

距離型ファジィ推論法に基づく  
案内ロボットの軌道計画法のシミュレーション

高知工科大学

知能機械システム工学科

知能ロボティクス研究室

学籍番号 1030093

氏名 今田浩喜

# 目次

第一章	緒言	1
1.1	はじめに	1
1.2	本研究について	1
1.3	システム制御における軌道計画法	2
第二章	ファジィ推論法	4
2.1	ファジィとは	4
2.2	ファジィ集合	8
2.3	ファジィ推論法	17
2.3.1	ファジィ推論	17
2.3.2	min-max 重心法	18
2.3.3	簡略推論法	22
2.4	距離型ファジィ推論法とその特徴	24
第三章	距離型ファジィ推論法に基づく軌道計画法	44
3.1	案内ロボットのファジィ軌道計画法	44
3.2	ファジィ軌道計画の導入	44
3.3	軌道計画法の設計手順	46
3.4	言語ルールによる知識の抽出	47
3.5	知識の定式化	48
3.6	ファジィ集合による言語変数の定量化	49
3.7	軌道計画のアルゴリズム	52

---

第四章	軌道計画のシミュレーション	54
4.1	推論エンジンの特徴	54
4.2	シミュレーション内容	56
4.3	シミュレーション結果	57
4.4	考察	62
第五章	結言	63
5.1	本研究のまとめ	63
5.2	今後の課題	63
	参考文献	65
	謝辞	66

# 第一章 緒言

## 1.1 はじめに

移動ロボットの経路計画や軌道追従制御は盛んに研究されている。例えば、室内蛍光灯情報など環境設備を積極的に利用する移動ロボット、障害物回避を目的とするステレオビジョンセンサの利用法などが提案されている。これらの研究はロボットの対象となる分野が生産工場などの産業分野であり、如何に正確に作業を行うかといった高精度と、如何に早く作業を行うかといった高速度が望まれてきた。一方、最近ロボットはその活躍の場を産業分野から医療福祉分野、人間生活の場に拡大している。両者の大きな違いは、前者がシステムの中で人間を除外しているのに対して、後者はシステムの中で人間を対象にする点である。こうした背景下ではロボットと人間との間に生まれる相互関係の重要性が提起され、ロボットの動きによる感情表現法や感情自動獲得支援システムが提案されている。

## 1.2 本研究目的と意義

近年ロボットは活躍の場を産業分野から医療分野、人間生活に場に拡大している。したがって、人間と同じ空間で作業するロボット

においては，人間の意志，感情，行動に対しての理解，ロボットの動きによる感情表現，安全性などの人間との相互関係が重要になってくる．建物内での人間を誘導できる案内ロボットを開発する事を目的として，距離型ファジィ推論法を用いて人間との相互関係を平行移動時のみを考えてシミュレーションを行う事を目的とする．

### 1.3 トータルシステム制御における軌道計画法

図 1.1 にトータルシステムの構成を示す．

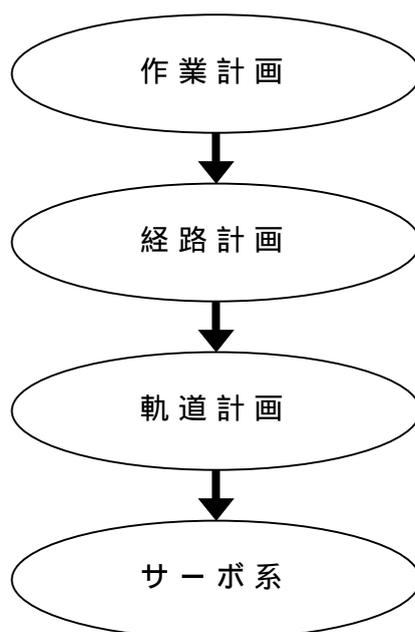


図 1.1 トータルシステムの構成

経路とはロボットの構造に関連した指定点が移動することによって描く空間的軌跡であり，軌道とはロボットの構造に関連した指定点が移動することによって描く時間的空間的軌跡と定義されている．経路計画，軌道計画とは様々な観点から与えられた情報をもとに経

---

路,軌道を決定することである.ここでこれらの計画を説明するために1つの例をあげる.ある人が月に行く計画を立てた場合を考える.まず,何かの目的のために月に飛び,月のどこかで着陸することを決定する.これは作業計画である.次に,地球のどこの場所からいつロケットで出発すれば良いか,そしてどのような経路を経て月に接近するのが良いか,また月のどこの場所に着陸すれば良いかなどの種々の観点から検討する.これが経路計画である.経路が決まったがロケットをいかなるスピードで飛ばす必要があるであろうか,地球の引力に逆らってロケットが地球の重力圏を離脱するためには決められたスピードで飛ばすことが必要である.その後,燃料を節約するためにロケットを最適なスピードで飛ばすことが要求される.そして,月に着陸するために当然スピードを徐々に下げなければならない.このようなロケットのスピードを状況に応じて変化させるような計画が必要である.これを軌道計画という.

## 第二章 ファジィ推論法

### 2.1 ファジィとは

ファジィは、「曖昧なもの」「はっきりしないもの」という風に理解されがちだが、実際には二値では表現できないものを精密に表現する事の出来るとても優れた表現法である。ファジィを使った例として、「もし温度が 30 以上ならば、クーラーを強めよ」というような制御があるとし、図 2.1 に示すように縦軸に評価値、横軸に温度をとったグラフを考える。ここで、もし温度が 29.9 だとする。29.9 というのは限りなく 30 に近いと言える。しかし二値の表現で判断する場合、グラフから分かるようにまだ 30 に達していない為、この例は無視される。対してファジィで表現した場合、図 2.1 から分かるように 30 以上であるという事に対して、評価値 0.9 ぐらいの可能性が残される。これはどういう事かと言うと、人間に近い評価がファジィには可能なのである。ファジィは少し値が大きい、ほんの少し値が小さい、という風に漠然としたデータでも取り扱うことが出来る。

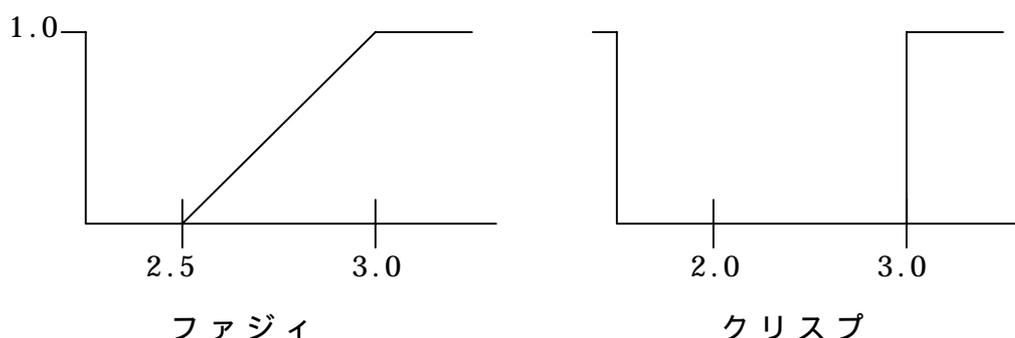


図 2.1 ファジィとクリスプ

つまりファジィ推論というのは，変数に曖昧さを持たせる事により 1 つの規則の守備範囲に広がりを持たせる事ができる訳である．これにより，事実がどのように変化しても各規則の条件部がぼやけているので，部分的に適合する規則が必ず存在する．しかも，事実の変化に沿った適切な結論を導き出す事ができる．

これらのことから，ファジィ推論は限られた知識の中からあらゆる事実に対して結論を導き出せる良いモデルであり，その枠組みは人間の持つある種の表層的な推論機能をモデル化するのに良い近似手段を与えるものだという事ができる．

ファジィ制御の最大の特徴は if-then 型のファジィ制御規則に基づいて制御を行っている点である．また，エキスパート的な制御を実現するという点もある．ここでエキスパート的とは制御対象のオペレータの操作を模範するという意味を含んでいる．つまり，制御対象のモデルを必要としない制御といえる．もちろん制御対象のモデルに基づく制御もできる．これらの点を踏まえた上でファジィ制御の特徴を以下にあげる．

1. あいまいさを含んだ言葉で制御方策を記述されること．

従来の制御ではもちろん制御方策は線形，非線形を問わず，関数によって記述されている．そのためその

中に現れるパラメータは誤差などの不確定要素を含む。しかし、これは言語に見られるようなあいまいさを含んでいない。

2. 制御方策が従来の制御のように単一の関数型ではなく、if-then 型の論理型である複数の制御規則で記述されていること。

従来の制御において、制御方策は制御装置のモデルの基づいて一つの式で統一的に表現されるが、ファジィ制御では複数の制御規則で分散的に表現されている。そのためファジィ制御系の実現を容易にする反面、解析を著しく困難にしている。

3. 制御に使われる制御装置から入力情報が、PID 制御のように、一つの関数にまとめて代入することによって制御装置の操作量を求めるのではなく、複数の規則に分散的に当てはめ、並列的なファジィ推論によって操作量を求めていること。

同時に複数の制御規則を適用することは計算を複雑にし、さらに規則の適用結果を統合するとき規則間の矛盾、競合といった問題を引き起こす可能性がある。しかし、計算の複雑化の問題はファジィ推論によって解消される。また、ハードウェアの発達によってもこの問題は解消される。規則間の矛盾、競合の問題はファジィ推論の効用によって複数の制御規則の協調という効果を生み出している。

4. 制御方策は、画一的な評価関数を定めて、制御対象のモデルに基づいて導かれているのではなく、多目的なものであること。

従来の典型的な最適制御においては制御の目的は一

つの評価関数にまとめる必要がある。この時、目的が複雑である場合は、それぞれの目的に応じて重みをつけて、これを線形に結合する。このことはユーザーの意図に沿った多目的な制御の実現を困難にしている。また、制御目的は制御対象のモデル内の変数を用いて明示的に記述しなければならない。一方、ファジィ制御では複数の目的を意識した上でそれぞれに応じた複数の制御規則を記述することによって多目的な制御が実現できる。例えば、乗り物の乗り心地を良くするために速度の変化の回数を小さくしたいという要求があった場合、従来の制御では乗り物の速度変化の回数を評価関数にした上で、理論的に最小化アルゴリズムを求めて実現しようとする。しかし、乗り物の速度変化の回数というものはモデルの変数を用いて定式化できないのでこのような制御はもともと不可能である。一方、ファジィ制御では、乗り物の速度変化の回数を小さくできると考えられる現実的な理論を組み立てた上で、これに必要な入力情報を集めて、制御規則として用いることで実現しようとする。

5. 制御に使われる情報は、物理的に計測可能な制御装置の状態量、制御偏差などの画一的なものではなく、評価可能な多様な情報であること。

従来の制御ではPID制御に見られるような関数型であるため、制御に用いられる入力情報はオフライン計測できる物理量で、さらに数値として表現できるものに限定されている。この入力情報のほとんどはフィードバックできる制御対象の出力情報や状態量である。これに対してファジィ制御ではエキスパートの視覚によ

る観察結果などの数値化が困難であるようなものも利用することができる。つまり、制御器が if-then 型であるので、前件部の if の条件部には何を記述しても良い。このことより言語による制御状態の判断、制御対象の出力情報の質的評価、画像情報など、多様な情報を効果的に利用することができる。

## 2.2 ファジィ集合

ファジィ集合とはある要素がその集合に属している度合いが 0 と 1 の間の値をとる集合のことで、言葉のあいまいさを定量的に表現するときに用いられる。一方、従来の集合のようにある要素がその集合に属している度合いが 0 か 1 のどちらかである、つまり、属しているか、いないかがはっきりした集合のことをクリस्प集合という。図 2.3 にクリस्प集合、図 2.4 にファジィ集合を示す。

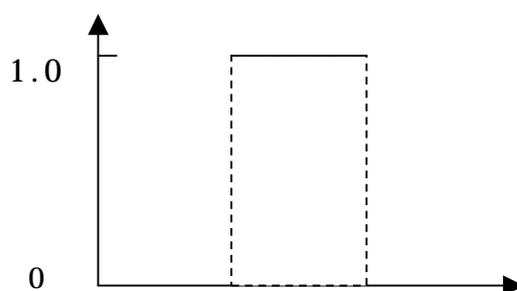


図 2.3 クリस्प集合

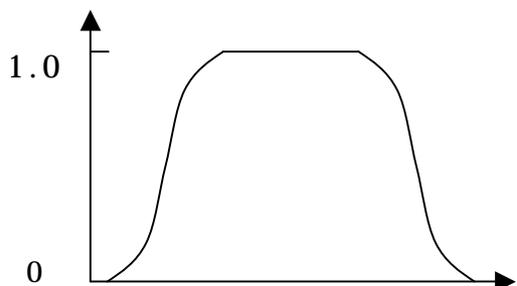


図 2.4 ファジィ集合

例えば、「1より非常に大きい数字の集まり」や「美人の集まり」、  
「背の高い人の集まり」などは集合の定義に当てはまらない。これはある対象をもってきたときに、それが果たしてその集合に属しているかいないかが明確に定まらないからである。つまり、これらの「集まり」は、いいかえると境界がはっきりしない集合のようなものであり、現実では集合のように扱われているが集合ではない。今、集合と集合のようなものとの違いを「年齢15歳以上30歳以下の人の集まり」と「若者の集合」を例にあげて述べる。全体集合  $X$  として、ある人々の集団を例えば  $X = \{A \text{ さん}, B \text{ さん}, C \text{ さん}, D \text{ さん}, E \text{ さん}, \dots\}$  とする。そして、この集団に属する人の年齢は表 2.1 とする。

表 2.1

名前	A さん	B さん	C さん	D さん	E さん	...
年齢	25	28	12	17	32	...

