

# 遺伝的アルゴリズムによるファジィルールの自動チューニング

大木 誠・宮田 仁志・柴田 修吾\*  
森山 敏明・大北 正昭

電気電子工学科・\*関西日本電気株式会社

(1995年9月1日受理)

Self-tuning of Fuzzy Control Rules with the Genetic Algorithm

by

Makoto OHKI, Hitoshi MIYATA, Shugo SHIBATA\*  
Toshiaki MORIYAMA, Masaaki OHKITA

Department of Electrical and Electronic Engineering

\*NEC Corporation Kansai, Ltd.

(Received September 1, 1995)

We have developed an autonomous mobile robot "DREAM-1" controlled by the fuzzy theory. For its travelling control of DREAM-1, fuzzy control rules have to be determined. For its determination, the genetic algorithm is used. In this report, the genetic algorithm is outlined and applied to a self-tuning of fuzzy rules for travelling control of an autonomous mobile robot at a course of a crossing. For its tuning of fuzzy rules, two kind of membership functions, i.e., bell - and triangular - shaped membership functions, are used as the mathematical basic functions and their computational procedures are discussed with simulation results.

Key Words : Autonomous mobile robot, Fuzzy control, Self-tuning, Genetic algorithm, Computer simulation.

## 1. はじめに

ファジィ制御は、1965年にカリフォルニア大学のザデー教授により発表されたファジィ理論の応用分野として早くから注目され、多くの成功を収めている。ファジィ制御は、入出力関係を伝達関数などで記述する、いわゆる自動制御の手法とは異なり、制御アルゴリズムを if-then 形式のプロダクション・ルールで記述する一種のエキスパート・システムである。

我々は、このファジィ制御に注目し、自律移動ロボットの走行制御に導入することにより、種々のコースを走行するロボットの製作にすでに成功している[1]。ファジィ制御を用いることによりロバスト性が得られ、ある程度の外界の変化、例えばコースの形状変化などに対応できることが一般に知られている。しかしながら、固定的なルールだけでは外界の変化に対応しきれない場合がある。我々の研究対象である自律移動ロボットの場合でいうと、外界の状況が、制御ルールが許容している対応可能なスタート位置や、コースの幅などを超えた状態などが想定できる。そのような場合でも迅速に状況に応じて制御ルールを調整する必要が生じる。そこで、我々はこれまでに、最急降下法やパターン探索法などによりルールの自動チューニングを試みた[2][3]。ここでは、遺伝的アルゴリズム (Genetic Algorithm, GA) による自動チューニング[4]を行った結果について報告する。

まず、2章ではファジィ制御への遺伝的アルゴリズムの導入について述べ、3章では遺伝的アルゴリズムの理論について述べる。また、4章では本手法を我々の自律移動ロボットに適用し、計算機シミュレーションを行った結果について述べる。

## 2. 制御ルールとメンバーシップ関数の構成

遺伝的アルゴリズムとは、最適化したい数値対象をあらかじめ遺伝子コード化しておいて、これを突然変異や交叉などの操作を加えて変化させたものについて評価し、良いものを残すという手法である。最急降下法やパターン探索法などと大きく異なる点は、解空間の状態によらず探索点が連続的に移動せず、確率的要因によるところが大きい点と、良い因子を保存できる可能性が高い点である。このため、遺伝的アルゴリズムによると、計算時間に制限がなければ必ず局所解でない最適解に到達できることが知られている。

ここでは、自律移動ロボットの走行制御のために我々がとったルールの構成方法について説明し、その制御ルールを構成するメンバーシップ関数の定義について述べる。最後に、制御ルールに対応した遺伝子コードについて説明する。

## 2. 1 制御ルールの構成

本研究で用いたファジィ推論モデルでは、後件部が実数値を出力する出力関数で表されるものを用いた。いま、入力変数を  $x_1, x_2, \dots, x_m$  とし、出力変数を  $\theta$ 、出力関数を  $Y: [0, 1] \rightarrow \mathbb{R}$  とし、推論ルールは次式で表される。

$$\text{Rule } i: \text{if } x_1 \text{ is } A_{1i} \text{ and } x_2 \text{ is } A_{2i} \text{ and } \dots \text{ and } x_m \text{ is } A_{mi} \text{ then } \theta_i = Y_i(\mu_i) \quad (1)$$

ここで、 $A_{ji}$  は第  $i$  ルールにおける、第  $j$  変数  $x_j$  に関する前件部メンバーシップ関数

$A_{ji}: x_j \in \mathbb{R} \mapsto [0, 1]$ 、 $\mu_i$  は第  $i$  ルール前件部の適合度を表す。また、 $\theta, Y$  に添字  $i$  が付いているのは、第  $i$  ルールに関して定義されていることを示す。本研究では、"and" に相当する演算に、min-演算を用いているため、(1)は次式のように書き直せる。

$$\mu_i = \bigwedge_j A_{ji}(x_j) \quad , \quad \theta_i = Y_i(\mu_i) \quad (2)$$

