ファジィ・ガウシアン・ニューラルネットワーク制御器と その移動ロボット車の制御への応用*

湯 軍^{*1}, 渡 辺 桂 吾^{*2}
中 村 政 俊^{*3}, 古 賀 信 之^{*3}

Fuzzy Gaussian Neural Network Controller and Its Application to the Control of a Mobile Robot

Jun TANG, Keigo WATANABE,
Masatoshi NAKAMURA and Shinji KOGA

This paper describes a new fuzzy neural network (FNN) controller in which a Gaussian function is applied as an activation function, referred to here as a fuzzy Gaussian neural network (FGNN) controller. The learning architecture adopted is specialized so that we can tune the membership functions without using an expert's manipulated data. As an example of application, a tracking control problem for the speed and azimuth of a mobile robot driven by two independent wheels is solved using the FGNN controller. In order to facilitate the application of an FGNN controller to multi-input multi-output systems, we then propose a learning controller consisting of m FGNNs based on independent reasoning and a connection network, where m denotes the order of output of the control object. The effectiveness of the proposed method is illustrated using computer simulation.

Key Words: Fuzzy Set Theory, Mobile Robot, Neural Network, Speed and Azimuth Control, Fuzzy Control, Learning Control, Mechatronics and Robotics

1. は じ め に

近年, 移動ロボット車の自律機能の高度化を目指して, 最適制御やファジィ制御などが広く応用されている^{(1)~(3)}。特にファジィ制御はオペレータのもつ制御対象制御のための定性的な制御戦略を IF-THEN 型の言語的制御則で表現でき, 制御対象の厳密な定量的モデルが得られなくても, 制御対象の入出力の定性的関係のみで制御則を構成できる。しかし, 単一入出力系は別として, 多入出力系のファジィ制御アルゴリズムを作成するとき, そのアルゴリズムを IF-THEN 型の制御ルールとして正確に把握することは非常に困難である。また, ファジィ制御器を構成した後のメンバシップ関数や入出力スケーラなどのパラメータ調整には, 試行錯誤を伴うことから, このようなパラメータ調整を自動化することも重要である。

以上のことから, 最近ニューラルネットワーク理論とファジィ推論を融合させたいわゆるニューロ・ファジィ制御⁽⁴⁾⁽⁵⁾が注目されている。これは, ニューロン

の学習機能を利用してファジィ推論のルールとメンバシップ関数の自動調整を行わせるものである。堀川ら⁽⁶⁾は, 熟練者の良好な操作パターンから制御ルールを同定でき, しかも前件部メンバシップ関数の微調整もできるファジィ・ニューラルネットワーク (FNN) 制御器を提案している。この手法は, ニューラルネットワーク内にファジィ制御ルールを構築する方法であり, いろいろな応用問題にも適用されている⁽⁷⁾⁽⁸⁾。しかし, 彼らの方法では, いわゆる一般化学習機構を採用していることから, 模範的な制御結果 (パターン) が手元にない場合は適用できない。また, 疑似台形型メンバシップ関数を二つの符号の異なるシグモイド関数の和で構成しているため (このことより, 堀川らの制御器をファジィ・シグモイダル・ニューラルネットワーク (Fuzzy sigmoidal neural network: FSNN) と呼ぶことにする), 前件部メンバシップ関数を構成するための余分な中間層が必要となり, またファジィ集合のラベル数を増やすに連れて, その中間層のユニット数が増大する欠点がある。同様な問題は, $\tan^{-1}$ をユニット関数として採用している水上・普家⁽⁹⁾の FNN 制御器においても起きる。さらに, 多入出力制御系においては, ラベル数と前件部への入力データ数から直接制御ルール数を求める場合, 各制御ルールの前件部適合