今、年齢15歳以上30歳以下の条件を満たしているかどうかは表 2.1 から判断すれば条件を満たしているかは容易に判断できる。つまり、どの個人が「年齢15歳以上30歳以下の人の集まり」に属しているかがすぐに判断できる。この集団では  $A$  は、 $A = \{A \text{ さん}, B \text{ さん}, D \text{ さん}, \dots\}$  というふうに、境界がはっきりした集団として表せる。 $A$  は具体的な人間の集まりが実在している。一方、「若者の集まり」といった「集まり」の場合には、判断の容易ではない。これはある人々の集団  $X$  において、その中の誰が「若者の集まり」に属しているかいないかが明確ではないである。あいまいなイメージとして「若者の集まり」という「集まり」は存在しているが、具体的な形で実在するものとして記述できない。こうした「集まり」は、非存在のままでは数学的に取り扱うことができない。数学的に取り

扱えるようにするためには「集まり」を実在化する必要がある。この役割を果たすのがメンバーシップ関数である。ここでメンバーシップ関数を説明するために一般的な集合の話をする。全体集合  $X$  の一般的な意味での部分集合を、クリスプ集合と呼ばれている。クリスプ集合では、 $x$  がその集合に属しているかいないかが一義的に決定される。つまり、クリスプ集合  $A$  については、点  $x$  がに属しているかいないかが  $x$  の関数として決まっている。ここで  $x$  の関数

$$\varphi_A(x): X \longrightarrow \{0,1\} \quad (2.1)$$

を

$$\varphi_A(x) = \begin{cases} 1, & x \in A \\ 0, & x \notin A \end{cases} \quad (2.2)$$

によって定義すると、

$$x \in A \Leftrightarrow \varphi_A(x) = 1 \quad (2.3)$$

$$x \notin A \Leftrightarrow \varphi_A(x) = 0 \quad (2.4)$$

という関係を満たす。つまり、 $\varphi_A(x)$  は集合を表現する関数になっている。ここで、 $\varphi_A(x)$  は  $A$  の特性関数もしくは定義関数と呼ばれている。 $X$  の任意の与えられたクリスプ集合に対しては一義的に特性関数が定まる。逆に、 $X$  上の 1 と 0 の 2 値だけをとる関数が与えられると、それに対して  $X$  の集合

$$\{x \in X \mid \varphi_A(x) = 1\} \quad (2.5)$$

が一意に定まり、関数  $\varphi_A(x)$  はこの集合の特性関数になる。つまり、集合と特性関数は 1 対 1 に対応している。このように集合と特性関数が 1 対 1 に対応している場合には、それらをたがいに同一視したり、集合とすべきところを特性関数で代用したり、逆に特性関数とすべきところを集合で代用しても問題はない。つまり、集合は集合であると同時に特性関数であるといえ、逆に特性関数は集合であるともいえる。例えば、2 つの集合  $A$  と  $B$  の和集合を、集合によって

$$A \cup B = \{x \mid x \in A \text{ または } x \in B\} \quad (2.6)$$

と定義しても，特性関数を使って

$$\begin{aligned}\varphi_{A \cup B}(x) &= \varphi_A(x) \vee \varphi_B(x) \\ &= \max\{\varphi_A(x), \varphi_B(x)\}\end{aligned}\quad (2.7)$$

と定義しても，定義の中身は変わらない．記述方法が違ってこれと同じことを定義している．このように集合とその特性関数とは，同じものを別の角度からみて，別の言葉で呼んでいるだけであるから特性関数  $\varphi_A(x)$  をそのまま集合と呼んでも差し支えない．この考え方に基づいて，ファジィ集合の定義をする．ただし，ファジィ集合の場合は特性関数の役割を果たすものを特性関数ではなく，先ほど述べたメンバーシップ関数という．集合の延長上にあるファジィ集合と特性関数の延長上にあるメンバーシップ関数との間にファジィ集合と特性関数との関係を考慮してメンバーシップ関数をそのままファジィ集合と呼んでも差し支えないとする．このことより次の定義を得る．

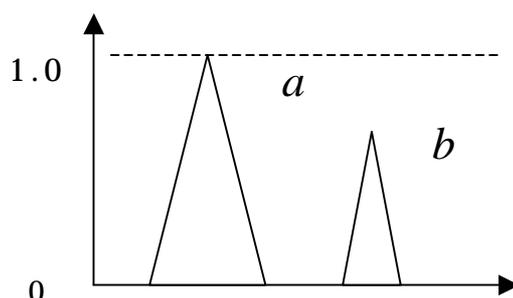
定義 1: 空間  $X$  から  $[0,1]$  への関数を， $X$  上のメンバーシップ関数という．メンバーシップ関数を同一視してファジィ集合という．メンバーシップ関数  $\mu_A(x)$  と同一視されるファジィ集合を  $A$  で表す．このとき，メンバーシップ関数によって特性づけられるという．

$\mu_A(x)=1$  を満たす  $x$  が存在するとき，つまり，

$$\max_{x \in X} \mu_A(x) = 1 \quad (2.8)$$

であるとき，メンバーシップ関数  $\mu_A(x)$  あるいはファジィ集合  $A$  は正規であるという．また，そうでないときは非正規であるという．

定義 2:  $X$  のファジィ部分集合の全体をファジィ集合族といい， $F(X)$  と記述する．

図 2.5 正規なファジィ集合  $a$  と非正規なファジィ集合  $b$ 

通常の場合では  $x$  が  $A$  に属するかしないかは一義的に定まり、その特性関数  $\varphi_A(x)$  は、 $x$  が  $A$  に属する場合には 1、 $x$  が  $A$  に属さない場合は 0 をとる。それに対して、ファジィ集合では、 $x$  が  $A$  に属するかしないかは一義的に定まらない。このことは、 $X$  上におけるメンバーシップ関数値  $\mu_A(x)$  が 1 と 0 以外の値をとるということを意味している。これを真理値を使っていいかえれば、 $x$  が  $A$  に属するという命題は、 $A$  が通常の場合の場合は真と偽という 2 つの真理値だけをとるが、ファジィ集合の場合は真偽とは異なった真理値をとる。今、この真理値に不明が当てられたとする。真理値を数値で表す場合には、真には 1、偽には 0 があてられ、そして不明には 0.5 が当てられるとする。もし、 $x$  の  $A$  への属し方が不明以外に何種類か考えられる場合にはもっと別の真理値をあてればよいと考えられる。ファジィ集合については、 $x$  が  $A$  に属するという命題の真理値として真と偽以外に 1 つあるいはいくつかの真理値があって、それらの真理値は真と偽の間のどこかに位置する。ただし、それらの真理値が真と偽との間の実数値で表せるとは必ずしもいえない。全ての真理値が常に比較可能で、大小の順に並べられるとは限らないからである。例えば、真と偽を含む 5 つの真理値を、真を最大値、偽を最小値として配置するやり方には全部で 5 種類があることはよ

く知られている．もともと，真を 1 で，偽を 0 で表すことにも，数で表すこと自体にも必然性がない．とにかく異なった記号を真と偽に割り当てて，両者を区別できればよい．よって，1 と 0 としても，真と偽を英語の頭文字からととしてもよい．真理値で 1 と 0 をとるのはわれわれの感覚にあるからであろう．こうした自由性は，広い意味でのファジィ集合を考える際に面白い役割を果たす．値域としては，いくつかの基本的な要件を満たしてさえいけば，その最小値を 0 で，最大値を 1 で表すどのような集合をもってきても，それなりのファジィ集合は構成できるのである．これにより，いろいろな集合を値域として選んでくることによって，いろいろなファジィ集合が得られる．

ところで，ファジィ集合  $A$  はメンバーシップ関数  $\mu_A(x)$  で特性づけられると定義したが，このように，ファジィ集合  $A$  は，一方では，関数型で形式的に定められるだけで，別に何らかの意味が付与されているわけではないが，一方で，「若者の集まり」とか「背の高い人の集まり」などを集合的に表したものという一面もある．つまり，ファジィ集合には形式と内容の二面性がある．例えば，年齢に関するファジィ集合を取り上げる．ファジィ集合が定められる空間としては  $X$  と  $T$  の 2 つがあげられる． $X$  は全体集合であり，メンバーシップ関数はこの空間で定義される．つまり，ファジィ集合の形式を定める．年齢の場合には  $(0, +\infty)$  としてもよいだろう．一方で， $T$  は語の空間であり，ファジィ集合の意味を規定する．例えば， $T = \{\text{非常に若い}, \text{若い}, \text{中年である}, \text{熟年である}, \text{老年である}, \dots\}$  とする．ある語  $t \in T$  を表すファジィ集合を  $A$  とすると， $A$  は  $T$  から  $F(X)$  への写像  $M$  によって

$$M : t \rightarrow A = M(t) \quad (2.9)$$

と表される．このとき

1.  $t$  をファジィ集合  $A$  のラベルと呼ぶ．

## 2. ファジィ集合 $A$ を $t$ の意味と呼ぶ .

この写像  $M$  によって、言葉とファジィ集合が結び付けられるのだが、 $M$  をどう決定するか、いいかえると、どのファジィ集合をどの語に結びつけるかが大きな問題になってくる。逆に、われわれがそれぞれの語に対して漠然と持っている集合的なイメージをどのようにメンバーシップ関数として定着するかといってもよい。例えば、 $t = \text{若者}$  に対するファジィ集合  $A = M(\text{若者})$  はどういうものになるだろうか。どのようなメンバーシップ関数をとってくればファジィ集合「若者の集まり」をもっともよく表現できるかについてはいろいろと議論されてきた。このファジィ集合とメンバーシップ関数との関係はファジィ理論の創始期からの問題であり、現在でもファジィ集合を一義的に表すメンバーシップ関数は存在しない。メンバーシップ関数はそれぞれが主観によって自由に決定してよいというように考えられている。今日までのファジィ理論の進展で見ると、限りにおいてはメンバーシップ関数を主観によって自由に決定してもそれほど問題ではなかった。このことは、ファジィ理論が応用、それもファジィ制御主導で理論が展開してきたという事実が関係している。ファジィ制御においては、1回1回の制御出力が、不適当なメンバーシップ関数が原因で少々最適出力からはずれていても、こまめなフィードバックと計算の速さで十分カバーでき、短時間に最適な制御に到達する。こう点では、メンバーシップ関数の形状はそれほど重要ではなかった。一方、今後応用の範囲が広がって、あるいはもっと高速の制御が要求されるようになれば、メンバーシップ関数の形が問われる可能性もある。また、メンバーシップ関数をそれぞれの主観で決めてもよいといっても、その主観があまりにも普遍的なものから遠ければ、その結果作られたシステムに普遍性が欠け、使い物にならないおそれもある。つまり、メンバーシップ関数を決定する主観にもある程度の普遍性、客観性が要求される。ここで多く

の人に共有する普遍性を持ったメンバーシップ関数を取り出す決定方法は有益な手段であるといえる。

ここでいくつかのファジィ集合の基本形を紹介する。

### 1. 三角型ファジィ集合

図 2.6 で示されるようなファジィ集合を三角型ファジィ集合という。

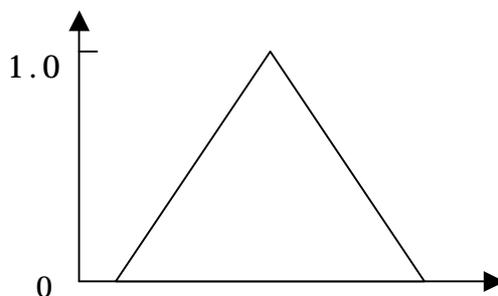


図 2.6 三角型ファジィ集合

三角型メンバーシップ関数の基本形は次の式 (2.10) で表される。

$$T(t) = \begin{cases} 0, & t \leq -1 \\ t+1, & -1 \leq t \leq 0 \\ 1-t, & 0 \leq t \leq 1 \\ 0, & 1 \leq t \end{cases} \quad (2.10)$$

### 2. 台形型ファジィ集合

図 2.7 で示されるようなファジィ集合を台形型ファジィ集合という。

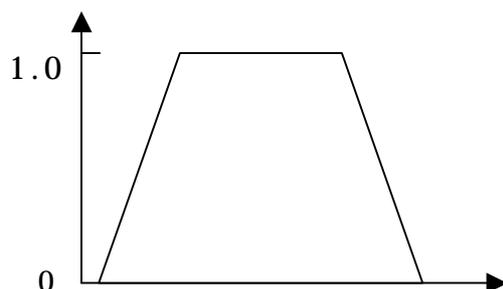


図 2.7 台形型ファジィ集合

台形型メンバーシップ関数の基本形は次の式(2.12)で表される。

$$T(t) = \begin{cases} 0, & t \leq -2 \\ t+1, & -2 \leq t \leq -1 \\ 1, & -1 \leq t \leq 1 \\ 1-t, & 1 \leq t \leq 2 \\ 0, & 2 \leq t \end{cases} \quad (2.11)$$

### 3. 型ファジィ集合

図 2.8 で示されるようなファジィ集合を 型ファジィ集合という。

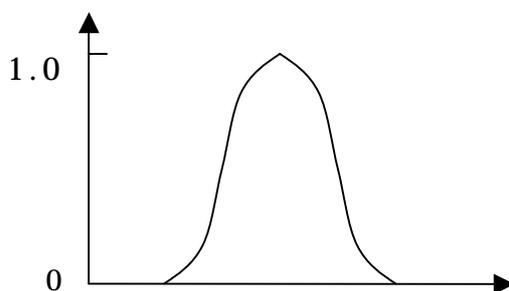


図 2.8 型ファジィ集合

型メンバーシップ関数の基本形は次の式(2.12) ,

(2.13) , (2.14) で表される .

$$\Pi(t) = \begin{cases} S(t+1), t \leq 0 \\ Z(t), 0 \leq t \end{cases} \quad (2.12)$$

$$S(t) = \begin{cases} 2t^2, 0 \leq t \leq 0.5 \\ 1 - 2(1-t)^2, 0.5 \leq t \leq 1 \end{cases} \quad (2.13)$$

$$Z(t) = S(1-t), 0 \leq t \leq 1 \quad (2.14)$$

## 2.3 ファジィ推論法

### 2.3.1 ファジィ推論

ファジィ推論とは , 入力情報とファジィ集合をつきあわせて必要な制御操作量を決定することである . 一般的にファジィ制御で用いられるファジィ推論は , ファジィ論理に基づくファジィ推論法よりも簡略化された推論法を用いる . これは以下の理由が挙げられる .

1. ファジィ推論のための入力情報が一般にファジィ集合ではなく , 実数値である .
2. ファジィ制御ではファジィ推論の計算速度を速める必要性がある .

このためファジィ制御のためのファジィ推論法としては様々な方法が提案されている .

ファジィ推論法の方法には , 一般的に直接法と間接法がある . 直接法はファジィ命題の真理値として , メンバシップ関数の値を直接用いて推論する方法である . この推論法は非常に理解しやすく , 従来のファジィ制御において最も扱われてきた .

直接法の代表的なファジィ推論法には以下の推論法が挙げられる .