全てのルールの適合度および出力値より、次式によって推論結果を算出する。

$$\theta = \frac{\sum_i \mu_i \theta_i}{\sum_i \mu_i} \quad (3)$$

本研究の自律移動ロボットの走行制御においては、このような推論法により前輪のステアリング切り角を決定する。

## 2. 2 メンバーシップ関数の定義

推論ルールで用いる前件部メンバーシップ関数および後件部出力関数の定義について説明する。本研究では、釣り鐘型メンバーシップ関数と三角型メンバーシップ関数を用い、比較を行う。

## (1) 釣り鐘型メンバーシップ関数

この関数は、中心値の左右で異なる標準偏差を持ったガウス基底として知られた関数である。まず、前件部メンバーシップ関数は次式で与えられる。

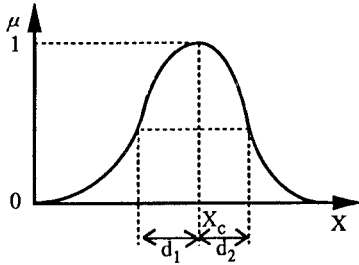
$$\mu = \begin{cases} \exp\left[-\frac{(x-x_c)^2}{2d_1^2}\right], & x < x_c \\ \exp\left[-\frac{(x-x_c)^2}{2d_2^2}\right], & x_c \leq x \end{cases} \quad (4)$$

メンバーシップ関数の概形は図1(a)に示す通りである。チューニングの対象となるパラメータは、中心位置  $x_c$  および2つの標準偏差  $d_1, d_2$  の3個である。これらの標準偏差によって、メンバーシップ関数の広がり方が決定する。

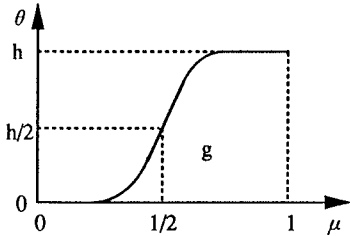
次に、後件部出力関数を定義する。

$$\theta = \frac{h}{1 + \exp[-g(\mu - 0.5)]} \quad (5)$$

出力関数の概形は図1(b)に示す通りである。チューニングの対象となるパラメータは、高さすなわち飽和値を与える  $h$  と、傾きを与える  $g$  の2個である。



(a)前件部メンバーシップ関数



(b)後件部出力関数

図1 釣り鐘型メンバーシップ関数

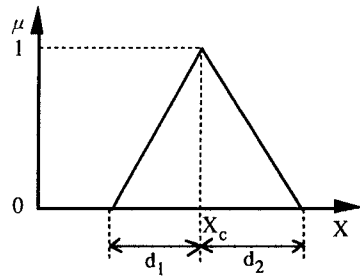
(2) 三角型メンバーシップ関数

このタイプの関数は、ファジィ制御の分野では広

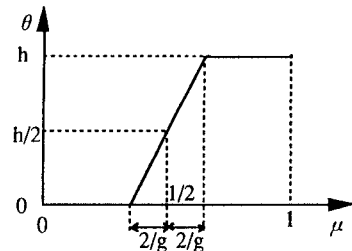
く用いられている関数である。左右で非対称な底辺を有し、定義域を4つの区間に区分して定義される。このうち、0以外の値を有する区間は2つだけで、しかも簡単な1次式で定義できるため計算も容易である。前件部メンバーシップ関数は次式で与えられる。

$$\mu = \begin{cases} \frac{1}{d_1}x + 1 - \frac{x_c}{d_1}, & x_c - d_1 \leq x < x_c \\ -\frac{1}{d_2}x + 1 + \frac{x_c}{d_2}, & x_c \leq x < x_c + d_2 \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

メンバーシップ関数の概形は、図2(a)に示す通りである。



(a)前件部メンバーシップ関数



(b)後件部出力関数

図2 三角型メンバーシップ関数

チューニングの対象となるパラメータは三角形の頂点の位置  $x_c$  と、各斜辺と横軸が交わる位置  $d_1, d_2$  の3個である。

次に、後件部出力関数は、釣り鐘型で示した後件部出力関数にならって段差型とした。3つの区間でそれぞれ直線的に定義され、しかも、0もしくは  $h$  以外の値をとる区間が1つだけである。後件部出力関数は次式で与えられる。

$$\theta = \begin{cases} 0 & , \mu < \frac{1}{2} - \frac{2}{g} \\ \frac{gh}{4} \left( \mu - \frac{1}{2} \right) + \frac{h}{2} & , \frac{1}{2} - \frac{2}{g} \leq \mu < \frac{1}{2} + \frac{2}{g} \\ 1 & , \frac{1}{2} + \frac{2}{g} \leq \mu \end{cases} \quad (7)$$