* 平成 4 年 10 月 23 日 第 2 回 FAN シンポジウムにおいて講演, 原稿受付 平成 4 年 10 月 19 日。

^{*1} 佐賀大学大学院。

^{*2} 正員, 佐賀大学理工学部 (〒840 佐賀市本庄町 1)。

^{*3} 佐賀大学理工学部。

度を求める中間層のユニット数は膨大な数になることから、何らかの有効なネットワークの構成法を考える必要もある。

本論文では、ファジィ制御ルールを多層ニューラルネットワークで実現する点では堀川らの考え方を踏襲し、メンバシップ関数をガウシアン関数で表現し、そのネットワークの学習においては特殊化学習機構を採用する、ファジィ・ガウシアン・ニューラルネットワーク (Fuzzy gaussian neural network: FGNN) 制御器を提案する。本制御器は、前件部メンバシップ関数を構成するための余分な中間層を排除することができ、また熟練者の良好な制御操作パターンを入手する必要はない。また、本制御器の応用として移動ロボット車の速度と方位の軌道追従制御問題を扱う。両輪独立駆動型移動ロボット車に対して、多入出力系の FGNN の計算の簡単化の方法として、速度情報から左駆動力、方位情報からの右駆動力を推論するような独立推論では、速度と方位を同時に制御できないと思われることから、ここでは独立推論結果を結合させる結合ネットを追加する方法を示す。

以下では、まず提案するファジィ・ガウシアン・ニューラルネットワークによる学習制御器の構成と学習法について述べる。次に、2 輪独立駆動型移動ロボットのモデルについて述べ、左右輪の制御トルクと制御対象からの出力情報との関係を得るために、移動ロボットの速度と方位を用いて逆動力学モデルを利用するトルク算出制御則を示す。さらに、移動ロボット車に対する FGNN 制御システムの構成を説明する。最後に、提案した FGNN 制御器の有効性をシミュレーションで調べる。

2. ファジィ・ガウシアン・ニューラルネットワーク制御器

2.1 構成方法 ニューラルネットワークでファジィ制御ルールを実現しやすくするために、本稿では Mamdani の方法^{(10)~(12)}を簡略化した、つまり後件部が定数値をとる「簡略化ファジィ推論」⁽¹³⁾を利用する。ただし、各制御ルールにおける前件部適合度の計算はミニマム演算の代りに代数積とする。この方法では、 n 個の入力変数 ($x_1, \dots, x_n$)、 p 個の出力変数 ($u_1, \dots, u_p$) の場合、任意の i 番目の制御ルールは以下のように書く。

$$\begin{aligned} R_i: & \text{If } x_1 = A_{i1} \text{ and } \dots \text{ and } x_n = A_{in} \\ & \text{then } u_1 = B_{i1} \text{ and } \dots \text{ and } u_p = B_{ip} \end{aligned} \quad (1)$$

ここで、 R_i は i 番目の制御ルール、 A_{ij} は i 番目の制御ルールにおける j 番目の入力変数に対応する前件部

ファジィ集合(あるいはファジィ数)を表し、 B_{ij} は i 番目の制御ルールにおける j 番目の出力変数に対する定数値である。前件部適合度 h_i はメンバシップ関数の n 個の適合度 $\mu_{A_{i1}}(x_1), \dots, \mu_{A_{in}}(x_n)$ を用いて

$$h_i = \mu_{A_{i1}}(x_1) \mu_{A_{i2}}(x_2) \dots \mu_{A_{in}}(x_n) \quad (2)$$

とする。このとき、 j 番目の出力の推論結果は

$$u_j^* = \frac{\sum_{i=1}^r h_i B_{ij}}{\sum_{i=1}^r h_i}, \quad j=1, \dots, p \quad (3)$$

となる。ここに、 r は総制御ルール数であり、各入力変数に対して前件部ファジィ数の種類数(つまりラベル数)を等しく取りそれを l とするとき、一般に $r = l^n$ 個となる。なお、上記では前件部ファジィ集合を制御ルールの番号で表現しているが、逆に前件部ファジィ集合のラベル番号を与えて制御ルールの番号を決める場合は、前件部適合度 h_i は

$$h_i = \mu_{A_{j_1 1}}(x_1) \mu_{A_{j_2 2}}(x_2) \dots \mu_{A_{j_n n}}(x_n) \quad (4)$$

のように書ける。この場合、制御ルール番号 i は

$$i = j_n + (j_{n-1} - 1)l + \dots + (j_1 - 1)l^{n-1} \quad (5)$$

のように表現できる。ここに、 $j_n = 1, \dots, l; j_{n-1} = 1, \dots, l; \dots; j_1 = 1, \dots, l$ である。