1. min-max 重心法
3. 簡略推論法

これらの推論法についてはそれぞれ 2.3.2, 2.3.3 で述べる。

一方, 間接法はファジィ命題の真理値として, ファジィ真理値を用いて推論する方法である。一般的に間接法は推論手順が複雑であるため, ファジィ制御ではあまり使われない。間接法では, 制御規則そのものの真らしさを区別して取り扱っているため, ファジィ制御での応用の可能性を持っている。間接法については 2.3.5 で述べる。

### 2.3.2 min-max 重心法

この推論法は推論手順が理解しやすく, 今日までのファジィ制御で最も広く使われている。この推論法は, 事実が与えられて推論結果を求めるまでに, min 演算, max 演算, 重心法を用いているために min-max 重心法と呼ばれている。

次に min-max 重心法の方法について述べる。

まず最初に表 2.2 のような推論規則を与えたとする。

表 2.2 推論規則

$R^1$	: IF $x_1$ is $A_1^1$ and $x_2$ is $A_2^1$ then $y$ is $B^1$
$R^2$	: IF $x_1$ is $A_1^2$ and $x_2$ is $A_2^2$ then $y$ is $B^2$
事実	: $x_1$ is $A_1$ and $x_2$ is $A_2$
推論結果	: $y$ is $y_0$

ここで,  $x_1, x_2$  は前件部変数といわれるもので制御対象の状態変数であり, ファジィ制御装置の入力変数である。  $y$  は後件部変数といわれるもので制御対象の操作量であり, ファジィ制御装置の出力変数である。また,  $A_1^1, A_2^1, A_1^2, A_2^2$  は前件部の言語変数であり,

それぞれ前件部変数の全体集合  $X_1, X_2$  におけるファジィ集合であらわされる。一方,  $B^1, B^2$  は後件部の言語変数であり, それぞれ後件部変数の全体集合  $Y$  におけるファジィ集合であらわされる。事実とはファジィ制御装置の入力情報である。ここでは前件部変数  $x_1, x_2$  に対して事実  $A_1, A_2$  が与えられたことになる。また, この事実  $A_1, A_2$  は前件部変数の全体集合  $X_1, X_2$  におけるファジィ集合であらわされる。推論結果はある事実が与えられたときにその事実と推論規則を照らしあわせて求められる出力情報である。ここでは後件部変数  $y$  に対して推論結果  $y_0$  が与えられている。また, この推論結果  $y_0$  は後件部変数の全体集合  $Y$  におけるファジィ集合であらわされる。

次に図 2.9, 2.10 のような前件部, 後件部のメンバーシップ関数を与えたとする。

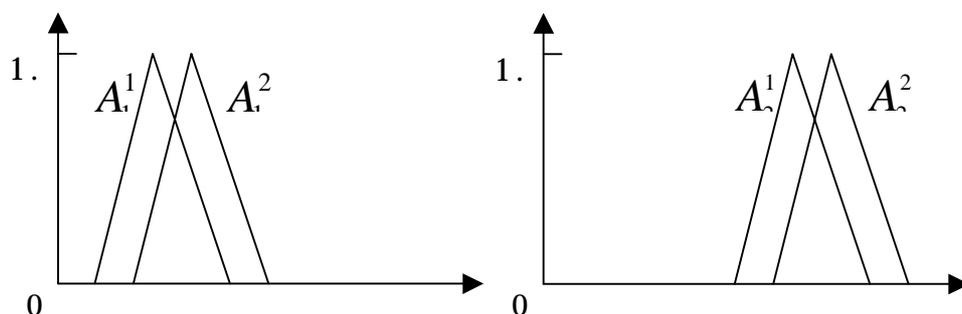


図 2.9 前件部のメンバーシップ関数

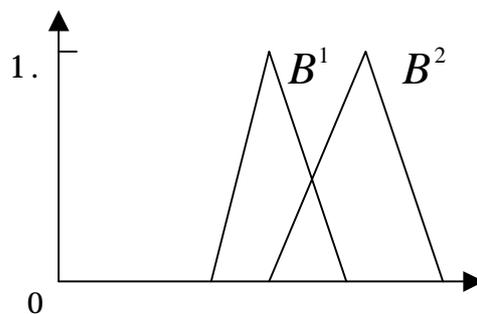


図 2.10 後件部のメンバーシップ関数

min-max 重心法は以下に手順により行われる。

Step 1: 各規則の適合度  $\omega_i$  を計算する。

$$\omega_i = \mu_{A_1^i}(A_1) \wedge \mu_{A_2^i}(A_2) \quad (2.15)$$

ただし,  $i=1,2$

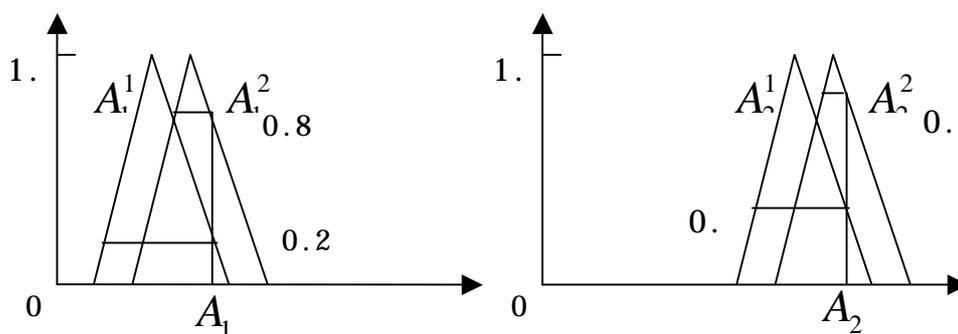


図 2.11 適合度

Step 2: 各規則の後件部の推論結果  $B^{i'}$  を計算する。

$$\mu_{B^{i'}}(y) = \omega_i \wedge \mu_{B^i}(y) \quad (2.16)$$

ただし,  $i=1,2$

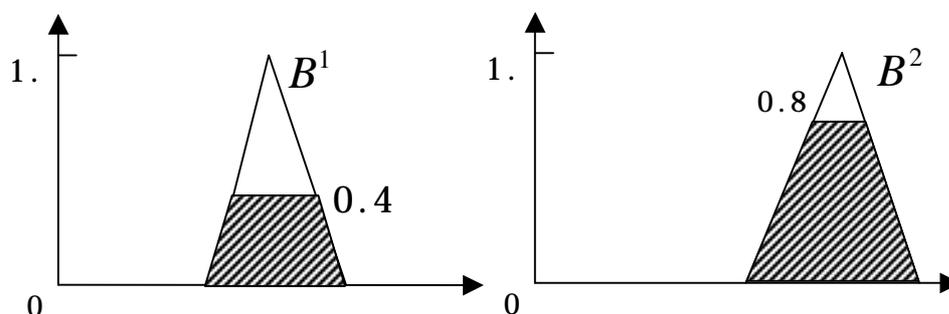


図 2.12 各規則の推論結果  $B^{i'}$

Step 3 : 各規則の推論結果  $B^{i'}$  を統合して , 規則全体の推論結果  $B'$  を計算する .

$$\mu_{B'}(y) = \bigvee_{i=1}^n \mu_{B^{i'}}(y) \quad (2.17)$$

ただし ,  $i=1,2$

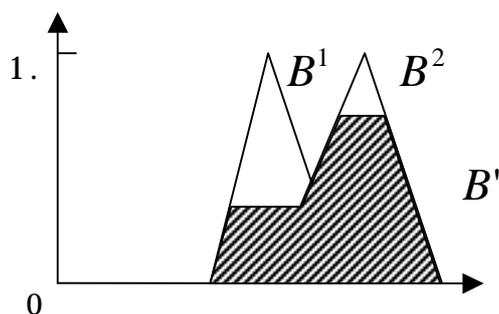


図 2.13 推論結果  $B'$

Step 4 : 推論結果  $B'$  の重心を計算する .

$$y_0 = \frac{\int y \mu_{B'}(y) dy}{\int \mu_{B'}(y) dy} \quad (2.18)$$

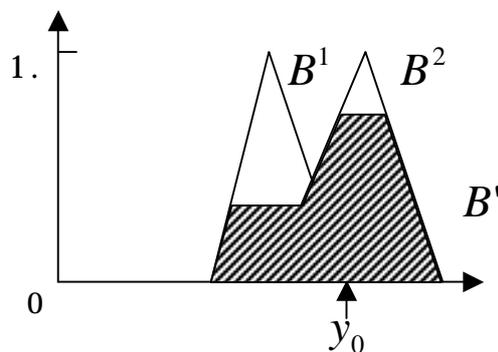


図 2.14 推論結果  $B'$  の重心

ここで  $\mu_A(x)$  は  $x$  がファジィ集合  $A$  に属する度合またはグレードを表している。また、 $\wedge$  は min 演算、 $\vee$  は max 演算を表している。つまり、適合度  $w_i$  とはファジィ制御装置の入力情報  $A_1$ 、 $A_2$  が番目の規則にどの程度マッチングしているかを表している。推論結果はファジィ集合の形で求められる。このためこの推論結果は一般的に実際の制御対象の操作量として入力することは困難である。そのためファジィ制御では制御対象の制御量として確定した値、つまり非ファジィ値を求める必要がある。この時、一般的に推論結果の代表値を求める手段として重心法が用いられている。この重心法によって求められた値を推論結果としてファジィ制御装置の出力として扱っている。また、この非ファジィ値を求める一連の操作を非ファジィ化といい、今回用いた重心法の他に面積法、最大値法などがある。ところで、この推論法は後件部のファジィ集合が前件部の適合度によって頭切りされることから頭切り法とも呼ばれている。

### 2.3.3 簡略推論法

この推論法は、ファジィ推論の高速化と簡略化を主眼とした推論法である。この推論法は、制御規則の後件部がファジィ集合ではなく、定数で与えられる場合を考えている。

次に、簡略推論法の方法について述べる。

まず、表 2.3 のような推論規則を与える。

表 2.3 推論規則

$R^1$	:	IF $x_1$ is $A_1^1$ and $x_2$ is $A_2^1$ then $y$ is $B^1$
$R^2$	:	IF $x_1$ is $A_1^2$ and $x_2$ is $A_2^2$ then $y$ is $B^2$
事実	:	$x_1$ is $A_1$ and $x_2$ is $A_2$
推論結果	:	$y$ is $y_0$

$x_1, x_2$ は前件部変数であり,  $y$ は後件部変数である. 前件部  $A_1^1, A_2^1, A_1^2, A_2^2$ はファジィ集合で与えられる. 一方, 後件部  $B^1, B^2$ はファジィ集合ではなく実数値で与えられる. 事実  $x_1, x_2$ も非ファジィ値  $A_1, A_2$ で与えられる.

簡略推論法は以下の手順によって行われる.

Step 1: 各規則の適合度  $\omega_i$ を計算する.

$$\omega_i = \mu_{A_1^i}(A_1) \times \mu_{A_2^i}(A_2) \quad (2.19)$$

$$(\text{もしくは } \omega_i = \mu_{A_1^i}(A_1) \wedge \mu_{A_2^i}(A_2))$$

ただし,  $i=1,2$

Step 2: 推論結果  $y_0$ を次式で計算する.

$$y_0 = \frac{\sum_{i=1}^n \omega_i \cdot B_i}{\sum_{i=1}^n \omega_i} \quad (2.18)$$

ただし,  $i=1,2$

この簡略型推論法は, 先ほどの min-max 重心法と比べて, 非ファジィ化の演算操作が必要ないため, 演算速度が格段に早くなる. このことは制御において制御周期を短縮することができるという利点につながる. 実際, 多くのファジィ制御において後件部を実数値で与えている場合が多い. また, この推論法は単純な四則演算をつかった式となっているため, 解析が容易になるといった特徴をもっている.

## 2.4 距離型ファジィ推論法とその特徴

ところで，今までに紹介した直接推論法は，前件部と与えられた事実との共通集合の高さを前件部と事実がどの程度一致しているかを示す適合度をファジィ推論の根拠として用いている．そのためいくつかの欠点がある．例えば，複数個の推論規則における前件部が疎である場合，つまり，前件部集合が互いに空の共通部分を持つ場合，事実がこの空集合に入力されると直接推論法では推論の結論を求めることができない．図 2.15 に簡単な例を示す．

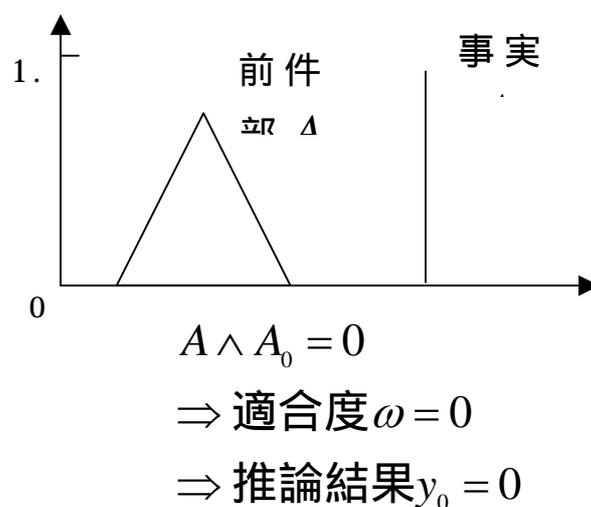


図 2.15 直接ファジィ推論法の欠点 1

また，推論規則の前件部，後件部と事実が凸なファジィ集合である場合には推論結果も凸なることが望まれる．しかし，従来の直接推論法では必ずしも推論結果が望まれる結果になるとは限らない．図 2.16 に簡単な例を示す．

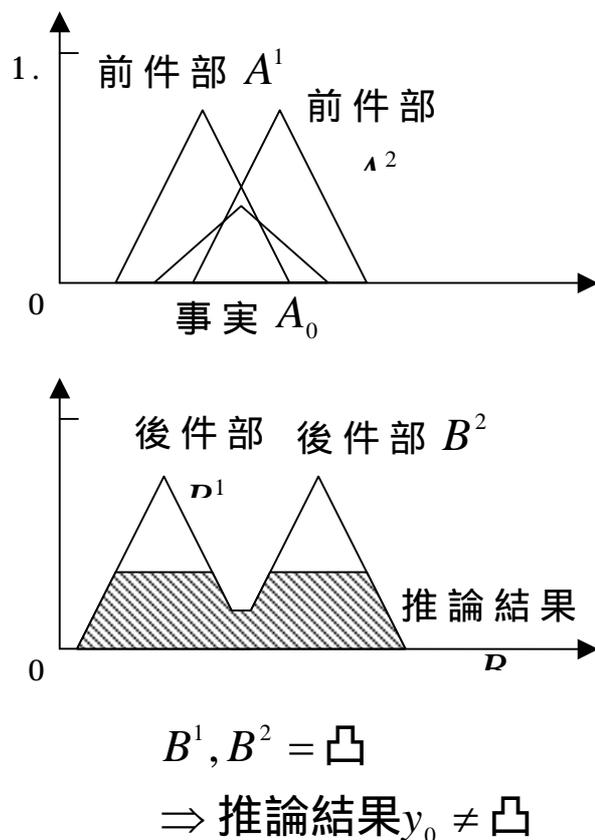


図 2.16 直接ファジィ推論法の欠点 2

前件部と事実が一致した場合，推論規則において前件部に対応した後件部が推論結果として望まれるが，この場合においても必ずしも望まれる推論結果になるとは限らない．これは分離規則が満たされていないことになる．図 2.17 に簡単な例を示す．