関数の概形を図2(b)に示す。式(7)からもわかる通り、(5)のg, hと同値であれば中心区間の傾きは(5)の $\mu=1/2$ の点での傾きと同じになる。チューニングの対象となるパラメータは、出力関数の高さを与えるhと傾きに寄与するgの2個である。

2. 3 制御ルールの遺伝子コード化

本研究で行った制御ルールの遺伝子コード化について述べる。突然変異や交叉などの遺伝的操作は、

この遺伝子コードに対して行われる。ここでは、釣り鐘型メンバーシップ関数を用いた場合について説明するが、三角型メンバーシップ関数を用いた場合でも同様である。

図3に遺伝子コード化の概要を示す。ここでは、実際にロボットの制御で用いた入力変数と同じ4つの変数 $x_1, x_2, x_3, \phi$ と出力変数 $\theta$ を想定している。

まず、4つの入力変数のそれぞれに対する前件部メンバーシップ関数 $A_{1i}, A_{2i}, A_{3i}, A_{\phi i}$ の形状を決めるパラメータ各3個を抜き出す。前件部の計12個のパラメータは各9bit、後件部の2個のパラメータは7bitで、それぞれ10進-2進変換する。これらを並べた144bitを1個のルールの遺伝子とする。n個のルールを用いる場合だと、144nbitの遺伝子コードとなる。三角型の遺伝子コードも同様に144nbitとなる。

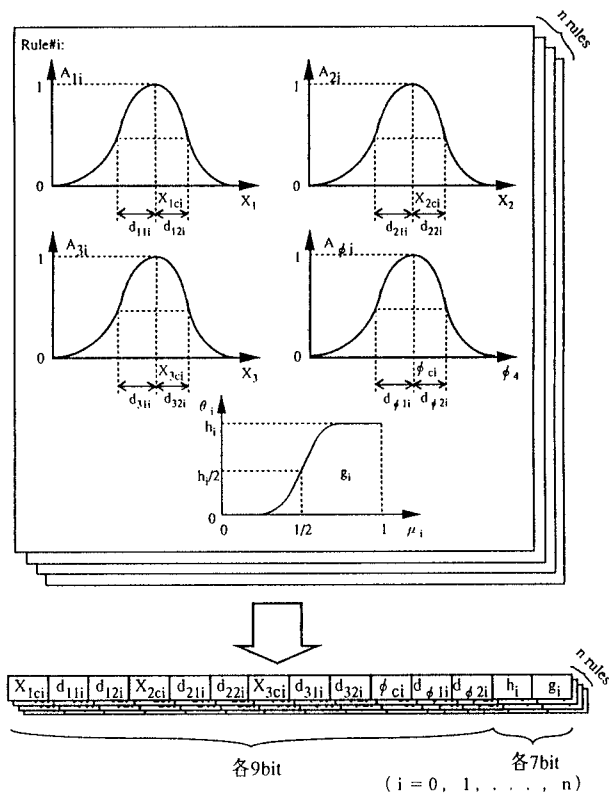


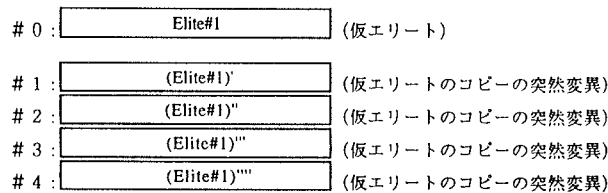
図3 遺伝子コード化

3. 遺伝的アルゴリズム

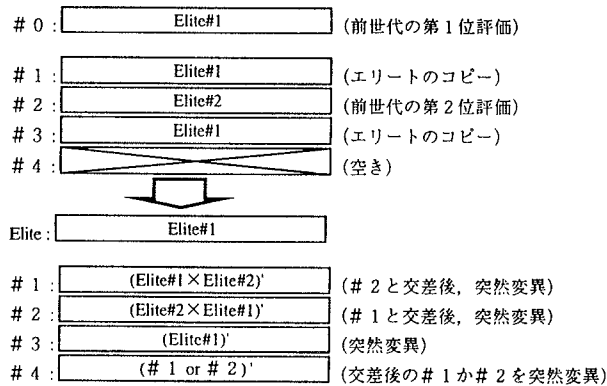
遺伝的アルゴリズム[5][6]では、評価の良いものを残しておくためのエリート遺伝子フィールドとトライ・アンド・エラーのための探索遺伝子フィールドが必要となる。エリート遺伝子フィールドを複数用意する手法や、多くの探索遺伝子フィールドを用意する手法があるが、本研究ではエリート遺伝子フィールドを1個、探索遺伝子フィールドを4個とした。探索遺伝子フィールドは多ければ多いほど早い世代で終了する傾向にあるが、その分1世代での無駄な操作も増大する。本研究では、自律移動ロボット上で、出来るだけ実時間に近い速度でチューニングすることを目標としているため、このような比較的少ないフィールド数で学習することにした。