図 1 は 2 入力 (x_1, x_2)、1 出力 (u_1^*)、前件部ラベル数 3 の場合の、本 FGNN 制御器の構成例を示す。このとき、同定可能な制御ルール数は、 $r = 3^2$ 個である。図中の各層の丸印はユニットを、ユニット間の変数 w_s, w_c, w_d および w_b は結合荷重を表し、記号 { } の付いた変数はネットワークを通過する信号を意味する。また、-1 の記号を持つユニットは -1 の値を出力し、 Σ の記号を持つユニットは入力線の和を出力し、 Π の記号を持つユニットは入力線の積を出力する。D 層における f の記号を持つユニットにおいては、その入力に対する出力は、次のガウシアン関数で定義する。

$$f(x) = e^{\ln 0.5 \cdot x^2} \quad (6)$$

図 1 ファジィ・ガウシアン・ニューラルネットワーク

また、記号 Σ と g を持つ F 層におけるユニットは、その入力 x_i の線形和を関数

$$g(x) = \frac{1}{x} \dots\dots\dots (7)$$

を通し出力する。さらに、記号のないユニットは、単に入力を出力に分配する。

図 1 において A～E 層がファジィ制御ルールの前件部に対応し、G～H 層が後件部に対応する。A 層に加えられた入力 x_i は結合荷重 w_s によってスケールされる。このスケールされた入力に対し、C 層においてバイアスである結合荷重 $-w_c$ が加えられ、それに w_d をかけたものが D 層のガウシアン関数の入力になり、出力 $o^{(D)}$ は

$$o^{(D)} = e^{\ln 0.5 \cdot (w_s x_i - w_c)^2 w_d^2} \dots\dots\dots (8)$$

となる。つまり、結合荷重 w_c 、 w_d はそれぞれガウシアン関数の規格化された台集合上における中心値および偏差値の逆数を決定するものである。これらの w_c 、 w_d の値を事前に適切に設定すれば、この D 層から図 2 のように前件部メンバシップ関数であるガウシアン型メンバシップ関数を構成できる。E 層においては、各制御ルールにおける前件部適合度 h_i が求められ、F 層の第 1 ユニットではそれらの和が取られた後その逆数が計算され、G および H 層において B_{ii} を意味する結合荷重 w_b の h_i による荷重平均値が結論として得られる。

なお、 $[0, 1]$ に値域を持つシグモイド関数を利用する FSNN 法の場合には、 w_c 、 w_d はそれぞれ中心値（つまり 0.5 の関数値に対応する規格化された台集合上の値）および傾きになる。また、その方法では疑似台形型メンバシップ関数を構成するために、 $[0, 1]$ に値域を持つシグモイド関数と $[-1, 0]$ に値域を持つ

図 2 ガウシアン型メンバシップ関数

シグモイド関数を重ね合わせる。したがって、ラベル数が 2 以上では C 層においてそれらの関数の中心値を設定する余分なユニットを必要とし、そして傾きをかけた後の関数値を評価する中間層、さらにはこれらの二つの符号の異なる値を加算する層を通してメンバシップ関数を得ることになる。

2.2 学習方法 ここではバックプロパゲーション法を用いて、FGNN 制御器の結合荷重 w_s 、 w_c 、 w_d 、 w_b を学習させ、制御ルールの同定ならびに前件部メンバシップ関数の微調整を行わせる方法を示す。本稿では、熟練者の良好な制御操作パターンを入手できなくても、あらかじめ設定された目標値と初期設定された結合荷重のもとに得られる実際の制御応答との偏差を最小化するように学習を行える、いわゆる特殊化学習機構を採用する。

いま、第 k 層の入力が関数 $f(\cdot)$ を通って出力され、 $k+1$ 層への入力が加算（つまり Σ ）演算の場合をケース A、同様に第 k 層の入力が関数 $f(\cdot)$ を通って出力され、 $k+1$ 層への入力が積（つまり Π ）演算の場合をケース B とする。すると、出力層 M の j ユニットにおけるデルタ量 δ_j^M と、任意の中間層 k の j ユニットにおけるデルタ量 δ_j^k はそれぞれ以下のように得られる。

$$\text{出力層: } \delta_j^M = f'(i_j^M) \sum_{i=1}^m (y_{di} - y_i) \frac{\partial y_i}{\partial u_j} \dots\dots\dots (9)$$

$$\text{中間層: } \begin{cases} \delta_j^k = f'(i_j^k) \sum_l \delta_l^{k+1} w_{jl}^{k+1} & \text{for case A} \\ \delta_j^k = f'(i_j^k) \sum_l \delta_l^{k+1} w_{jl}^{k+1} \left(\prod_{i \neq j} w_{ii}^{k+1} o_i^k \right) & \text{for case B} \end{cases} \dots\dots\dots (10)$$