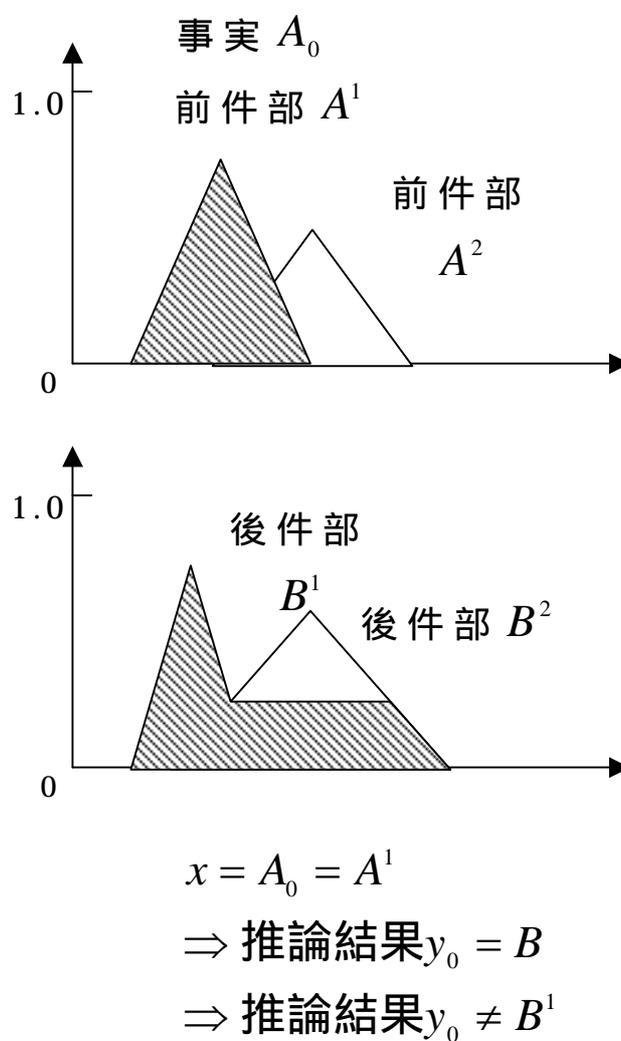


図 2.17 直接ファジィ推論法の欠点 3

こうした直接推論法の欠点を克服するためには、従来のファジィ推論の仕組みとは異なる出発点から議論することが必要である。推論の根拠としてファジィ集合間の距離に基づく距離型ファジィ推論法を提案した。この推論法は先ほど述べたように推論の根拠がファ

ジィ集合間の距離に基づいており，先ほどの直接法の欠点を克服している．つまり，

1. 前件部と事実の集合が相容れない場合でも推論が可能である．
  2. 後件部の凸なファジィ集合であれば，推論結果も凸なファジィ集合になる．
  3. 分離規則を厳密に満たしている．
- といった特徴をもっている．

次にファジィ集合間の距離の公理について述べる．

実数の全体  $R$  を全体集合として， $F(R)$  で  $R$  におけるファジィ集合の全体を表す．また異なる概念を区別して表すときには， $R$  の右下に添え字を付けて区別する．

定義 1: ファジィ集合  $A \in F(R)$  が有界であるとは， $\forall \alpha \in (0,1]$  に対して， $A$  の  $\alpha$ -レベル集合

$$A_\alpha = \{x \in R \mid \mu_A(x) \geq \alpha\} \quad (2.19)$$

が有界であるときである．

定義 2: ファジィ集合  $A \in F(R)$  の高さ  $M_A$  は次のように定義される．

$$M_A = \sup \mu_A(x) \quad (2.20)$$

定義 3: 高さ  $M_A$  が 1 であるファジィ集合  $A$  は正規なファジィ集合と呼ばれる．

定義 4:  $\forall_{x_1, x_2} \in R$  に対して，ファジィ集合  $A \in F(R)$  のメンバーシップ関数  $\mu_A(x)$  が次式を満たしていれば，ファジィ

集合  $A$  は凸なファジィ集合であるという。ただし、

$$\forall x_0 \in [\min(x_1, x_2), \max(x_1, x_2)].$$

$$\mu_A(x_0) \geq \min[\mu_A(x_1), \mu_A(x_2)] \quad (2.21)$$

以下では有界凸なファジィ集合の全体を  $\overline{F}(R)$  で表す。また、 $\overline{F}_n(R)$  で  $\overline{F}(R)$  における正規なファジィ集合の全体を表す。つまり、 $\overline{F}_n(R)$  は有界かつ正規な凸ファジィ集合の全体を表す。

定義 5: 任意の有界なファジィ集合  $A$  のメンバーシップ関数  $\mu_A(x)$  をとして、 $A$  の  $\alpha$ -レベル集合を  $A_\alpha$  と表して、 $A_\alpha$  の上限  $\sup A_\alpha$  と  $A_\alpha$  の下限  $\inf A_\alpha$  を次のように定義する。

$\alpha \neq 0$  のとき

$$\sup A_\alpha = \sup\{x \in R \mid \mu_A(x) \geq \alpha\} \quad (2.22)$$

$$\inf A_\alpha = \inf\{x \in R \mid \mu_A(x) \geq \alpha\} \quad (2.23)$$

$\alpha = 0$  のとき

$$\sup A_0 = \sup\{x \in R \mid \mu_A(x) > 0\} \quad (2.24)$$

$$\inf A_0 = \inf\{x \in R \mid \mu_A(x) > 0\} \quad (2.25)$$

以下では混乱を起こさない限り  $\sup A_\alpha$  を  $\overline{A}_\alpha$ 、 $\inf A_\alpha$  を  $\underline{A}_\alpha$  と書くことがある。

$A$  が有界凸なファジィ集合である場合、 $A$  の  $\alpha$ -レベル集合  $A_\alpha$  は閉区間になり次のように表せる。

$$A_\alpha = [\inf A_\alpha, \sup A_\alpha] = [\underline{A}_\alpha, \overline{A}_\alpha] \quad (2.26)$$

定義 6: 高さ  $M_A$  のファジィ集合  $A$  を正規化したファジィ集合  $A_M$  のメンバーシップ関数  $\mu_{A_M}(x)$  は次のように定義する .

$$\mu_{A_M}(x) = \frac{1}{M_A} \mu_A(x), \forall x \in R \quad (2.27)$$

もし,  $A$  が  $\overline{F}(R)$  におけるファジィ集合であれば, で正規化したファジィ集合  $A_M$  は  $\overline{F}_n(R)$  中の要素となり, 定義 5 により  $A_M$  の上限と下限は次の式で表される .

$$\sup A_{M_\alpha} = \sup \{x \in R \mid \mu_{M_\alpha}(x) \geq \alpha \cdot M_A\} \quad (2.28)$$

$$\inf A_{M_\alpha} = \inf \{x \in R \mid \mu_{M_\alpha}(x) \geq \alpha \cdot M_A\} \quad (2.29)$$

ただし,  $a \in (0,1)$

定義 7: ある台集合上で定義された任意の 2 つのファジィ集合  $A, B$  に実数  $d(A, B)$  が対応し, 次の 3 つの条件を満足するとき,  $d(A, B)$  をこの台集合上の距離関数という . ただし,  $C$  は台集合上における任意のファジィ集合である .

$$d(A, B) \geq 0; d(A, B) = 0 \Leftrightarrow A = B \quad (2.30)$$

$$d(A, B) = d(A, B) \quad (2.31)$$

$$d(A, B) \leq d(A, C) + d(C, B) \quad (2.32)$$

定義 8: 連続メンバーシップ関数をもつファジィ集合  $A, B \in \overline{F}(R)$  に対して, 次式で定義される実数関数  $d(A, B)$  を  $\overline{F}(R)$  上の距離関数という . ただし,  $1 \leq p < \infty$ ,  $|\cdot|$  は絶対値で表す .

$$\begin{aligned}
d(A, B) = & \left[ \int_0^1 |\inf A_{M\alpha} - \inf B_{M\alpha}|^p d\alpha \right]^{\frac{1}{p}} \\
& + \left[ \int_0^1 |\sup A_{M\alpha} - \sup B_{M\alpha}|^p d\alpha \right]^{\frac{1}{p}} \\
& + \left[ \int_{-\infty}^{\infty} \left| \left( \frac{1}{M_A} - 1 \right) \mu_A(x) - \left( \frac{1}{M_B} - 1 \right) \mu_B(x) \right|^p dx \right]^{\frac{1}{p}}
\end{aligned} \tag{2.33}$$

この式は定義 7 で述べた 3 つの条件を満たしている。

以上で距離型ファジィ推論法におけるファジィ集合間の距離の公理について述べた。ここで、距離関数の式において、 $p$  の値を 2 として、最もよく使われる典型的なファジィ集合間の簡略化距離計算公式を与える。実際の問題ではこれらの公式を利用すれば、非常に便利である。

次にいくつかのファジィ集合の基本形の距離の計算式を示す。

### 1. 三角型ファジィ集合の場合

$\bar{F}(R)$  における三角型ファジィ集合は、高さ、高さのとり座標、0-レベル集合の上限と下限により完全に表される。図 (2.18) に示すような三角型ファジィ集合  $A, B \in \bar{F}(R)$  をパラメータの形で書くと  $T_A(a_1, a_2, a_3, a)$ 、 $T_B(b_1, b_2, b_3, b)$  となる。

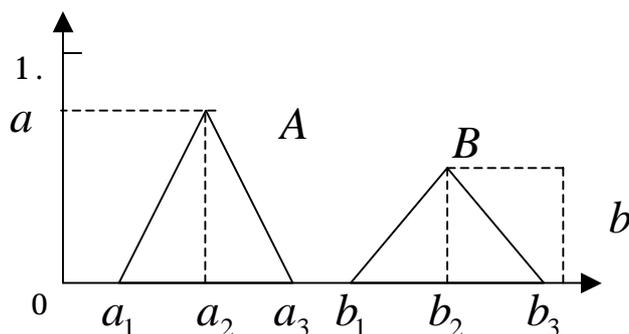


図 2.18 三角型ファジィ集合

距離関数に基づいて，図 2.18 のような三角型ファジィ集合  $A$  と  $B$  の距離計算式は次式になる．

$$d(A, B) = \frac{1}{\sqrt{3}} \sum_{i=1}^2 \left[ \sum_{j=i}^{i+1} (a_j - b_j)^2 + \prod_{j=i}^{i+1} |a_j - b_j| \right]^{\frac{1}{2}} + \frac{1}{\sqrt{3}} \left[ (1-a)(a_3 - a_1)^{\frac{1}{2}} + (1-b)(b_3 - b_1)^{\frac{1}{2}} \right] \quad (2.34)$$

式(2.34)は，もし  $A, B$  が共に正規なファジィ集合であれば，式(2.34)の右辺の第 2 項が 0 になる．

## 2. 台形型ファジィ集合の場合

台形型ファジィ集合  $A, B \in \overline{F}(R)$  は，図(2.19)に示すような 5 つのパラメータが必要である．図(2.19)に示すような台形型ファジィ集合はそれぞれ  $T_A(a_1, a_2, a_3, a_4, a)$ ， $T_B(b_1, b_2, b_3, b_4, b)$  ととなる．

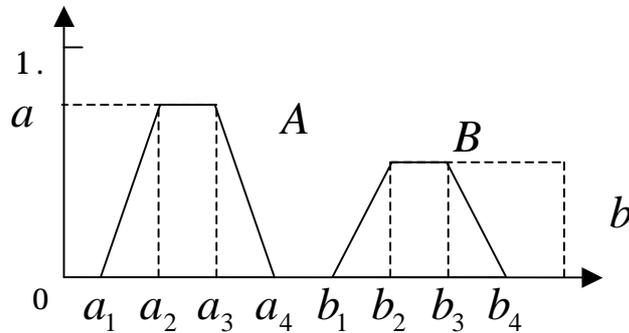


図 2.19 台形型ファジィ集合

距離関数に基づいて，図 2.19 のような台形型ファジィ集合  $A$  と  $B$  の距離計算式は式 2.35 になる．

$$\begin{aligned}
 d(A, B) = & \frac{1}{\sqrt{3}} \sum_{i=1,3}^{i+1} \left[ \sum_{j=i}^{i+1} (a_j - b_j)^2 + \prod_{j=i}^{i+1} |a_j - b_j| \right]^{\frac{1}{2}} \\
 & + \frac{1}{\sqrt{3}} (1-a) [a_4 + 2(a_3 - a_2) - a_1]^{\frac{1}{2}} \\
 & + \frac{1}{\sqrt{3}} (1-b) [b_4 + 2(b_3 - b_2) - b_1]^{\frac{1}{2}}
 \end{aligned} \tag{2.35}$$

$A$  と  $B$  が正規なファジィ集合であれば，式 (2.35) の右辺の第 2 項，第 3 項が 0 になる．また， $a_2 = a_3$ ， $b_2 = b_3$  とすれば，台形型ファジィ集合は三角型ファジィ集合となるので，式 (2.35) は式 (2.34) そのものになる．

### 3. 型ファジィ集合の場合

対称な 型ファジィ集合を表す場合，3 つのパラメータが必要であるが，ここでは一般的な非対称の 型ファジィ集合を対象とする場合もあることを考えて，図 (2.20) に示すような非対称な 型ファジィ集合間の距離

計算式を与える。このとき4つのパラメータが必要となり、 $\pi_A(a_1, a_2, a_3, K_a)$ 、 $\pi_B(b_1, b_2, b_3, K_b)$ と書く。AとBのメンバーシップ関数を詳しく書けば次式のようなになる。

$$\mu_{\pi_A}(x) = \begin{cases} \frac{2K_a}{(a_2 - a_1)^2} (x - a_1)^2, & a_1 \leq x \leq \frac{a_2 + a_1}{2} \\ K_a - \frac{2K_a}{(a_2 - a_1)^2} (x - a_2)^2, & \frac{a_2 + a_1}{2} \leq x \leq a_2 \\ K_a - \frac{2K_a}{(a_3 - a_2)^2} (x - a_2)^2, & a_2 \leq x \leq \frac{a_2 + a_3}{2} \\ \frac{2K_a}{(a_3 - a_2)^2} (x - a_3)^2, & \frac{a_2 + a_3}{2} \leq x \leq a_3 \end{cases} \quad (2.36)$$

$$\mu_{\pi_B}(x) = \begin{cases} \frac{2K_b}{(b_2 - b_1)^2} (x - b_1)^2, & b_1 \leq x \leq \frac{b_2 + b_1}{2} \\ K_b - \frac{2K_b}{(b_2 - b_1)^2} (x - b_2)^2, & \frac{b_2 + b_1}{2} \leq x \leq b_2 \\ K_b - \frac{2K_b}{(b_3 - b_2)^2} (x - b_2)^2, & b_2 \leq x \leq \frac{b_2 + b_3}{2} \\ \frac{2K_b}{(b_3 - b_2)^2} (x - b_3)^2, & \frac{b_2 + b_3}{2} \leq x \leq b_3 \end{cases} \quad (2.37)$$