図4(a),(b)に遺伝的アルゴリズムの概要を示す。初期個体の形状をコード化したものを4個コピーし、それぞれ突然変異させることで行った。ここで行った突然変異は、全てのルールを選択し、ルール内の各ビットについて25%の確率でビット反転させた。

第1世代はエリート初期値とそれを突然変異させた4個の探索遺伝子の計5個のルールについて評価し、第1位評価のものを次世代のエリートとし、第2位評価のものも次世代で利用する。第2世代以降は、探索遺伝子の4個について評価して、同様に第1、2位を保存する。第2世代以後では、各世代の探索終了後、第1位評価のものをエリート遺伝子フィールドに保存する。探索遺伝子フィールドの第1フィールドと第3フィールドにエリートをコピーしておく。また、前世代第2位のものを第2フィールドにコピーしておく。次に、第1フィールドと第2フィールドの間で交叉させ、各々突然変異させる。第3フィールドは単独で突然変異させる。交叉後の第1フィールドか第2フィールドのいずれかを50%ずつの確率で選択し、突然変異させたものを第4フィールドにおく。ここで用いた突然変異は、遺伝子内の各ルールが25%の確率で選択され、各ルール内の各ビットが25%の確率でビット反転するものである。



(a) 第1世代での初期個体の生成



(b) 第2世代以後での遺伝的操作

図4 遺伝的アルゴリズム

4. 計算機シミュレーション

4. 1 4輪モデル

移動ロボットは4輪型とし、低速度で走行するものとする。すなわち、旋回運動時に遠心力による車輪の横滑りなどは無視できると考える。図5に前進走行時の理想的な4輪モデルの旋回運動を示す[7]。これにより、各車輪の接地点は1点Oを中心に回転し、次の各式で与えられる回転半径を有することになる。

$$R_1=L/\sin \theta \tag{8}$$

$$R_2=L/\sin \theta' \tag{9}$$

$$R_3=R_1 \cos \theta -(B_f-B_r)/2 \tag{10}$$

$$R_4=R_2 \cos \theta'+(B_f-B_r)/2 \tag{11}$$

$$\theta'=\tan^{-1}[L/(B_f+L \cot \theta)] \tag{12}$$

ここで、 $\theta$ は内側前輪のステアリング切り角、 $\theta'$ は外側前輪のステアリング切り角、 $B_f$ 、 $B_r$ はそれぞれ前、後輪のトレッド長を示している。また、 $L$ はホイールベースを示している。

以上のような走行モデルに基づいて、計算機上で移動ロボットの走行軌跡を算出することが出来る。

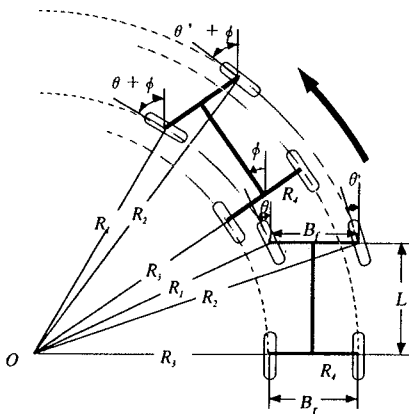


図5 4輪モデル

4. 2 コースの設定

本研究では、移動ロボットが走行するコースとして交差点を取り上げた。図6に交差点コースを示す。図6に示すように、移動ロボットはコースに対して、自身の左右方向のコース境界までの距離、および交

差点進入時の前方信号機までの距離を計測できるものとする。特に、コース終端に向かっては、障害物はないものとし、この方向への距離計測は最大値の300となるものとする。(すなわち、位置\*2、\*3の前方距離 $x_3$ 、および交差点内でこれらの方向に $x_1$ 、 $x_2$ を計測しようとした場合に300となる。)

移動ロボットが走行制御するために用いる入力変数は、図6中の $x_1$ 、 $x_2$ 、 $x_3$ および車体姿勢角 $\phi$ とする。また、制御の対象である出力はステアリング切り角 $\theta$ とする。移動ロボットは、一定の速度で進行するものとする。

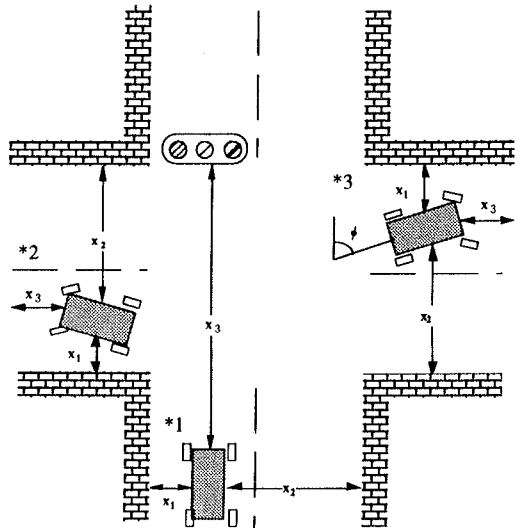


図6 交差点コース

4. 3 走行の評価

移動ロボットの走行制御のためのルールを学習するためには、評価指標が必要となる。本研究では図7に示すように、交差点の入り口付近と交差点の出口付近において評価関数を定義する。学習においては、この評価関数の値がより小さくなるものを良好因子として位置づける。左折学習では、次式によって評価関数を定義する。