ただし、出力層において、 i_j^M は j 番目のユニットへの入力信号を、 m はプラント出力数を、 y_i 、 y_{di} は第 i 番目のプラント出力およびその目標値を、 u_j は第 j 番目のプラントへの入力信号を、 f' はユニットの活性化関数の導関数を表す。また、任意の第 k 番目の中間層においては、 i_j^k は第 j 番目のユニットへの入力信号を、 o_i^k は第 i 番目のユニットからの出力信号を、 w_{jl}^{k+1} は第 k 層の第 j ユニットから第 $k+1$ 層の第 l ユニットへの結合荷重を意味する。このように、ケース B を適用してデルタ量を求めるのは D 層と F 層である。ただし、D 層および F 層第 1 ユニットにおける f' は、式 (6)、(7) の定義より

$$f'(i_j^D) = 2 \ln(0.5) i_j^D o_j^D \text{ for D 層} \dots\dots\dots (11)$$

$$f'(i_j^F) = -(o_j^F)^2 \text{ for F 層第 1 ユニット} \dots\dots\dots (12)$$

となり、それら以外の線形ユニットでは $f'(i_j^k) = 1$ となる。

以上の結果を用いると、結合荷重の更新式は

$$w_{ij}^{k-1,k}(t+1) = w_{ij}^{k-1,k}(t) + \eta \delta_j^k o_i^{k-1} + \alpha \Delta w_{ij}^{k-1,k}(t) \text{ for case A} \quad (13)$$

$$w_{ij}^{k-1,k}(t+1) = w_{ij}^{k-1,k}(t) + \eta \delta_j^k o_i^{k-1} \left(\prod_{i \neq i} w_{ij}^{k-1,k} o_i^{k-1} \right) + \alpha \Delta w_{ij}^{k-1,k}(t) \text{ for case B} \quad (14)$$

となる。ただし、 t は t 回目の更新時刻を、 η は学習率で正の小定数を示しており、 $\Delta w_{ij}^{k-1,k}(t)$ は t ステップにおける結合荷重の増分を示し、 α は安定化定数で正の小定数である。したがって、結合荷重 w_s, w_c, w_d, w_b の更新は式(13)で行える。また、式(14)の更新式を適用する結合荷重は、本稿ではすべて1に固定していることから、学習時には式(14)は使用しない。なお、スケール係数 w_s の学習については堀川らも指摘しているように、例えばA層への入力信号が x_1, x_2 でそれらのオーダーが大きく異なる場合、対応するスケール係数の値が著しくことなり、したがって結合荷重の変化率も大きく異なり、互いに異なる学習率を設定しなくてはならない。これを回避するために式(13)の右辺で δ_j と o_i^{k-1} をそれぞれ $w_{ij}^{k-1,k}(t)$ で規格化した

$$w_{ij}^{k-1,k}(t+1) = w_{ij}^{k-1,k}(t) + \eta \delta_j^k w_{ij}^{k-1,k}(t)^2 o_i^{k-1} + \alpha \Delta w_{ij}^{k-1,k}(t) \quad (15)$$

