

距離型ファジィ推論法によるファジィモデリング

木村 太*・王 碩玉**・土谷 武士***

Fuzzy Modeling Using A Distance-type Fuzzy Reasoning Method

Futoshi KIMURA, Shouyu WANG and Takeshi TSUCHIYA

Abstract

In this paper, we propose a fuzzy modeling with the learning algorithm on the basis of a distance-type fuzzy reasoning method. This reasoning method is able to apply in the case of sparse fuzzy rule sets. In this proposal, the premise is a singleton membership function and the consequence is a linear equation that is the Taylor expansion of non-linear equation.

1. はじめに

現実のシステムにおいて、非線形性はあらゆるところに存在し、線形性を保っている状態の方がまれである。近年、非線形システムの制御系設計法として、非線形システムの動特性に基づいたファジィ制御系設計法、いわゆるモデルに基づくファジィ制御¹⁾²⁾が提案されている。この設計法は、制御対象をファジィモデルで表現し、さらにモデルの1つのルールに対して、1つのファジィ制御器を設計するもので、きわめて自然な方法である。しかし、非線形システムをファジィモデルで表すためには、動作点近傍で局所的に線形化、あるいはセクタとなる線形システムを見つけ出す作業が必要である。³⁾

筆者らが提案している距離型ファジィ推論法による学習アルゴリズム⁴⁾は、有限個の教師信号に対して、新しいルールの追加、古いルールの更新、間違っただルールの訂正により、任意に指定した誤差範囲内に近似することが可能な手法である。従って、強い非線形性を持つ教師信号に対しても、動作点を探索することなく近似できる。

本稿では、距離型ファジィ推論法に基づく学習アルゴリズム、および関数型推論法⁵⁾を用いた非線形関数のファジィモデリング方法を提案し、さらに自由システムへの適用方法についても述べる。

2. 距離型ファジィ推論法

本章では、距離型ファジィ推論法に基づく関数

型推論法と学習アルゴリズムについて述べる。

2.1 関数型推論法

(1)式に示すように、後件部を関数とするif-thenルールにおいて、推論結果は(2),(3)式から求められる。

$$R^i: \text{If } x_1 \text{ is } A^{i1} \text{ and } x_2 \text{ is } A^{i2} \dots \text{and } x_m \text{ is } A^{im} \\ \text{then } y = f_i(\mathbf{x}) \quad (1)$$

ここで、 $i=1,2,\dots,n$, n はルール数、 $j=1,2,\dots,m$, m は前件部変数の数である。また、 $\mathbf{x}=[x_1, x_2, \dots, x_m]$ A^{ij} 、 $f_i(\mathbf{x})$ 、 $\mathbf{x}_0=[x_{01}, x_{02}, \dots, x_{0m}]$ 、 y_0 はそれぞれ前件部変数、前件部ファジィ集合、後件部、入力、推論結果を表し、

$$\mathbf{x}, \mathbf{x}_0, f_i(\mathbf{x}), y_0 \in R$$

とする。

なお、前件部ファジィ集合は有界凸なファジィ集合とする。

$$y_o = \frac{\sum_{i=1}^n \left\{ f_i(\mathbf{x}_0) \prod_{j=1, j \neq i}^n d_j \right\}}{\sum_{i=1}^n \prod_{j=1, j \neq i}^n d_j} \quad (2)$$

$$d_i = \sum_{j=1}^m d(A^{ij}, x_{0j}) \quad (3)$$

ただし、 $d(\cdot, \cdot)$ はファジィ集合間の距離を定義する距離関数である。

2.2 学習アルゴリズム

距離型ファジィ推論法の分離特性に基づく学習アルゴリズムは、4つのステップから構成される。なお、教師信号の個数は L 個とする。

Step1: 与えられた教師信号 $\{A_i^j(k), B_i(k)\}$ に対して前件部の入力 $A_i^j(k)$ から推論結果 $B(k)$ を求める。

* 助 手 電気電子工学科

** 教 授 高知工科大学

*** 教 授 北海道大学

ただし、 $j=1,2,\dots,m$ $k=1,2,\dots,L$

Step2: 教師信号と推論結果との距離を計算し、もし $d(B_i(k), B(k)) \leq \varepsilon$ であれば、次の教師信号に対して、Step1に戻る。ここで、 ε は任意に指定した誤差である。

Step3: もし、 $d(B_i(k), B(k)) > \varepsilon$ であれば次のルールを新しいルールとして追加し、Step1に戻る。
 R^p : If x_1 is $A_i^1(k)$ and x_2 is $A_i^2(k)$...and x_m is $A_i^m(k)$
 then $y = B_i(k)$ (4)