$$f_{1a}=2|x_1-24|+3|\phi-0| \tag{13}$$

$$f_{2a}=3|x_1-24|+2.5|\phi-90| \tag{14}$$

$$f_a=f_{1a}+f_{2a} \tag{15}$$

これにより、移動ロボットが交差点入り口(評価点

1)において、やや左より ( $x_1=24$  の点)で真直 ( $\phi=0$ ) に進入し、交差点出口付近 (評価点2) において、やや左より ( $x_1=24$  の点) で、真直 ( $\phi=90$ ) に進入する場合を目標として定義している。

右折走行学習時では、次によって評価関数を定義する。

$$f_{1c}=3|x_1-64|+2|\phi-0| \quad (16)$$

$$f_{2c}=3|x_1-64|+2|\phi-90| \quad (17)$$

$$f_c=f_{1c}+f_{2c} \quad (18)$$

これにより、移動ロボットが交差点入り口 (評価点1) において、やや右より ( $x_1=64$  の点) で真直 ( $\phi=0$ ) に進入し、交差点出口付近 (評価点2) において、やや右より ( $x_1=64$  の点) で真直 ( $\phi=90$ ) に進入する場合を目標として定義している。

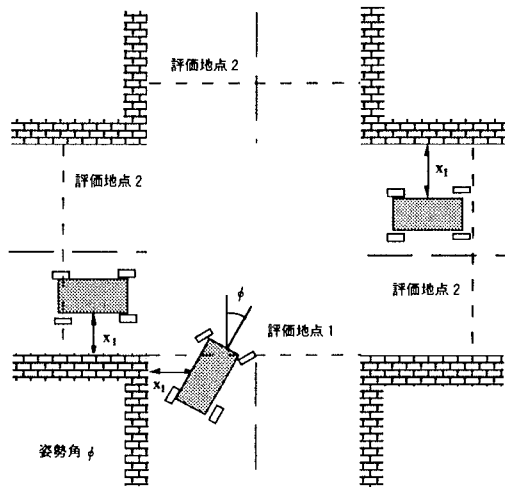


図7 評価関数

#### 4. 4 終了条件

走行終了条件は、移動ロボットがコース境界に接触した時点、経路終端に達した時点、および交差点内を円運動するなどして車体の2.5倍の距離を走行した時点をもって走行終了とする。走行終了の時点をもって、前述の評価値を算出し、走行の評価とする。

次に学習終了の条件は、いずれかの遺伝子による走行の評価が70以下になった場合をもって学習を終了し、その遺伝子による入力メンバーシップ関数

および出力関数を学習結果とする。

#### 5. シミュレーション結果

今回、我々は釣り鐘型メンバーシップ関数と三角型メンバーシップ関数の2種類のメンバーシップ関数について、交差点コースを走行する移動ロボットをモデルとして、その制御ルールの学習を行った。

まず、釣り鐘型メンバーシップ関数を用いた場合の初期値となるメンバーシップ関数の形状を図8に示す。今回は、6個のルールで制御する場合を扱った。この初期値メンバーシップ関数は特に意図したものではなく適当な値を人間が設定したものである。

釣り鐘型メンバーシップ関数を用いた場合で、左折走行を学習した場合の学習結果の例を図9に、そのメンバーシップ関数による走行軌跡の例を図10に示す。各メンバーシップ関数の形状が初期値 (図8) に比べて大きく変化していることがわかる。

次に、三角型メンバーシップ関数を用いた場合の初期値となるメンバーシップ関数の形状を図11に示す。釣り鐘型と同様に、6個のルールにより制御する場合を対象とした。図8と比べてわかるように、釣り鐘型と同様のパラメータ値を初期値としている。これにより、メンバーシップ関数の中心位置や  $1/2$  をとる点の横軸の位置が等しくなる。

三角型メンバーシップ関数を用いて左折走行を学習した場合の学習結果の例を図12に、そのメンバーシップ関数による走行軌跡の例を図13に示す。

また、釣り鐘型メンバーシップ関数および三角型メンバーシップ関数を用いて右折走行学習させた結果の走行軌跡の例を図14、15に示す。

最後に、釣り鐘型メンバーシップ関数および三角型メンバーシップ関数を用いた場合の学習を31回行い、それぞれの平均学習終了世代数を表1に示す。

表1 学習終了世代数の比較

	平均終了世代数 [世代]
釣り鐘型MSF	655
三角型MSF	427

表からわかるように、三角型メンバーシップ関数を用いた場合の方が早い世代で学習を終了する傾向にあるといえる。また、それぞれのメンバーシップ関数の定義式から、三角型メンバーシップ関数を用いた場合の方が少ない計算ステップ数で制御 (学習時での試行錯誤的な走行も含めて考えることが出来る) が行えることがわかる。従って、本研究の例では釣り鐘型メンバーシップ関数を用いた場合より、三角型メンバーシップ関数を用いた場合の方が優位であるといえる。