を用いることにする。

3. 移動ロボット車のモデル

移動ロボットを平面上を移動する剛体であるとし、図3に示すように平面上に固定された絶対座標系を $O-XY$ とし、車の重心を原点とする移動座標系を $O'-X'Y'$ とする。このとき、移動ロボットの動特性は次の運動方程式で表せる⁽¹⁴⁾⁽¹⁵⁾。

$$I_v \ddot{\phi} = D_r l - D_l l \quad (16)$$

$$M \dot{v} = D_r + D_l \quad (17)$$

また、左右輪の駆動系の動特性は

$$I_w \ddot{\theta}_i + c \dot{\theta}_i = k u_i - r D_i; i = r, l \quad (18)$$

図3 2輪独立駆動型移動ロボット車

となる。ここに、各パラメータおよび変数は以下のよう定義する。

I_v : 重心回りの慣性モーメント、 M : 車重

D_l, D_r : 左右輪駆動力、 l : 左右輪と重心間の距離

ϕ : 車両進行方位、 v : 車両速度

I_w : 車輪慣性モーメント、 c : 粘性摩擦係数

k : 駆動ゲイン係数、 r : 車輪回転半径

θ_i : 車輪回転角、 u_i : 駆動入力

一方、変数 ϕ, v, θ_i の間の幾何学的な関係は次式となる。

$$r \dot{\theta}_r = v + l \dot{\phi} \quad (19)$$

$$r \dot{\theta}_l = v - l \dot{\phi} \quad (20)$$

以上により、ロボットの状態変数を $x = [v \ \phi \ \dot{\phi}]^T$ とし、操作量を $u = [u_r \ u_l]^T$ 、出力を $y = [v \ \phi]^T$ とするとき、以下の状態方程式を得る。

$$\dot{x} = Ax + Bu \quad (21)$$

$$y = Cx \quad (22)$$

ここに、

$$A = \begin{bmatrix} a_1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & a_2 \end{bmatrix}, B = \begin{bmatrix} b_1 & b_1 \\ 0 & 0 \\ b_2 & -b_2 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$$a_1 = -2c/(Mr^2 + 2I_w)$$

$$a_2 = -2cl^2/(I_v r^2 + 2I_w l^2)$$

$$b_1 = kr/(Mr^2 + 2I_w)$$

$$b_2 = krl/(I_v r^2 + 2I_w l^2)$$

である。

4. トルク算出制御

3章で定義した各パラメータが正確に測定でき、理想的なモデルが作成できるならば、モデルベース制御器として最適サーボ制御器が設計できる。しかし、最適サーボ器の導出ではその問題をレギュレータ問題に変換し、リッカチ方程式から制御ゲインを決定することから、制御入力表現にプラント出力情報とモデルの利用状況が明確な形で現れてこない。ここでは、ロボットマニピュレータの軌道制御などによく使用されるトルク算出制御(computed torque control)による制御器の設計から得られる知見を、後述のFGNN制御器の結合荷重の初期設定に利用する。

式(21)より u について解くと

$$u_r = \frac{\ddot{\phi}^*}{2b_2} - \frac{a_2 \dot{\phi}}{2b_2} + \frac{\dot{v}^*}{2b_1} - \frac{a_1 v}{2b_1} \quad (23)$$

$$u_l = -\frac{\ddot{\phi}^*}{2b_2} + \frac{a_2 \dot{\phi}}{2b_2} + \frac{\dot{v}^*}{2b_1} - \frac{a_1 v}{2b_1} \quad (24)$$

となる。ただし、上記で $\dot{\phi}^*$ は $\dot{\phi}_d$ に PD 型サーボを追加し、 $\dot{v}^*$ は $\dot{v}_d$ に P 型サーボを追加した

$$\ddot{\phi}^* = \ddot{\phi}_d + K_{\phi v} \dot{e}_\phi + K_{\phi p} e_\phi \dots \dots \dots (25)$$

$$\dot{v}^* = \dot{v}_d + K_{vp} e_v \dots \dots \dots (26)$$

で与えられるとする。ここに、

$$e_\phi = \phi_d - \phi \dots \dots \dots (27)$$

$$e_v = v_d - v \dots \dots \dots (28)$$

とし、 ϕ_d および v_d は方位および速度の目標値である。

ここで注目することは、速度および加速度情報は左右両車輪で同一量を利用するが、方位の微分とさらにその微分情報は左右両車輪で全く相反する量を利用することである。図4には、移動ロボット車の速度・方位制御に対するトルク算出制御器による制御システムのブロック線図を示す。ここで、 $\hat{R}^{-1}$ は導出した移動ロボットの逆動力学モデル式(23)、(24)の推定値である。

