p、適合度 μ_i^p、出力データ y^{rp}、および[Step b 4]で更新した後件部の実数値 w_i を用いて、(15)および(16)式により前件部のメンバーシップ関数の中心値 a_{ij}、幅 b_{ij} を更新する。

[Step b 7]

データ番号 p を、同定用データの総数 P と比較し、$p<P$ ならば p を 1 増加させて[Step b 2]に戻る。$p=P$ ならば、次の[Step b 8]に進む。

[Step b 8]

次式により推論誤差 $D(t)$ を計算する。

$$D(t)=\frac{1}{P}\sum_{p=1}^{P}(y^p - y^{rp})^2 \qquad (21)$$

誤差 $D(t)$ が次式を満足する場合、学習のアルゴリズムを終了する。

$$D(t)\leqq\delta \qquad (22)$$

ただし、δ は学習のアルゴリズムを終了させるためのしきい値であり、学習開始前に設定する。(22)式が満足されない場合は、データ番号 $p=1$ とし[Step b 2]に進む。

入力 x_1, x_2 それぞれに対して、「NB, NS, PS, PB」の 4 つのメンバーシップ関数を設定し、推論ルールの総数を 16(=4^2)個として、上記したアルゴリズムを用いて同定を行った。なお、後件部の実数値の初期値は 0、アルゴリズムの終了を判定するしきい値 δ は 0.001、学習係数は $K_a=1.0$、$K_b=1.0$、$K_w=1.0$ とした。

本手法を用いた推論ルールの学習結果を表 1 および図 4 に示す。表 1 では、System 1 のデータに対して得られたファジィ推論のルールテーブルを、図 4 では、一例として表 1 のルールテーブルの中の網掛けで示した推論ルールの前件部を示す。

表 1 ルールテーブル (System 1)

		x_2			
		NB	NS	PS	PB
x_1	NB	0.2045	0.5453	0.6751	0.1347
	NS	−0.1560	−0.3217	−0.2182	−0.3294
	PS	0.5674	−0.0189	−0.1435	0.6293
	PB	0.8193	0.9597	0.6517	0.7348

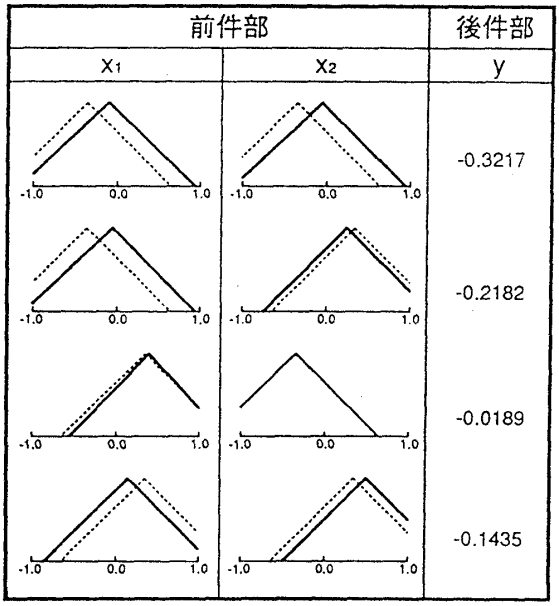


図 4 学習後の推論ルールの一例 (System 1)

図 4 中の点線は学習前の初期のメンバーシップ関数の形状を示している。表 1・図 4 よりデルタルールによる学習により、後件部の実数値と前件部のメンバーシップ関数の形状が変更されていることがわかる。

表 2 に、学習が終了するまでの学習回数と構築された推論ルールを用いて推論を行ったときの評

価用データに対する推論誤差を示す。なお、評価用データに対する推論誤差 D' は次式を用いて計算した。

$$D'=\frac{1}{Q}\sum_{q=1}^{Q}(y^q - y^{rq})^2 \qquad (23)$$

また、表2では逐次型ファジィモデリング[3),4)]、バックプロパゲーション型のニューラルネット[5)]を用いて実験を行った結果もあわせて示す。ただし、逐次型ファジィモデリングは、後件部を実数値とし推論ルール数は16個とした。また、ニューラルネットは3層で、入力層のユニットの数は2、中間層のユニット数は16、出力層のユニット数は1個とした。