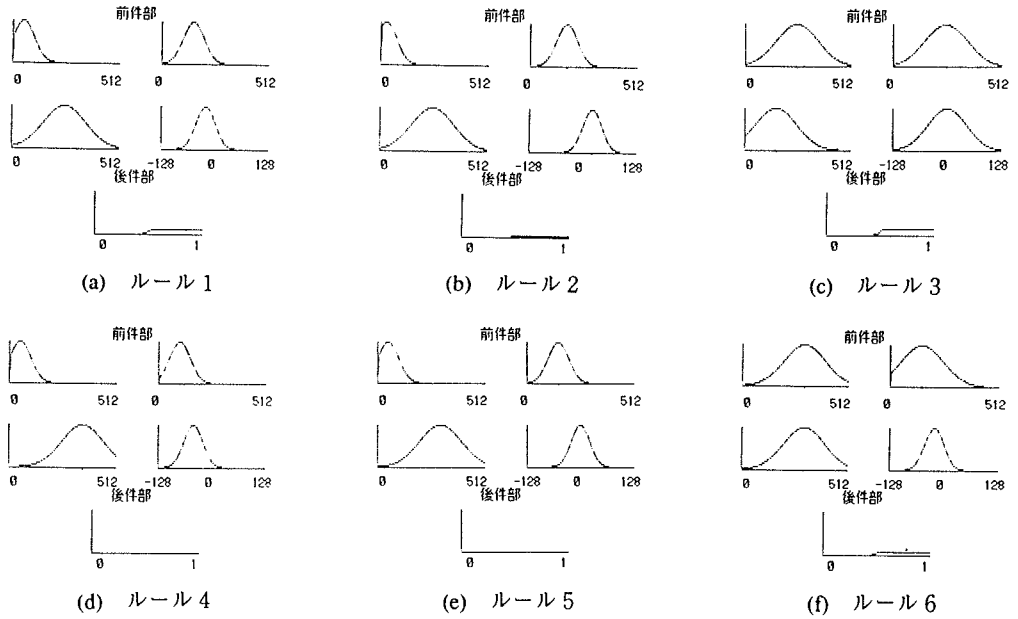


図 8 釣り鐘型メンバーシップ関数の初期値

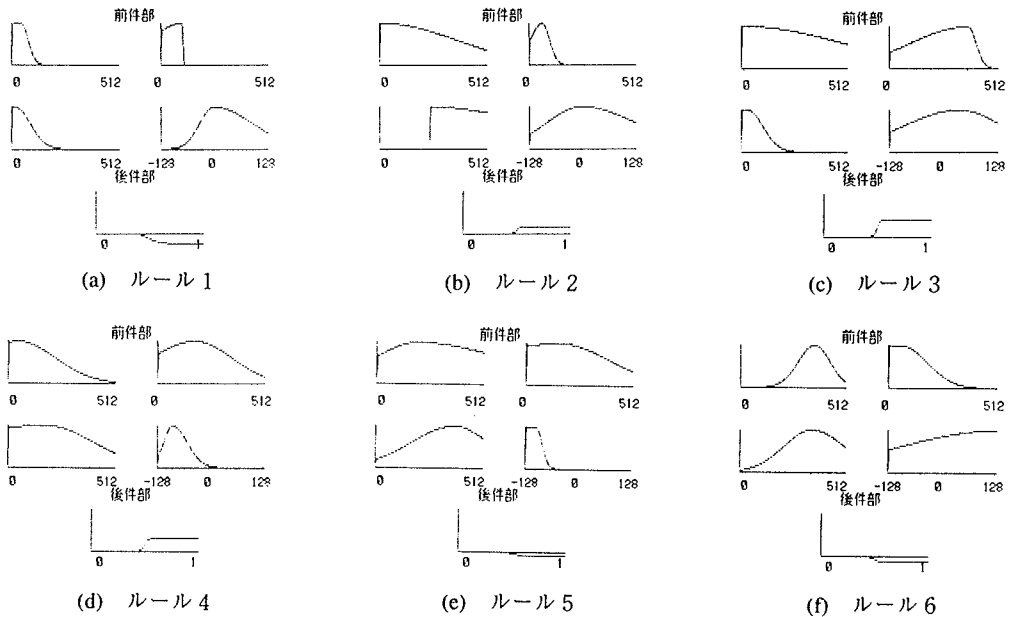


図 9 左折走行学習後の釣り鐘型メンバーシップ関数



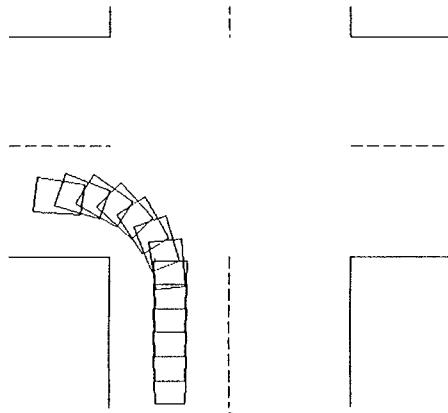


図10 左折走行学習後の釣り鐘型メンバーシップ関数による走行軌跡

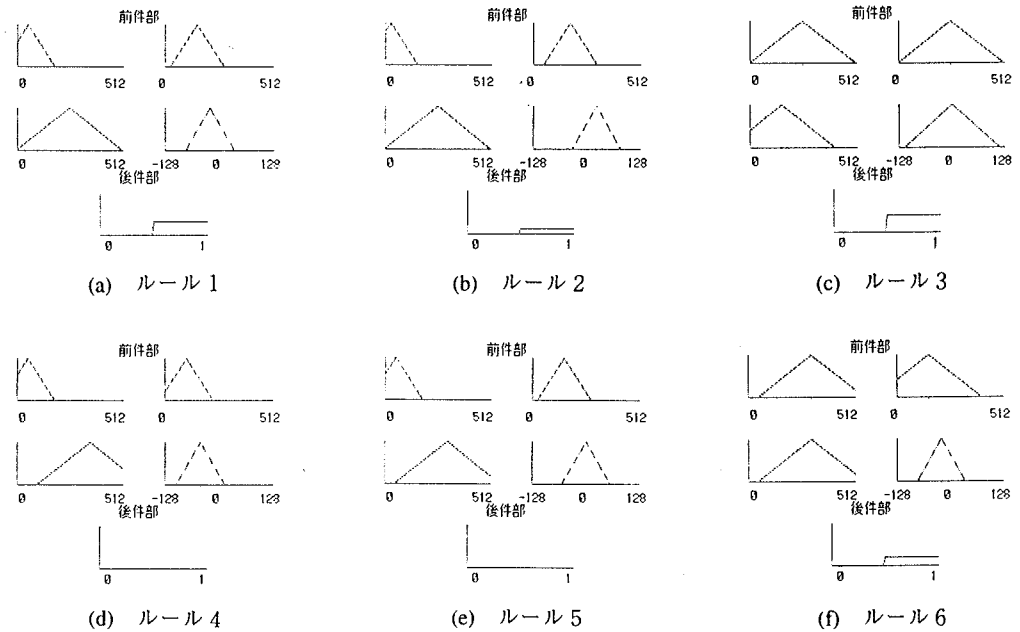


図11 三角型メンバーシップ関数の初期値

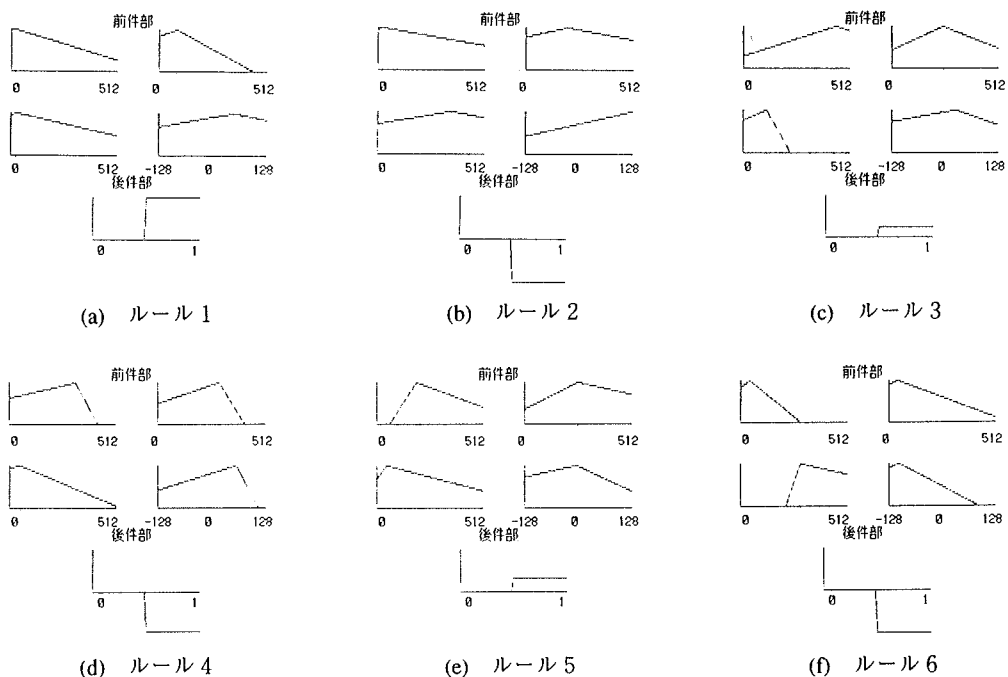


図 1 2 左折走行学習後の三角型メンバーシップ関数

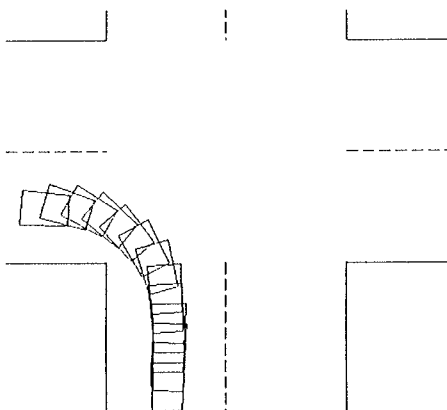


図 1 3 左折走行学習後の三角型メンバーシップ関数による走行軌跡

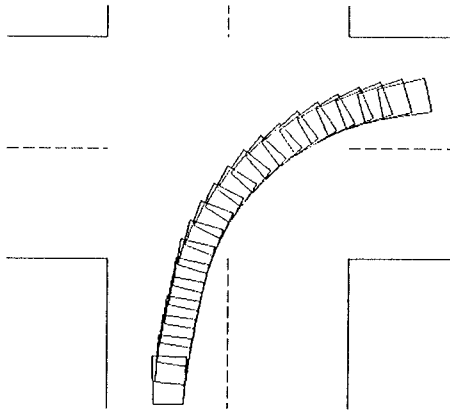


図14 右折走行学習終了後の釣り鐘型メンバーシップ関数による走行軌跡

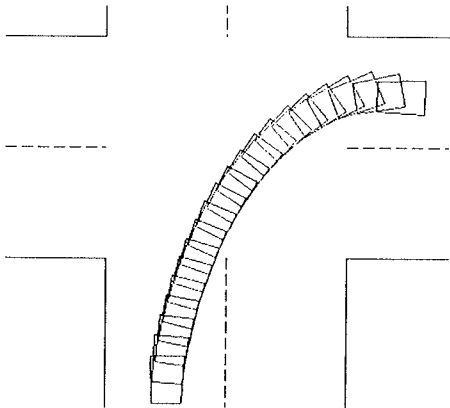


図15 右折走行学習終了後の三角型メンバーシップ関数による走行軌跡