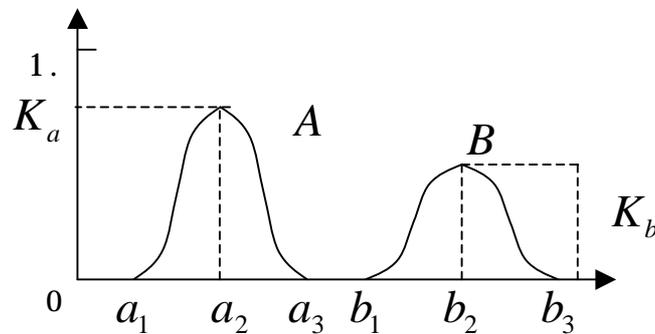


図 2.20 型ファジィ集合

距離関数に基づいて，式(2.36)と式(2.37)を用いて求められる距離計算式は次式のようなになる．

$$\begin{aligned}
 d(A, B) = & \sum_{i=1}^2 \left[ \frac{7}{24} \sum_{j=i}^{i+1} (a_j - b_j)^2 + \frac{5}{12} \prod_{j=i}^{i+1} |a_j - b_j| \right]^{\frac{1}{2}} \\
 & + (1 - K_a) \left[ \sum_{i=1}^2 \frac{23}{60} (a_{i+1} - a_i) \right]^{\frac{1}{2}} \\
 & + (1 - K_b) \left[ \sum_{i=1}^2 \frac{23}{60} (b_{i+1} - b_i) \right]^{\frac{1}{2}}
 \end{aligned} \tag{2.38}$$

もし， $K_a = K_b = 1$ ，つまり  $A$  と  $B$  が共に正規なファジィ集合であれば，式(2.38)の右辺にある第2項，第3項が0になる．

) 三角型ファジィ集合とシングルトンの場合

ファジィ制御などにおいて，推論規則における前件部と後件部を表すファジィ集合としては三角型ファジィ集合がよく使われる．また，事実はファジィ集合ではなく，通常は電圧，位置，速度などのようなセンサからの実数値となる．このとき，図 2.21 のような三角型ファジィ集合  $A$  とシングルトン  $B$  との距離計算式は式(2.39)で与えられる．尚，ここでシングルトンは三角型ファジィ集合の幅を0に収束させた時の極限状態として考えファジィ集合として扱っている．

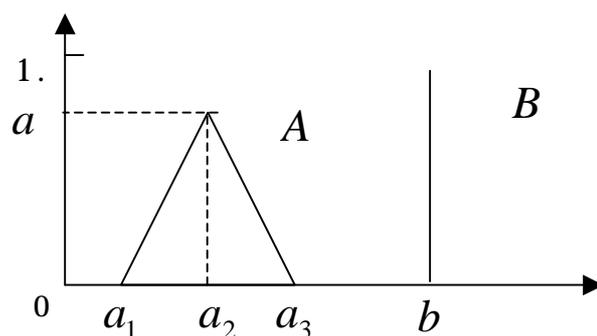


図 2.21 三角型ファジィ集合とシングルトン

$$d(A, B) = \frac{1}{\sqrt{3}} \sum_{i=1}^2 \left[ \sum_{j=i}^{i+1} (a_j - b)^2 + \prod_{j=i}^{i+1} |a_j - b| \right]^{\frac{1}{2}} + \frac{1}{\sqrt{3}} \left[ (1-a)(a_3 - a_1)^{\frac{1}{2}} \right] \quad (2.39)$$

) シングルトンの場合

図 2.22 のようなシングルトン  $A$  とシングルトン  $B$  を与える。

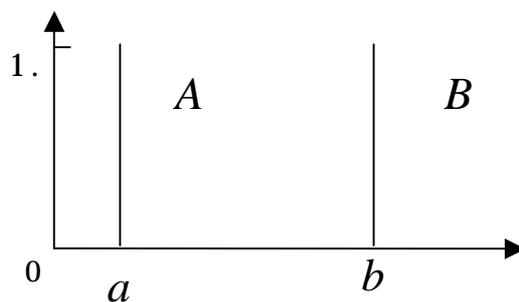


図 2.24 シングルトン

シングルトンの場合，メンバーシップ関数の最大値が 1 で， $\alpha$ -レベル集合の上限と下限が等しいので，距離計

算式は式(2.40)となる .

$$d(A, B) = |a - b| \quad (2.40)$$

実際にこれらの距離計算式をファジィ推論に用いるときは , なるべく計算量を減らすことを考えて , それらの計算式の前に掛けている定数  $1/\sqrt{3}$  を 1 としても差し支えはない . これは距離の公理を満たすすべての距離関数に 0 以外の任意の定数を掛けても , やはり距離の公理は満たされるからである .

次にこのファジィ集合間の距離を利用した距離型ファジィ推論法について述べる . まず表 2.5 のような推論規則を与える .

表 2.5 推論規則

$$R^1 : x_1 = A^{11}, \dots, x_m = A^{1m} \Rightarrow y = B^1$$

$$R^2 : x_1 = A^{21}, \dots, x_m = A^{2m} \Rightarrow y = B^2$$

$$\begin{array}{cccccc} \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{array}$$

$$R^i : x_1 = A^{i1}, \dots, x_m = A^{im} \Rightarrow y = B^i$$

$$\begin{array}{cccccc} \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{array}$$

$$R^n : x_1 = A^{n1}, \dots, x_m = A^{nm} \Rightarrow y = B^n$$

$$\text{事実} : x_1 = A^1, \dots, x_m = A^m$$

---


$$\text{推論結果} : y = B$$

ただし ,  $i = 1, 2, \dots, n$  ,  $j = 1, 2, \dots, m$  .  $A^{ij}$  ,  $B^i$  ,  $A^j$  ,  $B$  はそれぞれ前件部 , 後件部 , 与えられた事実および推論結果を表し ,

$A^{ij}, A^j \in F(R_{A^j}), B^i \in \overline{F}(R_B)$  とする . また , 次の 2 つの仮定のもとで推論を行う .

1. 前件部ファジィ集合  $A^{ij}$  と事実ファジィ集合  $A^j$  の種類について特に制限がないが , 推論規則  $R^1 \sim R^n$  において , 前件部がまったく同一である推論規則は存在しないとする . すなわち ,  $\forall_{q_1, q_2} \in \{1, 2, \dots, n\}, q_1 \neq q_2$  に対して ,
 
$$\sum_{i=1}^m d(A^{q_1 i}, A^{q_2 i}) \neq 0$$
 とする . この仮定は互いに矛盾する推論規則を排除するためである .
2. 後件部について ,  $k \in \{1, \dots, n\}$  が 1 つ以上存在して ,  $B^k \in \overline{F}_n(R_B)$  を満たす . この仮定は推論結果に正規性を持たせるためである . つまり ,  $B \in \overline{F}_n(R_B)$  を満たすためである . もし推論結果がこの性質を必要としない場合 , この仮定を外しても問題ない .

距離型ファジィ推論法は以下に手順によって行われる .

Step 1 : まず , 定義 7 の公理を満たす計算法を用いて ,  $A^{ij}$  と  $A^j$  との距離  $d_{ij}(A^{ij}, A^j)$  を計算する . 次式により  $d_1 \sim d_n$  を計算する .

$$d_i = \sum_{j=1}^m d_{ij}(A^{ij}, A^j) \quad (2.41)$$

$$i = 1, 2, \dots, n$$

このように計算された  $d_1 \sim d_n$  も距離の公理を満足する . また , 仮定 ) より  $d_1 \sim d_n$  の中には 2 つ以上 0

になるものは存在しないことがわかる .

Step 2: 次のように , 推論結果  $B$  の  $\alpha$ -レベル集合を求める .

$$B_\alpha = [\inf(B_\alpha), \sup(B_\alpha)] \quad (2.42)$$

$$\inf(B_\alpha) = \frac{\sum_{i=1}^n \left[ \inf(B_\alpha^i) \prod_{j=1, j \neq i}^n d_j \right]}{\sum_{i=1}^n \prod_{j=1, j \neq i}^n d_j} \quad (2.43)$$

$$\sup(B_\alpha) = \frac{\sum_{i=1}^n \left[ \sup(B_\alpha^i) \prod_{j=1, j \neq i}^n d_j \right]}{\sum_{i=1}^n \prod_{j=1, j \neq i}^n d_j} \quad (2.44)$$

または , 区間数の演算により式 (2.42) は式 (2.45) のように表せる .

$$\begin{aligned} B_\alpha &= \frac{\sum_{i=1}^n \left[ B_\alpha^i \prod_{j=1, j \neq i}^n d_j \right]}{\sum_{i=1}^n \prod_{j=1, j \neq i}^n d_j} \\ &= \left[ \frac{\sum_{i=1}^n \left[ \frac{B_\alpha^i}{\prod_{j=1, j \neq i}^n d_j} \right]}{\sum_{i=1}^n \prod_{j=1, j \neq i}^n d_j}, \frac{\sum_{i=1}^n \left[ \frac{\overline{B_\alpha^i}}{\prod_{j=1, j \neq i}^n d_j} \right]}{\sum_{i=1}^n \prod_{j=1, j \neq i}^n d_j} \right] \end{aligned} \quad (2.45)$$

もし , 後件部の中に  $B^k \notin \overline{F_n}(R_B)$  ,  $k = 1, 2, \dots, n$  が

存在する場合,  $\forall \alpha \in (\sup \mu_{B^k}(y), 1]$  に対する  $B_\alpha^k$  は空の集合となる. この場合においても推論を可能にするためには, 次の約束により推論を行う. つまり

$\forall \alpha \in (\sup \mu_{B^k}(y), 1]$  に対して,

$$B_\alpha^k = [\inf(B_\alpha^k), \sup(B_\alpha^k)] = [\inf(B_{M_{B^k}}^k), \sup(B_{M_{B^k}}^k)] \quad (2.46)$$

とする.

Step 3: 合成定理により式 (2.47) で推論結果  $B$  を求める.

$$B = \bigcup_{\alpha} \alpha \cdot B_{\alpha} \quad (2.47)$$

以上で距離型ファジィ推論法のアルゴリズムについて述べた. 通常のファジィ推論法では, 前件部はファジィ集合でなければ推論できない. これに対して距離型ファジィ推論法は推論の根拠が距離に基づいているので, 前件部がシングルトンである場合においても適切な推論結果を得ることができる.

次に直接型ファジィ推論法である関数型推論法と簡略推論法と同様にして, 距離型ファジィ推論法の推論規則の後件部を関数もしくは実数にした場合についての推論方法について述べる. まず, ファジィ集合間距離に基づく関数型推論法について述べる. 表 2.6 のような推論規則を与える.

表 2.26 推論規則

$R^1 : x_1 = A^{11}, \dots, x_m = A^{1m} \Rightarrow y = f_1(x_1, \dots, x_m)$
$R^2 : x_1 = A^{21}, \dots, x_m = A^{2m} \Rightarrow y = f_2(x_1, \dots, x_m)$
. . . . .
. . . . .
. . . . .
$R^i : x_1 = A^{i1}, \dots, x_m = A^{im} \Rightarrow y = f_i(x_1, \dots, x_m)$
. . . . .
. . . . .
. . . . .
$R^n : x_1 = A^{n1}, \dots, x_m = A^{nm} \Rightarrow y = f_n(x_1, \dots, x_m)$
事実 : $x_1 = A^1, \dots, x_m = A^m$
推論結果 : $y = y_0$

ただし,  $i = 1, 2, \dots, n$ ,  $j = 1, 2, \dots, m$ .  $A^{ij}$ ,  $f_i$ ,  $A^j$ ,  $y_0$ はそれぞれ前件部, 後件部, 与えられた事実および推論結果を表し,  $A^{ij}, A^j \in F(R_{A^j})$ ,  $y_0, f_i(x_1, \dots, x_m) \in R$ とする. また, 次の先ほど述べた 2 つの仮定のうち, ( ) だけを推論条件とする.

ファジィ集合間距離に基づく関数型推論法は以下の手順によって行われる.

Step 1: まず, 定義 7 の公理を満たす計算法を用いて  $A^{ij}$  と  $A^j$  との距離を計算する. 次の式 (2.50) により  $d_1 \sim d_n$  を計算する.

$$d_i = \sum_{j=1}^m d_{ij}(A^{ij}, A^j) \quad (2.48)$$

$$i = 1, 2, \dots, n$$

Step 2: 事実の重心  $A^1 \sim A^m$  をそれぞれ  $x_1^0 \sim x_m^0$  として, 推論結果  $y_0$  は式 ( ) で求める .

$$y_0 = \frac{\sum_{i=1}^n \left[ f_i(x_1^0, \dots, x_m^0) \prod_{j=1, j \neq i}^n d_j \right]}{\sum_{i=1}^n \prod_{j=1, j \neq i}^n d_j} \quad (2.49)$$

次に, ファジィ集合間距離に基づく簡略推論法について述べる .  
表 2.27 のような推論規則を与える .

表 2.27 推論規則

$$R^1 : x_1 = A^{11}, \dots, x_m = A^{1m} \Rightarrow y = C_1$$

$$R^2 : x_1 = A^{21}, \dots, x_m = A^{2m} \Rightarrow y = C_2$$

$$\begin{array}{cccccc} \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{array}$$

$$R^i : x_1 = A^{i1}, \dots, x_m = A^{im} \Rightarrow y = C_i$$

$$\begin{array}{cccccc} \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{array}$$

$$R^n : x_1 = A^{n1}, \dots, x_m = A^{nm} \Rightarrow y = C_n$$

$$\text{事実} : x_1 = A^1, \dots, x_m = A^m$$

---


$$\text{推論結果} : y = y_0$$

ただし,  $i = 1, 2, \dots, n$ ,  $j = 1, 2, \dots, m$ .  $A^{ij}$ ,  $C_i$ ,  $A^j$ ,  $y_0$  はそれぞれ前件部, 後件部, 与えられた事実および推論結果を表し,

$A^{ij}, A^j, y_0, C_i \in R$ とする . また , 次の先ほど述べた 2 つの仮定のうち , )だけを推論条件とする .

ファジィ集合間距離に基づく簡略推論法は以下の手順によって行われる .

Step 1 : まず , 定義 7 の公理を満たす計算法を用いて  $A^{ij}$  と  $A^j$  との距離を計算する . 次式により  $d_1 \sim d_n$  を計算する .

$$d_i = \sum_{j=1}^m d_{ij}(A^{ij}, A^j) \quad (2.50)$$

$$i = 1, 2, \dots, n$$

また , 事実が実数である場合は距離関数において  $p$  の値を 1 として

$$d_i = \sum_{j=1}^m |x^{ij} - x^j| \quad (2.51)$$

$$i = 1, 2, \dots, n$$

となる .