表2 数値例の結果

System No.		本手法	逐次型ファジィモデリング	ニューラルネット (Backpropagation)
1	学習回数	19	4	37268
	推論誤差 D'	1.71×10^{-2}	7.28×10^{-2}	2.20×10^{-2}
2	学習回数	19	8	29894
	推論誤差 D'	3.81×10^{-3}	10.78×10^{-3}	4.04×10^{-3}
3	学習回数	15	14	25463
	推論誤差 D'	7.11×10^{-3}	63.6×10^{-3}	8.98×10^{-3}

表2より以下のことがいえる。

1)本手法の学習回数は、逐次型ファジィモデリングよりわずかに多いが、ニューラルネットと比較するとかなり少ない。このことより、本手法は、ニュールラルネットと比較して高速な学習が可能であることがわかる。

2)本手法の評価データに対する推論誤差は、ニューラルネットや逐次型ファジィモデリングより小さくなっている。これは、本手法が前件部のメンバーシップ関数を推論ルールに対して独立にチューニングするためであると考える。この結果から、本手法は高い汎化能力を持つといえる。

5. 移動ロボットの動的障害物回避問題への応用

本手法を動的障害物の回避問題に応用し、学習能力と汎化能力についてさらに議論する。

図5に動的障害物回避の概念図を示す。ここでは、まず操作者が移動ロボットを手動で制御し、その制御動作から得られた入出力データを用いて推論ルールを自動生成する。生成した推論ルールにより、移動ロボットに人間のような柔軟な状況判断能力をもたせ、障害物を回避し目標点に到達させる制御を行う。

この制御問題の入力変数と出力変数を以下に示す。

入力変数：移動ロボットと目標点の距離 x_1[m]
移動ロボットと目標点の角度 x_2[deg]
移動ロボットと障害物の距離 x_3[m]
移動ロボットと障害物の角度 x_4[deg]

出力変数：移動ロボットのハンドル角の変化量 y [deg]

ただし、入力である角度 x_2, x_4 は、図6に示すように移動ロボットの向きに左右されない絶対座標系における角度であり、東(E)方向を起点(0 deg)とし、左回りを正としている。