## 6. おわりに

本研究において、4輪走行ロボットのファジィ制御ルールを自動チューニングするために、遺伝的アルゴリズムを適用した。ここで、我々は釣り鐘型メンバーシップ関数と三角型メンバーシップ関数の2種類のメンバーシップ関数について比較した。その結果、三角型メンバーシップ関数の方が有利であるという傾向を得た。

しかしながら、今回は31という比較的小さい標

本数しか得られておらず、今後更に多くの学習を行うことで統計的な傾向を明らかにする必要がある。また、突然変異率や交叉率の自動変更や、突然変異率の与え方などの遺伝的アルゴリズムの改良にも取り組む必要がある。また、得られたファジィ制御ルールの汎化能力についてもあわせて検討して行くつもりである。

## 謝辞

本論文を執筆するにあたり、大学院博士前期課程の田中幹浩氏に多大な御助力を賜りましたことを深く感謝いたします。

## 参考文献

- [1] M.Ohkita, H.Miyata, M.Miura, H.Kouno, Y.Yokouchi and S.Shibata: "Fuzzy Control of an Autonomous Mobile Robot for a Parallel Parking," Proc.of the 14th IMACS World Congress (July 11-15, Georgia Inst. of Tech, Atlanta, GA, USA), Vol.1, pp.404-406 (1994).
- [2] 宮田, 大木, 田, 柴田, 田中, 大北: "自律移動ロボットの走行制御におけるファジィルールの自動チューニング —降下法ファジィ制御を用いる場合—", 鳥取大学工学部研究報告, 25巻, 1号, pp.99-107 (1994).
- [3] 宮田, 大木, 川久保, 柴田, 田中, 大北: "自律移動ロボットの走行制御におけるファジィルールの自動チューニング —パターン探索法を用いる場合—", 鳥取大学工学部研究報告, 25巻, 1号, pp.109-116 (1994).
- [4] 宮田, 大木, 柴田, 田中, 大北: "自律移動ロボットの走行制御における遺伝的アルゴリズムを用いたファジィルールの自動チューニング", 第11回ファジィシステムシンポジウム講演論文集, FB2-2, pp.623-626 (1995).
- [5] 安居院猛, 長尾智晴: ジェネティックアルゴリズム, (株) 昭晃堂 (1993).
- [6] 古橋, 中岡, 内川: "人工生命的アプローチによるファジィ知識獲得", 第10回ファジィシステムシンポジウム講演論文集, TD4-2, pp.363-366 (1994).