Step 2 : 推論結果は式 (2.52) で求める .

$$y_0 = \frac{\sum_{i=1}^n \left[ C_i \prod_{j=1, j \neq i}^n d_j \right]}{\sum_{i=1}^n \prod_{j=1, j \neq i}^n d_j} \quad (2.52)$$

もちろん , これら 2 つの推論法は距離型ファジィ推論法の特徴を

有している。先に述べたように制御では入出力情報は、通常、実数値である。このため、これらの推論法は制御には非常に有利である。

距離型ファジィ推論法には、以下のような特徴が挙げられる。

1. 分離規則を満たす
2. 後件部が凸のファジィ集合であれば、推論結果も凸なファジィ集合となる
3. 漸近特性を持つ
4. 前件部集合が互いに疎な場合でも推論可能

分離規則とは、 $A = A^1$  なら  $B = B^1$  となる事である。つまり、前件部集合  $A^1$  に事実  $A$  が重なった場合、後件部  $B^1$  に推論結果  $B$  が重なる事を指す。

漸近特性とは、前件部集合  $A$  と事実  $A$  の距離が 0 に収束すれば、後件部  $B^1$  と推論結果  $B$  の距離も 0 に収束することを指す。

前件部集合が互いに疎な場合でも推論可能というのは、事実がどの前件部集合にも属することができない場合でも推論可能であるという事を示す。

---

## 第三章 距離型ファジィ推論法に基づく 軌道計画法

### 3.1 案内ロボットのファジィ軌道計画法

まず、案内役の言語的知識をファジィルールにより表現し、ファジィ集合を用いてファジィ変数を定量化する。次に距離型ファジィ推論法に基づく案内ロボットの軌道計画アルゴリズムについて説明する。

ファジィ集合とファジィルールの設定には十分な自由度があり、入出力関係においては多様な非線形特性が得られる。よって、ファジィ集合とファジィルールを適切に活用することにより、非線形の強い実システムをより精度よく近似することができる。これは、一つの方程式でシステムの特性のすべてを近似しようとする従来の数理モデル論と根本的に異なる案内役と人間との間に見られる相互関係は言語ルールで簡単に表現出来るので、ファジィ推論を用いて案内ロボットの目的軌道計画を行うことにした。

### 3.2 ファジィ軌道計画法の導入

案内ロボットの軌道計画設計における問題はロボットと人間との

相互関係を如何に取り入れるかである。この問題を解決する為にはファジィ推論の導入が適切である。人間とロボットとの間に見られる相互関係を案内知識としてとりだして表現する。ファジィ推論の導入にあたって以下の特徴があげられる。

1. 案内知識をファジィ推論における言語ルールを用いて表現する事によってファジィ推論を軌道計画に導入する事ができる。そのため厳密な数学的規則を設けることなく自由な計画を設計できる。また、案内知識を第三者が主観的に決定する事ができる為、軌道計画への導入が容易である。
2. ファジィ推論は従来のような画一的な関数型で表現されるのではなく、複数の制御規則によって表現されている。そのため単純な制御規則を追加していく事によって、従来の関数型では表現が難しい複雑な計画も容易に実現する事ができる。
3. 人間の言語と言うあいまいさを含んだ表現そのものを表現するのにファジィ推論が適している。実際に、近い、遠いといった自然言語を画一的な評価関数で求めるよりもファジィ推論で自然言語を表現する方が現実的な手段といえる。
4. 制御規則が言語ルールを用いて表現されている為、人間に制御の様子が理解しやすい。同時に人間になじみやすい制御とも言える。

以上の特徴からわかるように、ファジィ集合とファジィルールの設定には十分な自由度があり、入出力関係においては多様な非線形特性が得られる。よって、ファジィ集合とファジィルールを適切に活用することにより、非線形の強い実システムをより精度よく近似することができる。これは、一つの方程式でシステムの特性のすべ

---

てを近似しようとする従来の数理モデル論と根本的に異なる案内役と人間との間に見られる相互関係を言語ルールで簡単に表現できる。

### 3.3 軌道計画法の設計手順

図 3.1 に距離型ファジィ推論法に基づいた軌道計画法の設計手順を示す。まず、軌道計画を行なうにあたり、その目的を決定する。今回の軌道計画の目的はロボットと人間の相互関係に基づいた案内軌道計画の実施である。今回の実験の場合、具体的には、近い、遠いといった人間の評価からロボットが速度を上げる、下げるといった一連の動作を決定するといった基本的な案内動作を行なうことである。次に、軌道計画法の目的をもとに推論規則の設定を行ないます。

まず、軌道計画法の目的をもとにして言語ルールを作成します。さらに、この作成された言語ルールを定式化します。次の手順は、言語変数の定量化です。定式化した言語ルールをファジィ集合を用いて定量化する。最後に、ファジィ推論法を行いこの推論結果を軌道値として用いる。

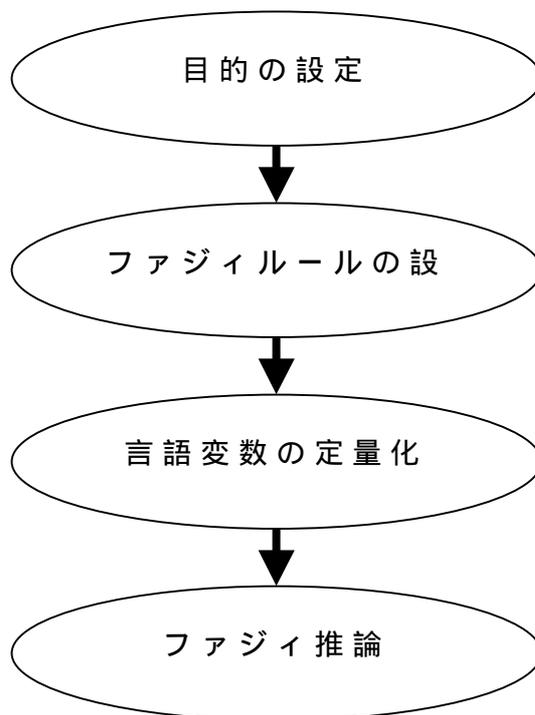


図 3.1 軌道計画法の設計手順

### 3.4 言語ルールによる知識の抽出

ファジィ計画における目的をもとに言語ルールの設定を行う。実験では評価基準として案内ロボットに設備された超音波センサにより検出した対象(人間)との距離を用いた。理由としては心理的評価が人間の視覚情報と通して評価できる。評価基準が明文化しやすい。導入が容易である。また、案内ロボットの軌道計画は速度指示によって行った。案内役の知識は表 3.1 に示す言語ルールに表現される。

表 3.1 言語ルール

- $R^1$ : もし, 対象との距離が非常に近ければ, 速度を出来る限り上げよ.
- $R^2$ : もし, 対象との距離が近ければ, 速度を上げよ.
- $R^3$ : もし, 対象との距離が少し近ければ, 速度を少し上げよ.
- $R^4$ : もし, 対象との距離が適切であれば, その速度を保て.
- $R^5$ : もし, 対象との距離が少し遠ければ, 速度を少し下げよ.
- $R^6$ : もし, 対象との距離が遠ければ, 速度を下げよ.
- $R^7$ : もし, 対象との距離が非常に遠ければ, 速度を出来る限り落とせ.

### 3.5 知識の定式化

表 3.1 の言語ルールにおける前件部と後件部を定式化する. 前件部変数は超音波センサから検出された対象との距離である. また, 後件部変数は軌道計画としての案内ロボットの速度, もしくは案内ロボットの速度の変化量を用いる. これから前件部変数は  $x_A$   $\{ A^1, A^2, A^3, A^4, A^5, A^6, A^7 \}$ , 後件部変数は  $y_B$   $\{ B^1, B^2, B^3, B^4, B^5, B^6, B^7 \}$ , もしくは  $y_C$   $\{ C^1, C^2, C^3, C^4, C^5, C^6, C^7 \}$ , と記述される.  $A^1, A^2, A^3, A^4, A^5, A^6, A^7$  及び,  $B^1, B^2, B^3, B^4, B^5, B^6, B^7, C^1, C^2, C^3, C^4, C^5, C^6, C^7$  はそれぞれのルールの前件部と後件部に対応した言語変数であり, 後件部  $y_B$  は案内ロボットの速度を, 後件部  $y_C$  は案内ロボットの速度の変化量を示している. 定式化表現すると次のページの表 3.2 のようになる.

表 3.2 推論規則

$$\begin{aligned}
 R^1: \quad & \text{IF } x \text{ is } A^1 \text{ then } y_B \text{ is } B^1 \\
 & \hspace{15em} (\text{or } y_C \text{ is } C^1) \\
 R^2: \quad & \text{IF } x \text{ is } A^2 \text{ then } y_B \text{ is } B^2 \\
 & \hspace{15em} (\text{or } y_C \text{ is } C^2) \\
 R^3: \quad & \text{IF } x \text{ is } A^3 \text{ then } y_B \text{ is } B^3 \\
 & \hspace{15em} (\text{or } y_C \text{ is } C^3) \\
 R^4: \quad & \text{IF } x \text{ is } A^4 \text{ then } y_B \text{ is } B^4 \\
 & \hspace{15em} (\text{or } y_C \text{ is } C^4) \\
 R^5: \quad & \text{IF } x \text{ is } A^5 \text{ then } y_B \text{ is } B^5 \\
 & \hspace{15em} (\text{or } y_C \text{ is } C^5) \\
 R^6: \quad & \text{IF } x \text{ is } A^6 \text{ then } y_B \text{ is } B^6 \\
 & \hspace{15em} (\text{or } y_C \text{ is } C^6) \\
 R^7: \quad & \text{IF } x \text{ is } A^7 \text{ then } y_B \text{ is } B^7 \\
 & \hspace{15em} (\text{or } y_C \text{ is } C^7)
 \end{aligned}$$

### 3.6 ファジィ集合による言語変数の定量化

前件部のメンバーシップ関数としては三角型ファジィ集合とシングルトン，及び台形型ファジィ集合を用いた．シングルトンはメンバーシップ関数の幅を 0 に収束させた時の極限状態とする．前件部において三角型ファジィ集合を用いたメンバーシップ関数を図 3.2 に，台形型ファジィ集合を用いたメンバーシップ関数を図 3.3 に示す．後件部のメンバーシップ関数はファジィ集合ではなく実数で与えた．また，後件部  $y_B$  の実数値を図 3.5 に示す．

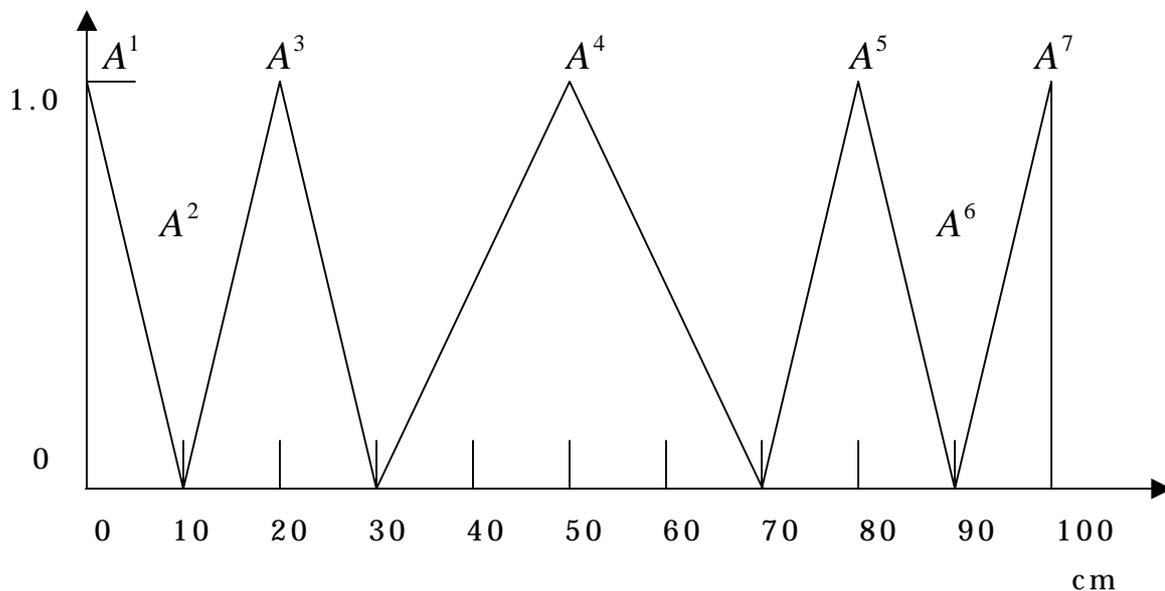


図 3.2 前件部のメンバーシップ関数(三角型)

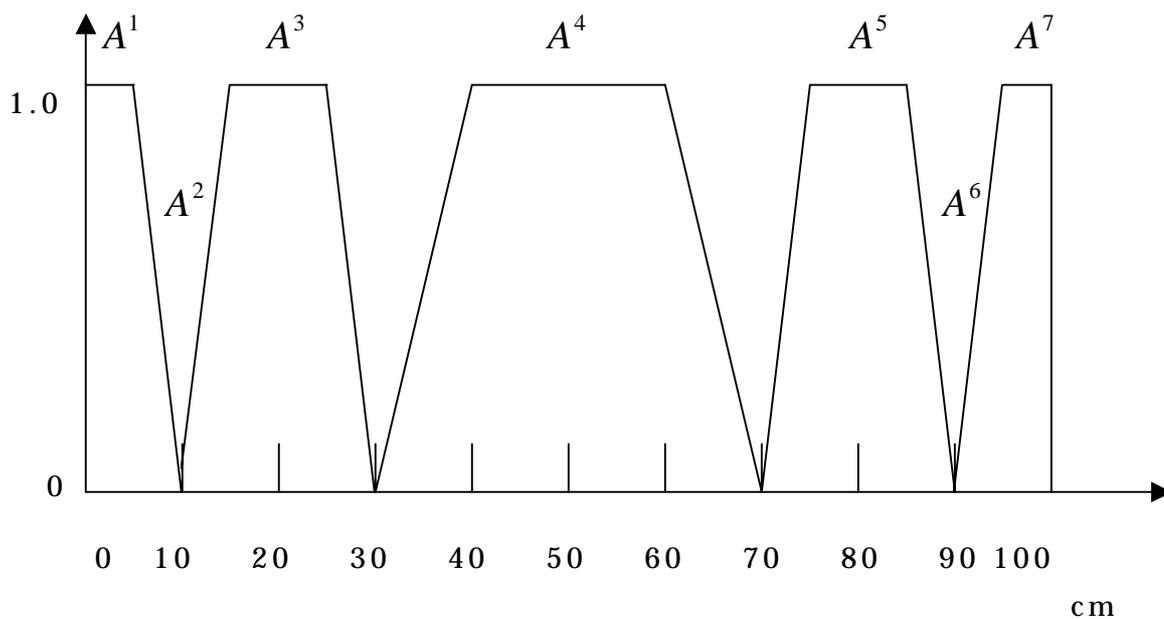


図 3.3 前件部のメンバーシップ関数(台形型)