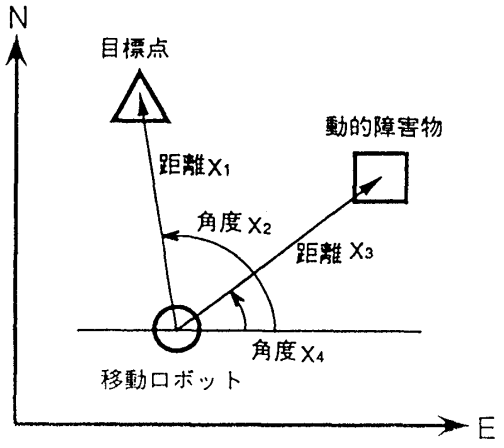


図5 障害物回避の概念図

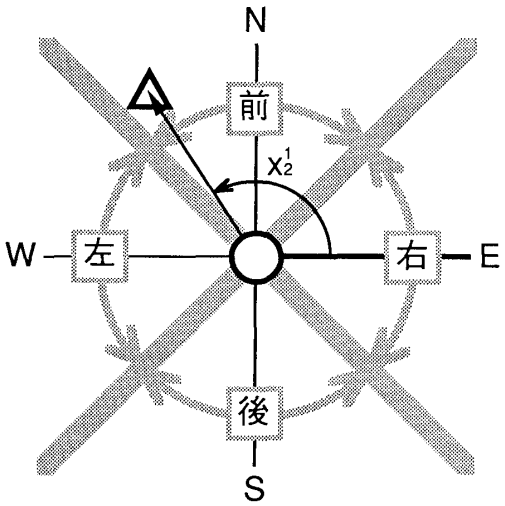


図 6 座標の設定

障害物回避の前提条件として、次のような仮定を行う。

1)移動ロボットは障害物までの距離 x_3 と角度 x_4 を検知することができる。

2)移動ロボットは1度に複数個の障害物に遭遇しない。

3)移動ロボットのハンドル角のみを制御し、駆動速度は制御しない。

4)移動ロボットの動特性は考慮しない。

5)障害物は移動するが、目標点は固定とする。

シミュレーション実験の手順を以下に示す。

[Step c 1] 推論ルールの初期設定を行った。図7のように、各入力変数に対して、5種類のメンバーシップ関数を初期設定し、推論ルールを合計 $5^4=625$ 個とした。各メンバーシップ関数のファジィラベルとして、入力変数 x_1, x_3 については、「かなり近い、近い、やや遠い、遠い、かなり遠い」の5種類を設定し、入力変数 x_2, x_4 については「左、後、右、前、左」の5種類とした。入力変数 x_2, x_4 に関する初期のメンバーシップ関数は、図6のように、北向き(N)が「前」、東向き(E)が「右」、西向き(W)が「左」、南向き(S)が「後」に対応する。たとえば、図6のように移動ロボットに対して角度 x_2^1 で目標点がある場合は、x_2^1 は図7の「前」と「左」というファジィラベルを持つ初期メンバーシップ関数にそれぞれ適合する。

また、入力変数 x_1, x_3 の定義域は $[-1,1]$ に正規化してあり、制御量のハンドル角は左回りの方向を正としている。後件部の実数値の初期値は0とした。

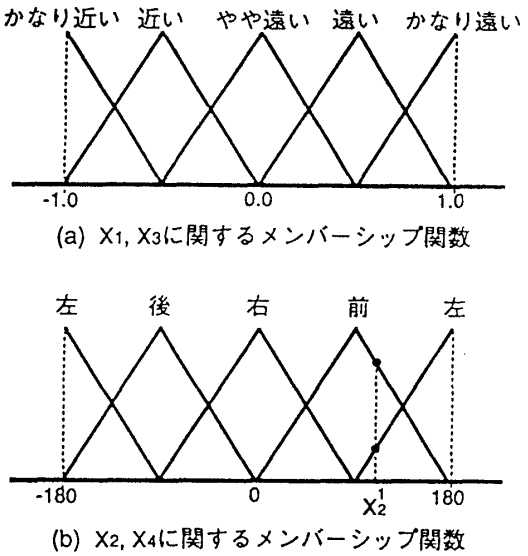


図 7 初期メンバーシップ関数

[Step c 2] 実験者は、手動コントローラを用いて移動ロボットのハンドル角を制御し、障害物を回避させる。この時、入力値(x_1、x_2、x_3、x_4)と実験者が入力したハンドル角の変化量 y^r を観測し入出力データを作成する。ここでは、上記の回避動作を2回繰り返し、66個の同定用データを得た。図8にここで教示させた移動ロボットの軌跡を示す。ただし、図中の○は移動ロボットを、□は障害物を表している。

[Step c 3] [Step b 2]~[Step b 8]を用いて推論ルールのチューニングを行った。ただし、ここではメンバーシップ関数の中心値 a_{ij} は固定とし、学習回数は10回とした。

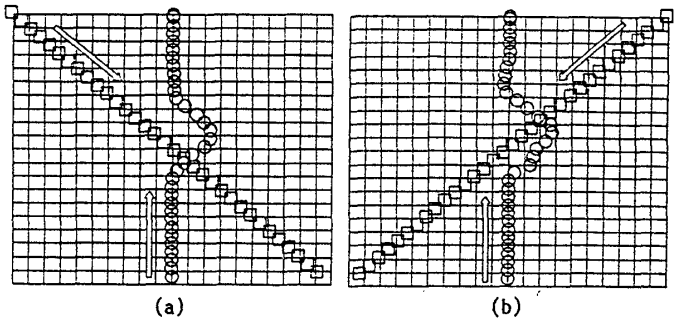


図 8 教示した移動ロボットの軌跡

チューニング後の推論ルールの一例を図9に示す。チューニングされた推論ルールは625個の推論ルールの内40個であった。図9では、チューニングされた推論ルール40個の内の6個を示している。図9のメンバーシップ関数につけられたファジィラベルは、チューニング前の初期のメンバーシップ関数に対してつけたファジィラベルを表している。例えば、図9のNo.1の推論ルールは、「もし、移動ロボットが目標点に近く、目標点が前方にあり、障害物の近くに位置し、障害物が移動ロボットの前方にあるならば、ハンドル角を39.9度右方向へ増やす。」を意味している。