5. 独立推論 FGNN+結合ネットによる学習型制御器

3章で述べた移動ロボット車は、速度と方位が左右駆動輪のトルクによって制御されるので、そのプラントは多入出力系である。この場合、上記の FGNN において入力変数を $e_v, e_\phi, \dot{e}_v, \dot{e}_\phi$ 、推論出力を u_r, u_l とするとき、前件部ラベル数 l が増えるに伴って、制御ルール数は l^4 で増加し、ニューラルネットワークのユニット数は爆発的に増加し、学習にも時間がかかることになる。

このような理由から、ここでは速度情報に関する FGNN と方位情報に関する FGNN を独立に設け、これらの推論結果を結合し左右輪への駆動トルクを配分する結合ネットを追加する学習型制御器を導入する。図5には、2個の FGNN と1個の結合ネット(connection net)から成る移動ロボット車の制御系のブロック線図を示す。ただし、学習において結合ネットの第2層のデルタ量計算に、あらためて式(9)を利用し、結合ネットの第1層および FGNN_i の出力層のデル

図4 トルク算出制御システムのブロック線図

タ量計算は、ケース A に対する通常の間接層のものを利用する。

このような方式では、それぞれの FGNN における結合荷重 w_s, w_c, w_d によって前件部メンバシップ関数の微調整を、 w_b によって制御ルールの同定を行い、独立に得られた推論結果の左右輪への配分量を結合ネットの結合荷重 w_{cn} の学習で調整することができる。

6. シミュレーション

提案した FGNN+結合ネット学習型制御器を用いてシミュレーションを行い、その制御方法の有効性を確認する。本シミュレーションでは、3章で述べたロボット車の運動方程式をルンゲクッタ・ギル法で解き、ステップ幅を 1 ms、サンプリング時間幅を 50 ms とし、ロボット車の物理パラメータとしては次のような値を設定した。

$$I_v = 10 \text{ kgm}^2, M = 200 \text{ kg}, l = 0.3 \text{ m}$$

$$I_w = 0.005 \text{ kgm}^2, c = 0.05 \text{ kgm}^2/\text{s}, r = 0.1 \text{ m}, k = 5$$

なお、移動ロボットの目標速度 v_d と目標方位 ϕ_d を 0.25 m/s, 45° の一定値とし、状態変数の初期値を $x = [0 \ 0 \ 0]^T$ と設定した。また、学習制御では学習区間 30 s を学習の 1 試行とし、安定化係数 α は零とした。

6・1 トルク算出制御器によるシミュレーション結果 この制御器では、各々のフィードバック・ゲインは試行錯誤で次のような値を実験的に見出した。

$$K_{\phi v} = 4.0(1/\text{s}), K_{\phi p} = 6.0(1/\text{s}^2),$$

$$K_{vp} = 15.0(1/\text{s})$$

図6には、このときの移動ロボットの速度と方位の制御応答および駆動トルクを示している。トルク算出制御器は逆動力学モデルを用いることから、制御応答の速い理想的な結果が得られていることがわかる。

6・2 独立推論 FGNN+結合ネットを用いたシミュレーション結果

[ケース1] ここでは、独立推論 FGNN+結合ネットを用いた場合のシミュレーションを行った。制御ルールについては、表1のような7ラベルに対する49制御ルールを扱った。したがって、この表内の値をそれぞれの FGNN に対する w_b の初期結合荷重とした。

図5 FGNN 制御器と結合ネット

また、それぞれのラベルは台集合 $[-6, 6]$ に配置するものとして、7 個のそれぞれの中心値 w_c を、 $-6, -4, -2, 0, 2, 4, 6$ 、分散値の逆数 w_d をすべて 1 とした。結合ネットの結合荷重 w_{cn} については、完全に w_{cn} のすべてが零の場合は特殊化学習では学習が不可能である。なぜならば、 $\partial y_i / \partial u_j$ の評価が ∞ となり、このときの計算処理としてこの値を零とするため、それ以後制御入力は零になるからである。また、方位に関係する結合ネットの w_{cn} の二つの初期結合荷重は同符号にしてはならない。もし同符号にした場合には、方位の値が変化しない。これはトルク算出制御の制御入力式 (23), (24) から理解できる。このようなことを考慮して w_s を $1.5, 1.5, 4.8, 4.8$, w_{cn} を $0.1 \times 10^{-2}, 0.1 \times 10^{-2}, 0.1 \times 10^{-3}, -0.1 \times 10^{-3}$ のように初期設定した、なお、 w_c, w_d, w_b については $\eta=0$ として