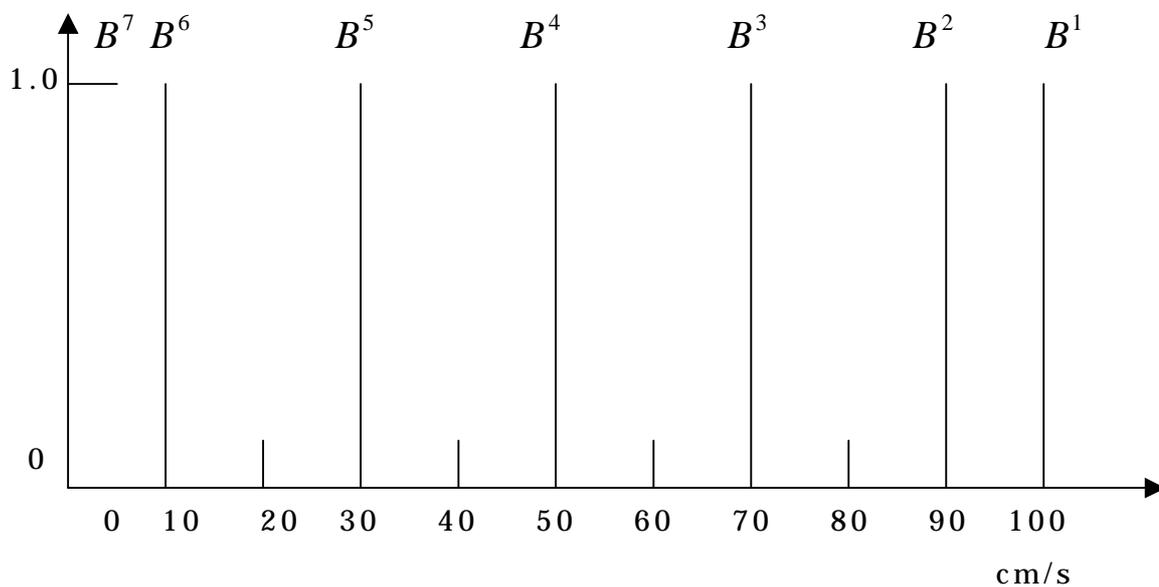


図 3.4 後件部  $y_B$  の実数値

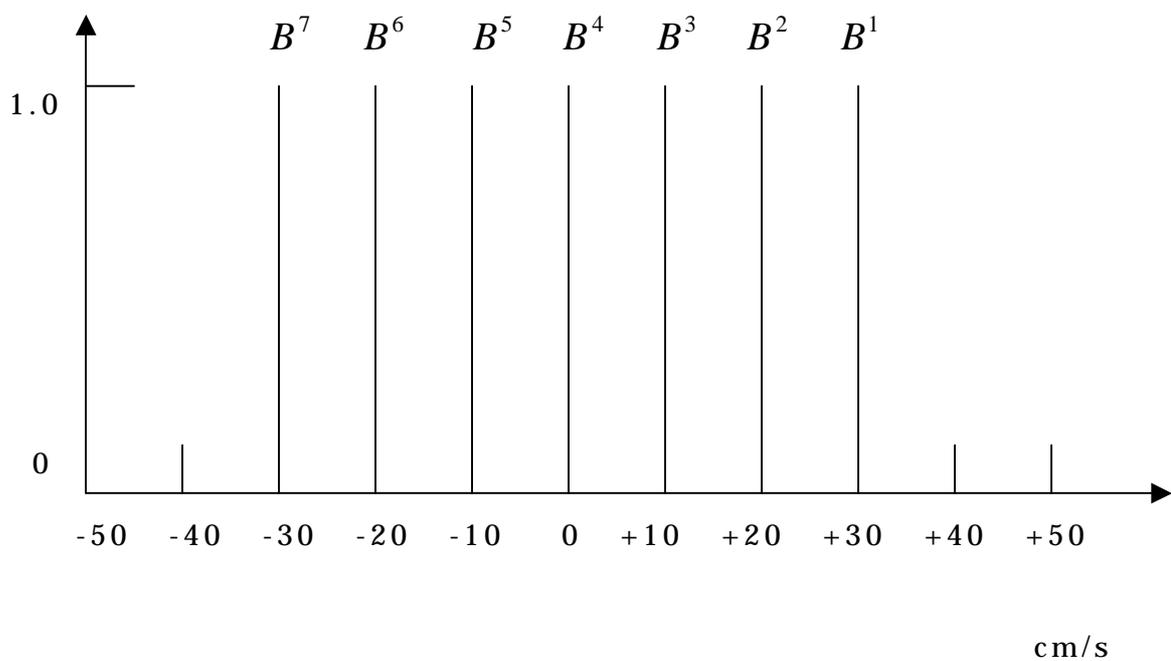


図 3.5 後件部  $y_C$  の実数値

### 3.7 軌道計画のアルゴリズム

ファジィ集合間の距離計算法について述べる。まず、色々な計算法を定義する事ができるが、以下の3つの公理を満たさなければならぬ。

$$d(A, B) = 0; d(A, B) = 0 \quad A = B \quad (3.1)$$

$$d(A, B) = d(B, A) \quad (3.2)$$

$$d(A, B) \leq d(A, C) + d(C, B) \quad (3.3)$$

距離型ファジィ推論法に基づいて軌道計画を行う。距離型ファジィ推論法では、前件部と事実との距離を推論の根拠としているので、前件部ファジィ集合が互いに覆われていないようなファジィルールの場合でも、適切な推論結果を得ることができる。また、後件部が凸なファジィ集合であれば、推論結果も凸なファジィ集合になるので、ファジィ数としての特徴は保持されている。さらに、分離規則は厳密に満たされているので、学習アルゴリズムにより容易に最適なルールを探すことができる。

軌道計画のアルゴリズムは次のステップからなる。今回のモデルはシングルトンである。

Step1: 事実と前件部との距離を求める。距離の計算は式(3.4)で与えられる。尚、 $A^i$ 、 $A$ は前件部、事実を表し、 $i$  = 推論規則の数、 $1 \leq p < \infty$ である。

$$\begin{aligned}
 d_i = d(A^i, A) = & \left[ \int_0^1 |\inf A^i_{M\alpha} - \inf A_{M\alpha}|^p d\alpha \right]^{\frac{1}{p}} \\
 & + \left[ \int_0^1 |\sup A^i_{M\alpha} - \sup A_{M\alpha}|^p d\alpha \right]^{\frac{1}{p}} \\
 & + \left[ \int_{-\infty}^{\infty} \left| \left( \frac{1}{M_{A^i}} - 1 \right) \mu_{A^i}(x) - \left( \frac{1}{M_A} - 1 \right) \mu_A(x) \right|^p dx \right]^{\frac{1}{p}}
 \end{aligned} \tag{3.4}$$

式(3.4)において， $p$ の値を2として一般的なファジィ集合間の簡略化距離計算式を与える。

Step2：次に軌道の結果を推論する。推論結果  $y_0$  は式(3.4)より求められた前件部と事実との距離  $d_i$  より式(3.5)で与えられる。尚， $B^i$  は後件部の実数値である。

$$y_0 = \frac{\sum_{i=1}^i \left[ B^i \prod_{j=1, j \neq i}^n d_j \right]}{\sum_{i=1}^i \prod_{j=1, j \neq i}^i d_j} \tag{3.5}$$

ここで求められた推論結果  $y_0$  を軌道値として与える。

---

## 第四章 軌道計画のシミュレーション

### 4.1 推論エンジンの特徴

この節では、Visual C++を用いて開発した推論エンジンの特徴について述べる。

推論法には第三章で述べた距離型ファジィ推論法を用いることで、従来の推論法では対応しきれなかった状況に対しても推論が可能となっている。

まず、今回の推論エンジンは、三角型ファジィ集合やシングルトンが表示出来るように、台形型ファジィ集合を表示出来るようにした。そして、ルール数を追加出来るようにした。図 4.1 に推論エンジンの画面と図 4.2 にルール数設定入力画面を表示します。

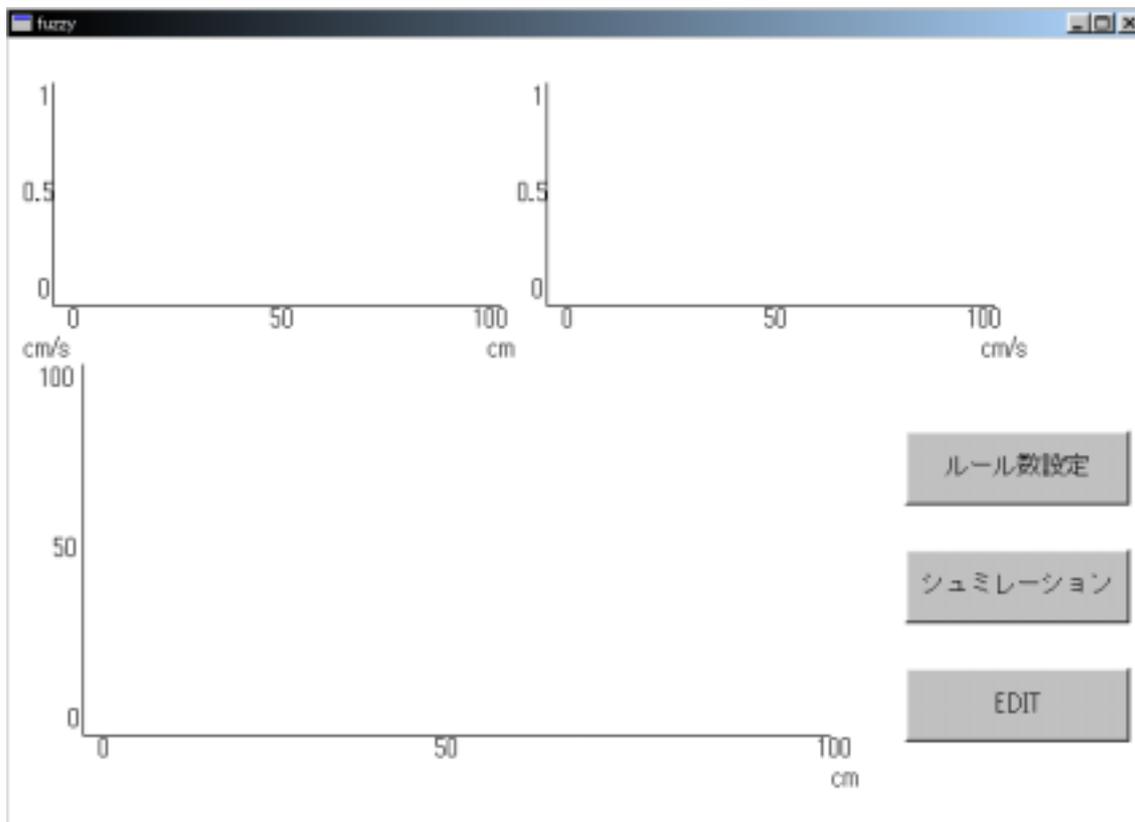


図 4.1 推論エンジンの画面



図 4.2 ルール数設定入力画面

## 4.2 シミュレーション内容

この章では、距離型ファジィ推論法を用いて設計されたファジィルールにおいて、事実に対する推論結果、つまり案内ロボットにおける軌道値をシミュレーションした。これにより設計されたファジィルールにおける距離型ファジィ推論法において、その軌道値の特性を知る事が出来る。このことは、案内ロボットの軌道計画法においてその特性と有効性を検証する上で大変有意義である。なお、本研究では、距離型ファジィ推論法に基づいた軌道値シミュレーションを、Visual C++で開発した。

シミュレーションモデルは案内ロボットの軌道計画法において超音波センサにより検出された人間との距離を1入力の事実としてこれに対するロボットの速度、つまり軌道値を1出力の推論結果とする。よって前件部は超音波センサにより検出された人間との距離である。また、後件部はロボットの速度である。なお、本シミュレーションでは前件部、後件部の範囲をそれぞれ0~100cm, 0~100cm/sとする。ルールの設定は台形型ファジィ集合で行う事とする。

### 4.3 シミュレーション結果

Sample :1 前件部に三角型ファジィ集合，後件部にシングルトンのルールを入力した時の推論結果を表示します．

図 4.3 の状態は，以下のようにルールを 2 つ与えて実行している状態の画像である．

RULE 1

IF X=TA1(0,0,0,50,1) THEN Y=TB1(20,20,20,20,1)

RULE 2

IF X=TA2(50,100,100,100,1) THEN Y=TB2(80,80,80,80,1)

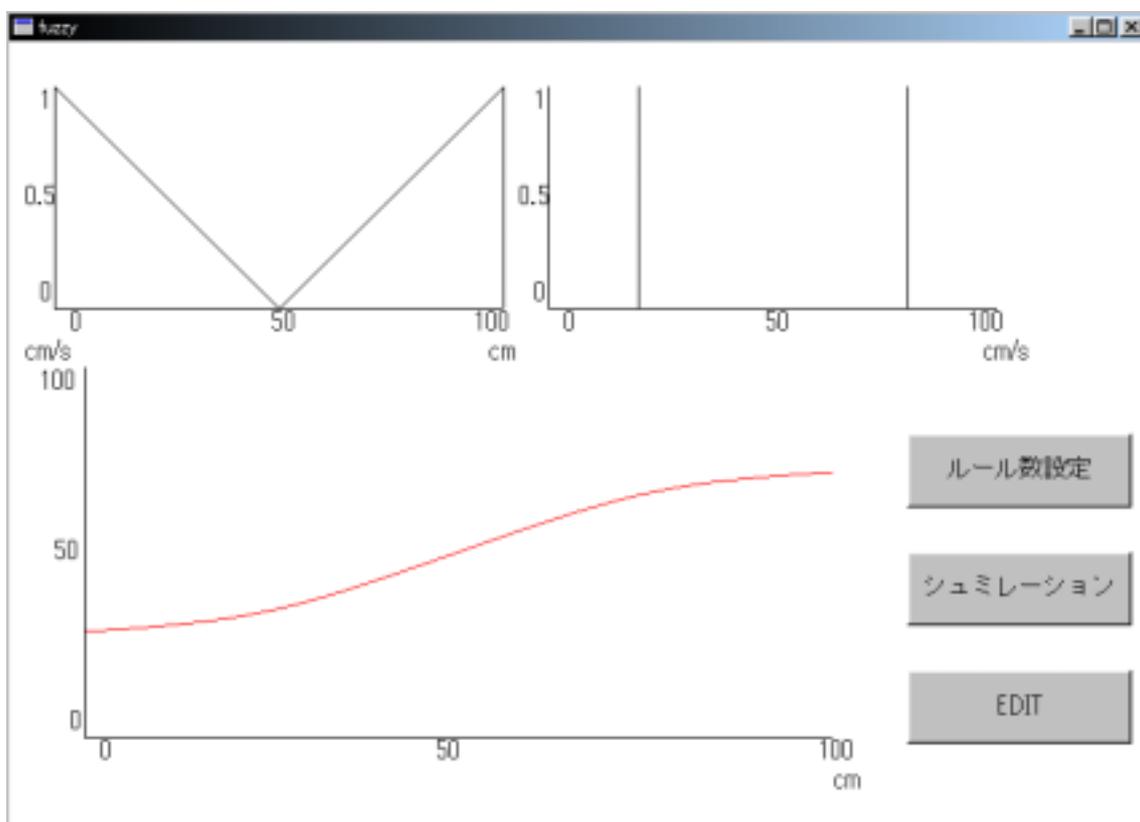


図 4.3 シミュレーション結果 1

Sample 2:前件部 , 後件部に三角型ファジィ集合 , ルールを入力した時の推論結果を表示します .