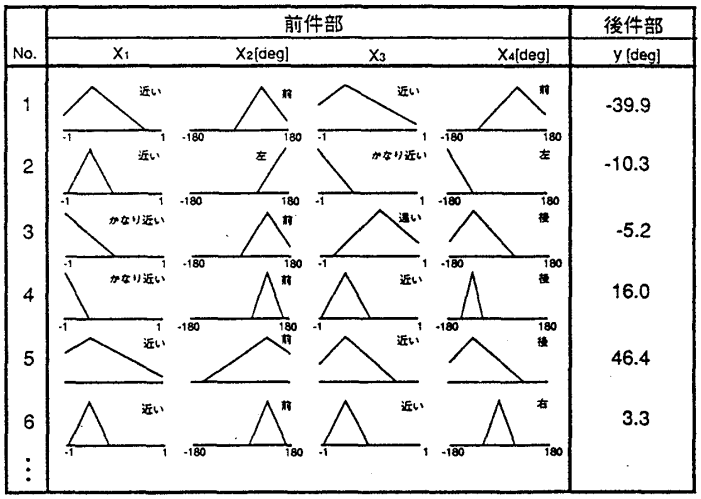


図9 獲得した推論ルールの一例

これらのチューニングされた推論ルールを用いて移動ロボットを制御させた結果を図10に示す。

ただし、図10(a')～(e')は、それぞれ以下のような制御結果を示す。

(a')(b') ：教示したときと同じ軌跡で障害物を動かした場合

(c') ：教示した軌跡よりも急な角度で障害物を動かした場合

(d') ：教示した軌跡よりもなだらかな角度で障害物を動かした場合

(e') ：教示した軌跡と逆方向に障害物を動かした場合

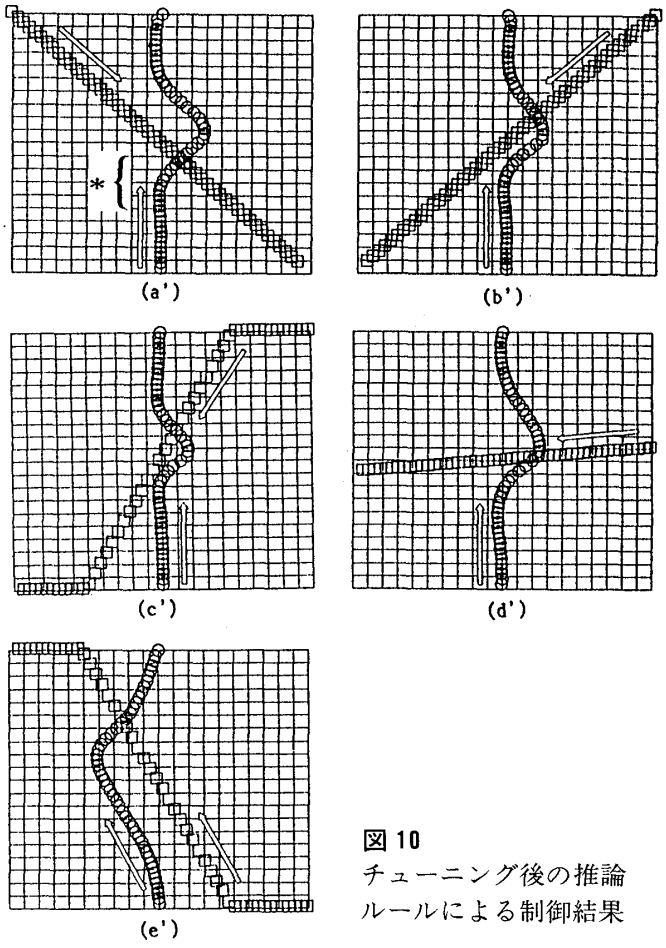


図10
チューニング後の推論ルールによる制御結果

図10(a')(b')では、移動ロボットは教示した軌跡よりなめらかな軌道で動いている。また、図10(c')(d')(e')の教示した軌跡と異なった軌跡で障害物を動かした場合でも、移動ロボットは障害物と衝突せずに目標点に到達している。このように、本手法では、教示したデータに対してだけでなく、教示していないデータに対しても、当初の目標を達成する制御が可能である。この結果から、本手法は高い汎化能力を持つと考える。

また、図9の推論ルールNo.1は、例えば図10(a')の＊印で示すような場合に適合する。図9のNo.2～6の推論ルールも、教示した走行軌跡の特徴をよく表している。以上のように、本手法は高い汎化能力を有しているだけでなく、獲得した知識の概要を人間にわかりやすい形で提示することもできる。したがって、システムの設計者は推論ルールの適合範囲や妥当性を検討することができる。

6．おわりに

本論文では、ニューラルネットの学習則であるデルタルールを用いたファジィ推論の自動チューニング手法を提案した。本手法は、推論ルールごとに独立に設定した前件部のメンバーシップ関数