表 1 ラベル数 7 の制御ルール

e_i	$\dot{e}_i$						
	NB	NM	NS	ZO	PS	PM	PB
NB	-6	-6	-4	-6	-4	-4	-4
NM	-6	-4	-2	-4	-4	-4	-2
NS	-4	-2	-2	-2	0	2	4
ZO	-6	-4	-2	0	2	4	6
PS	-4	-2	0	2	2	2	4
PM	2	4	4	4	2	4	6
PB	4	4	4	6	4	6	6

図 6 トルク算出制御による制御結果

固定し、 w_s については $\eta=0.1 \times 10^{-4}$, w_{cn} については $\eta=0.2 \times 10^{-3}$ として学習した。

図 7 の結果では、2 回学習させた後、移動ロボット車の方位 ϕ が目標方位 45° に達しており、速度 v の制御がほとんどできていない。この原因は、表 2 の w_{cn} の学習結果からもわかるように、設定した w_{cn} の初期値に対して同じ学習率を用いた場合、結果的に v に関する結合荷重が ϕ のものに比べて小さく学習され、したがって、方位の情報を主に利用し速度の情報をあまり利用しない結果となったためと思われる。

[ケース 2] ここでは、表 3 のようにケース 2 の入力スケリングの初期結合荷重 w_s はそのままとし、結合ネットの初期荷重 w_{cn} を $0.5, 0.5, 0.02, -0.02$ と設定し、この学習率を $\eta=0.5 \times 10^{-4}$ に変更した場合の結果を図 8 に示す。このときは速度情報を左右輪がほぼ同一量利用して、速度 v の制御をほぼ達成したが、方位 ϕ の制御がまったくできていないことがわかる。

以上のように、方位と速度を同時に学習させるよう

 表 2 結合荷重 w_s と w_{cn} の学習結果：ケース 1

	Velocity v		Azimuth ϕ	
	initial	1.5	1.5	4.8
w_s	initial	1.5	1.5	4.8
	2nd trial	1.50	1.50	4.81
	initial	0.001	0.001	0.0001
w_{cn}	initial	0.001	0.001	0.0001
	2nd trial	0.0108	0.00558	0.370
				0.0884

図 7 FGNN による制御結果：ケース 1

な結合ネットの初期結合荷重や学習率を見いだすことが容易ではないことから、次に結合ネットは固定して、 w_s と w_b を学習させる場合を考察する。

〔ケース 3〕 ここでは、まず結合ネットの結合荷重 w_{cn} を 1.0, 1.0, 1.0, -1.0 と固定する。その時、結合ネットを通して得られる制御入力は、以下ようになる。

$$u_r = u_v + u_\phi \quad \dots\dots\dots (29)$$

$$u_l = u_v - u_\phi \quad \dots\dots\dots (30)$$

ここで、 u_v は速度情報から決定された車両の速度制御のためのトルク、 u_ϕ は方位情報で決められた車両の方位制御のためのトルクである。

入力スケリングの初期結合荷重 w_s を 2.0, 2.0, 4.8, 4.8, 学習率を $\eta = 0.2 \times 10^{-2}$ のように設定した。 w_b の初期結合荷重についてはケース 1 と同じとし、学習率は、速度用の FGNN と方位用の FGNN でそれぞれ $\eta_v = 0.345 \times 10^{-2}$, $\eta_\phi = 0.78 \times 10^{-4}$ のように設定した。

図 9 には、この結合ネットの固定方式による制御結

果を示しており、また w_s と w_b の学習結果を表 4, 5 に示している。ただし、後件部定数値 w_b の学習については、浮動小数演算 5 けたで四捨五入したとき、表 1 の初期結合荷重から変動したものに対応するルール表のみを書いており、図中の“*”は初期値と同じ値を示している。図 9 からわかるように、4 回の学習で満

表 4 結合荷重 w_s の学習結果：ケース 3

	Velocity v		Azimuth ϕ	
initial	2.0	2.0	4.8	4.8
w_s				
4th trial	2.031	1.997	5.452	4.145