図 4.4 の状態は , 以下のようにルールを 2 つ与えて実行している状態の画像である .

RULE 1

IF X=TA1(0,0,0,40,1) THEN Y=TB1(100,100,100,60,1)

RULE 2

IF X=TA2(60,100,100,100,1) THEN Y=TB2(0,0,0,40,1)

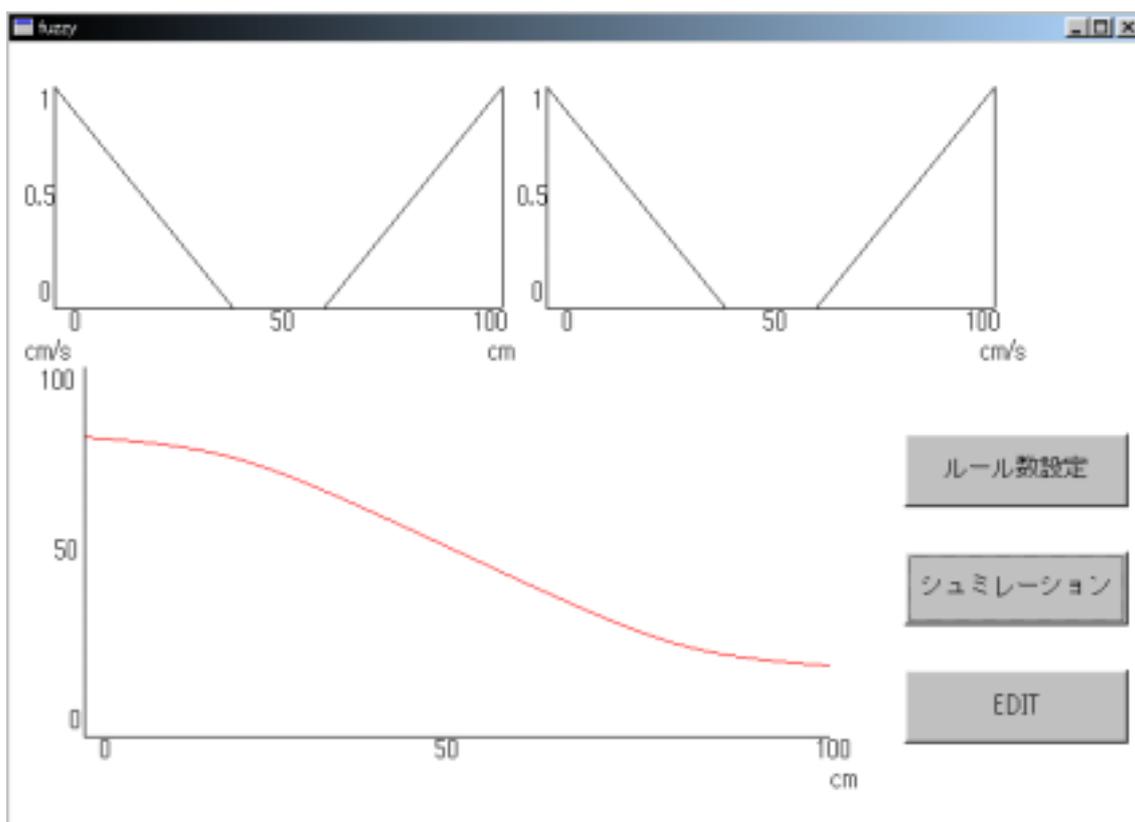


図 4.4 シミュレーション結果 2

Sample 3:前件部に三角型ファジィ集合，後件部にシングルトンのルールを入力した時の推論結果を表示します。

図 4.5 の状態は，以下のようにルールを 2 つ与えて実行している状態の画像である。

RULE 1

IF X=TA1(0,0,0,100,1) THEN Y=TB1(100,100,100,100,1)

RULE 2

IF X=TA2(0,100,100,100,1) THEN Y=TB2(0,0,0,0,1)

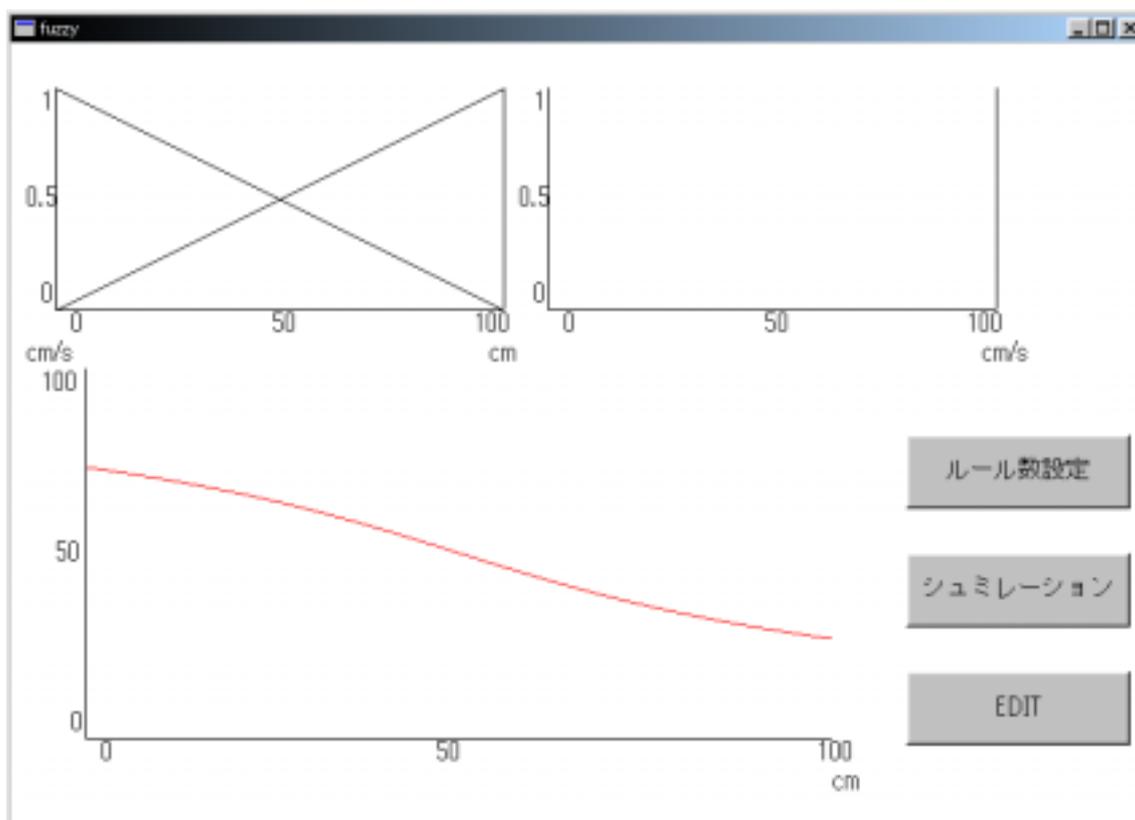


図 4.5 シミュレーション結果 3

Sample 4:前件部に三角型ファジィ集合，後件部にシングルトンのルールを入力した時の推論結果を表示します。

図 4.6 の状態は，以下のようにルールを 3 つ与えて実行している状態の画像である。

RULE 1

IF X=TA1(0,0,0,50,1) THEN Y=TB1(100,100,100,100,1)

RULE 2

IF X=TA2(25,50,75,1) THEN Y=TB2(50,50,50,50,1)

RULE 3

IF X=TA3(50,100,100,1) THEN Y=TB3(0,0,0,0,1)

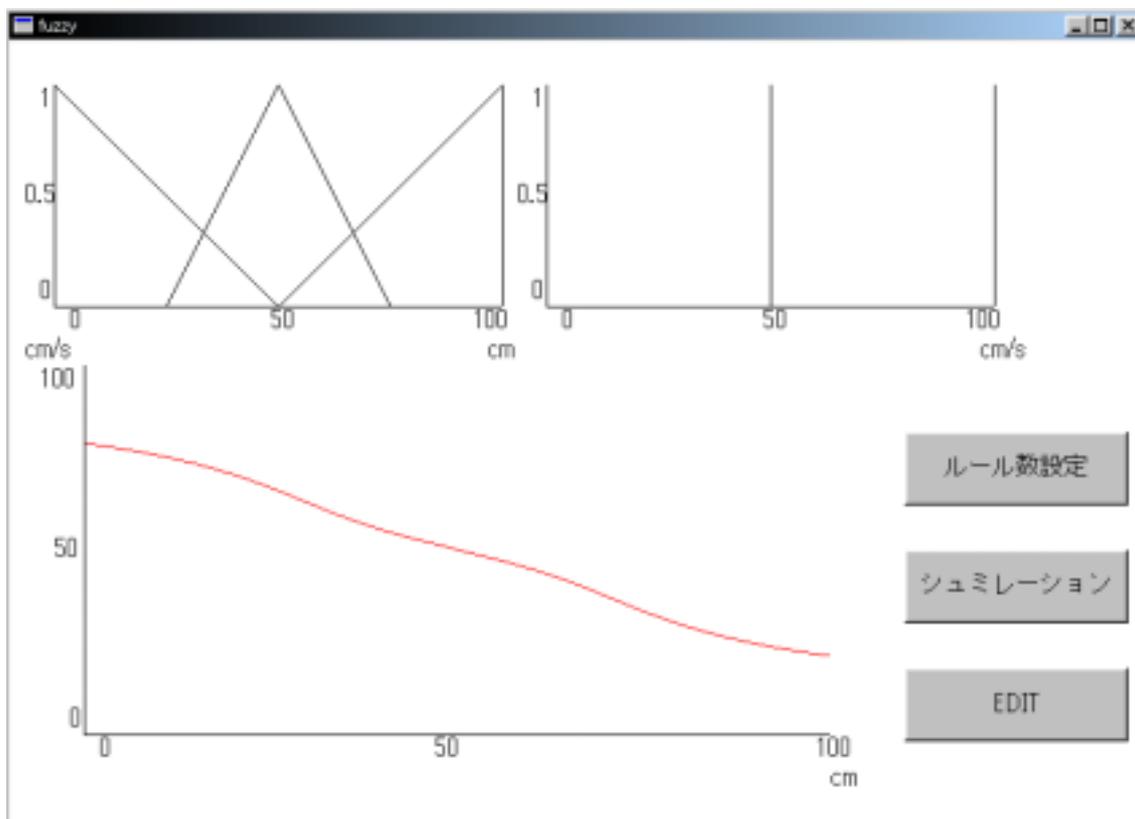


図 4.6 シミュレーション結果 4

Sample :5 前件部と後件部にシングルトンのルールを入力した時の推論結果を表示します。

図 4.7 の状態は、以下のようにルールを 2 つ与えて実行している状態の画像である。

RULE 1

IF X=TA1(0,0,0,0,1) THEN Y=TB1(100,100,100,100,1)

RULE 2

IF X=TA2(100,100,100,100,1) THEN Y=TB2(0,0,0,0,1)

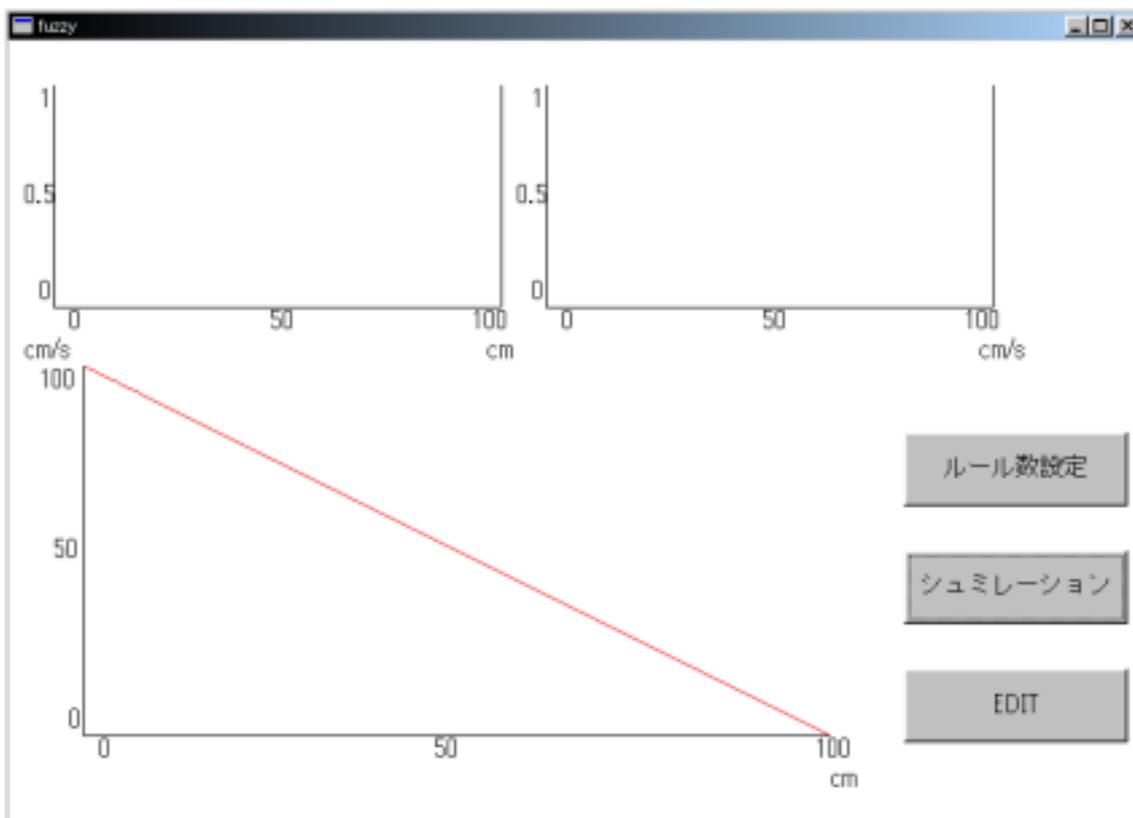


図 4.7 シミュレーション結果 5