と後件部の実数値をデルタルールにより最適化する。本手法の有効性を示すため、数値例と動的障害物回避問題に適用した。その結果、本手法は高い汎化能力を持つ推論ルールを短時間で構築でき、獲得した推論ルールを人間にわかりやすい形で表示できることを示した。

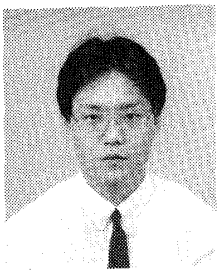
参考文献

1) 廣田，寺野：特集／ファジィ制御，コンピュートロール，No.29，(1989)
2) 林，野村，若見：ニューラルネット駆動型ファジィ推論による推論ルールの獲得，日本ファジィ学会誌，Vol.2，No.4，pp.133-145，(1990)
3) 市橋，渡辺：簡略ファジィ推論を用いたファジィモデルによる学習型制御，日本ファジィ学会誌，Vol.2，No.3，pp.429-437，(1990)
4) 渡辺，市橋：逆キネマティクス－逆ダイナミクスモデルを学習するマニュピレータのファジィ制御，第6回ファジィシステムシンポジウム講演論文集，pp.535-538，(1990)
5) Rumelhurt,D.E., Hinton,G.E., Willoams,R.J.: Learning internal representations by error propagation, Parallel distributed processing, pp.318-362, (1986)
6) 前田，村上：自己調整ファジィコントローラ，計測自動制御学会論文集，24-2，pp.191-197，(1988)
7) 市橋，田中：PIDとFuzzyのハイブリッド型コントローラー，第4回ファジィシステムシンポシウム講演論文集，pp.97-102，(1988)
8) F. Rosenblatt：The perceptron：A probabilistic model for information sorage and organization in the brain, Psychol. Rev. 65[6]，pp.386-408，(1958)
9) S.Amari：A theory of adaptive pattern classifiers, IEEE Trans. EC-16，pp.279-307 (1967)
10) 渡辺，市橋：n次メンバーシップ関数を用いた逐次ファジィモデリングとそのクレーン制御への応用，日本ファジィ学会誌，Vol.3，No.2，pp.347-356，(1991)
11) 前田，竹垣：ファジィ推論を用いた移動ロボットの動的障害物回避，日本ロボット学会誌，Vol.6，No.6，pp.50-54，(1988)

(1991年6月15日　受付)
(1991年10月22日　再受付)

[問い合わせ先]
〒570 大阪府守口市八雲中町3-15
松下電器産業㈱中央研究所
野村 博義　TEL：06-906-4849
FAX：06-904-7252
E-Mail： nomura@crl.mei.co.jp

著者紹介



野村 博義（のむら ひろよし）

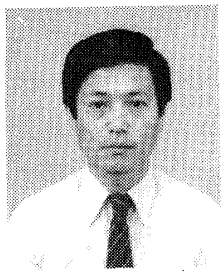
松下電器産業(株)中央研究所 電子機器基礎研究所　第6研究室

1986年九州工業大学・工学部・制御工学科を卒業、同年　松下電器産業(株)に入社、現在　中央研究所所属。主として、ファジィ制御、ファジィエキスパートシェル、ファジィ推論とニューラルネットの融合研究に従事。システム制御情報学会などの会員。

林　勲（はやし いさお）

松下電器産業㈱中央研究所 電子機器基礎研究所　第6研究室

1981年大阪府立大学工学部経営工学科卒業、1985年大阪府立大学大学院(経営工学専攻)修了。1987年松下電器産業㈱に入社、現在、中央研究所所属。主として、ファジィ推論とニューラルネットワークとの融合研究、ファジィ検索の研究に従事。1991年電気関係学会関西支部連合大会講演会奨励賞授賞。工学博士。人工知能学会、電気学会、日本ファジィ学会、国際ファジィシステム学会などの会員。



若見 昇（わかみ のぼる）

松下電器産業(株)中央研究所 電子機器基礎研究所　第6研究室

1971年大阪大学・基礎工学部・制御工学科卒業、同年、松下電器産業(株)入社、現在中央研究所　電子機器基礎研究所　第6研究室室長。ファジィ理論とその応用に従事。工学博士。日本ファジィ学会、情報処理学会、システム制御情報学会などの会員。