表 5 結合荷重 w_b の学習結果：ケース 3

e_v	$\dot{e}_v$		
	NS	ZO	PS
NS	*	-2.002	-0.8073×10^{-4}
ZO	-1.997	0.3556×10^{-1}	2.002
PS	0.7811×10^{-3}	2.010	2.001

表 3 結合荷重 w_s と w_{cn} の学習結果：ケース 2

	Velocity v		Azimuth ϕ	
initial	1.5	1.5	4.8	4.8
w_s				
2nd trial	1.50	1.54	4.82	4.85
initial	0.5	0.5	0.02	-0.02
w_{cn}				
2nd trial	0.827	0.827	-3.79	25.8

e_ϕ	$\dot{e}_\phi$		
	NS	ZO	PS
NS	*	*	-0.5991×10^{-6}
ZO	*	0.5541×10^{-3}	*
PS	0.1339×10^{-2}	2.001	*

図 8 FGNN による制御結果：ケース 2

図 9 FGNN による制御結果：ケース 3

足のゆく応答が得られ、特に方位については、図 6 の理想的なトルク算出制御結果に近いものが得られた。

7. お わ り に

本論文では、学習型ファジィ・ニューラルネットワーク制御器の一つとして、メンバシップ関数をガウシアン関数で表現し、ネットワークの学習としては特殊化機構を利用する方法を提案した。また、その応用として、2 輪独立駆動型移動ロボット車の速度と方位の軌道追従制御問題を考察した。多入出力制御系に対して、プラント出力誤差とその微分値(あるいは増分値)をファジィ推論への入力とすると、制御ルール数をラベル数とその入力数から決定するならば、それらの数の増加と共にルール数は爆発的に増加する。このことから、ここでは 2 入力 1 出力型の FGNN をプラント出力数分だけ独立に設けておき、それらの推論結果の配分度を学習させる結合ネットを追加する方法を示し、その有効性はシミュレーションにより検証した。

文 献

- (1) 石川, ファジィ制御を用いた自立移動ロボットの誘導方式の検討, 日本ロボット学会誌, 9-2(1991), 21-33.
- (2) 村上・前田, マイクロプロセッサを用いた自動車の最適速度制御, 計測自動制御学会論文集, 19-7(1983), 51-55.
- (3) 岩月・中野・大内, ファジィ推論による移動ロボット車の地点追従制御, 計測自動制御学会論文集, 27-1(1991), 70-76.
- (4) 北村・片井, 画像と制御, (1991), 71-90, 朝倉書店.
- (5) 水上・普家, ニューラルネットワークによるファジィ制御器の学習について, 第 1 回 FAN シンポジウム講演論文集, (1991), 263-266.
- (6) 堀川・古橋・大熊・内川, ニューラルネットワークによる学習型ファジィ制御器, 計測自動制御学会論文集, 27-2(1991), 208-215.
- (7) 中村・堀川・古橋・内川, ファジィニューラルネットワークによる制御戦略知識の獲得, 第 1 回 FAN シンポジウム講演論文集, (1991), 231-234.
- (8) 橋山・古橋・内川, ファジィニューラルネットワークによる顔グラフの表情評価モデル, 第 1 回 FAN シンポジウム講演論文集, (1991), 241-244.
- (9) 水上・普家, ニューラルネットワークによるファジィ制御器の学習について, 第 1 回 FAN シンポジウム講演論文集, (1991), 263-266.
- (10) 菅野, ファジィ制御, (1988), 67-136, 日刊工業新聞社.
- (11) Mamdani, E. H., Application of fuzzy algorithms for control of simple dynamic plant, *Proc. IEE*, 121-12(1974), 1585-1588.
- (12) Mamdani, E. H., Advances in the Linguistic Synthesis of Fuzzy Controller, *Int. J. Man-Machine Studies*, 8-6(1976), 669-679.
- (13) 水本, ファジィ制御向きのファジィ推論法, 計測と制御, 28-11(1989), 959-963.
- (14) 中村, 移動ロボット位置決めに関する基礎研究, 機械技術研究所報告, 第 125 号, (1983), 38-57.
- (15) 齊藤・津村, 移動ロボット群における衝突回避-非線形計画法に基づく局所的解法, システム制御情報学会論文集, 3-8(1990), 252-260.