Step4: もし、 $d(B_i(k), B(k)) > \varepsilon$ であり、なおかつ $\sum_{j=1}^m d(A_i^j, A_i^j(k)) = 0$ (ただし、 $\exists q \in \{1, 2, \dots, n\}$) であれば、 q 番目のルールの後件部を $B_i(k)$ に更新してからStep1に戻る。

3. ファジィモデリング

ここでは、学習アルゴリズムに基づくファジィモデリングの方法について述べ、さらに1変数および2変数の非線形関数に対するモデリングについて説明する。

3.1 学習アルゴリズムに基づくモデリング方法

本提案におけるモデリング方法は次の特徴を持つ。

(1) 学習アルゴリズムに用いる教師信号は、非線形関数上の点となるため、前件部ファジィ集合はシングルトンメンバーシップ関数となる。

(2) ルールの後件部は、非線形関数の1次項までのテイラー展開で得られる線形関数とする。

いま、次式に示す連続な非線形関数に対するモデリングは、次のステップから構成される。

$$y = f(\mathbf{x}) \quad (5)$$

ここで、 $\mathbf{x} = [x_1, x_2, \dots, x_m]$ 、 m は変数の個数であり、 $y, \mathbf{x} \in R$ である。

Step1: 非線形関数上の点 $(\mathbf{x}_1, y_1)$ と $(\mathbf{x}_2, y_2)$ を教師信号とし、この2点からそれぞれルール、 R^1, R^2 を作成する。

ただし、 $\mathbf{x}_1 = [x_{11}, x_{12}, \dots, x_{1m}]$ 、 $\mathbf{x}_2 = [x_{21}, x_{22}, \dots, x_{2m}]$

$$R^1: \text{If } x_1 \text{ is } x_{11} \text{ and } x_2 \text{ is } x_{12} \dots \text{and } x_m \text{ is } x_{1m} \\ \text{then } y_{(1)} = f(\mathbf{x}_1) + \sum_{i=1}^m \frac{\partial f(\mathbf{x}_1)}{\partial x_i} (x_i - x_{1i}) \quad (6)$$

R^2 : If x_1 is x_{21} and x_2 is x_{22} ...and x_m is x_{2m}

$$\text{then } y_{(2)} = f(\mathbf{x}_2) + \sum_{i=1}^m \frac{\partial f(\mathbf{x}_2)}{\partial x_i} (x_i - x_{2i}) \quad (7)$$

Step2: 与えられた教師信号 $(\mathbf{x}_t, y_t)$ に対して、入力 $\mathbf{x}_t$ から推論結果 y を求める。

$$y = \frac{\sum_{i=1}^n \left\{ y_{(i)} \prod_{j=1, j \neq i}^n d_j \right\}}{\sum_{i=1}^n \prod_{j=1, j \neq i}^n d_j} \quad (8)$$

$$d_i = \sum_{j=1}^m d(x_j, x_{ji}) \quad (9)$$

ただし、 $i=1, 2, \dots, n$, n はルール数、 $\mathbf{x}_t = [x_{t1}, x_{t2}, \dots, x_{tm}]$ である。

Step3: 教師信号と推論結果との距離を計算し、もし $d(y, y_t) \leq \varepsilon$ であれば、次の教師信号に対して、Step2に戻る。ここで、 ε は任意に指定した誤差である。

Step4: もし、 $d(y, y_t) > \varepsilon$ であれば、次のルールを新しいルールとして追加し、Step2に戻る。

R^k : If x_1 is x_{k1} and x_2 is x_{k2} ...and x_m is x_{km}

$$\text{then } y_{(k)} = f(\mathbf{x}_k) + \sum_{i=1}^m \frac{\partial f(\mathbf{x}_k)}{\partial x_i} (x_i - x_{ki}) \quad (10)$$

ただし、 $x_{ki} = x_{ti}$, $i=1, 2, \dots, m$

Step5: もし、 $d(y, y_t) > \varepsilon$ であり、なおかつ

$$\sum_{j=1}^m d(x_q, x_{ji}) = 0 \quad (\text{ただし、} \exists q \in \{1, 2, \dots, n\})$$

であれば、 q 番目のルールの後件部を(10)式に更新してからStep1に戻る。

$$y_{(q)} = f(\mathbf{x}_t) + \sum_{i=1}^m \frac{\partial f(\mathbf{x}_t)}{\partial x_i} (x_i - x_{qi}) \quad (11)$$

ただし、 $x_{qi} = x_{ti}$, $i=1, 2, \dots, m$

3.2 1変数のモデリング

図1に示すような1変数の非線形関数に対するモデリングは、まず関数上の2点 (x_1, y_1) 、 (x_2, y_2) から2ルールを次式のように作成する。

$$R^1: \text{If } x \text{ is } x_1 \text{ then } y_{(1)} = y_1 + \frac{df(x_1)}{dx} (x - x_1) \quad (12)$$

$$R^2: \text{If } x \text{ is } x_2 \text{ then } y_{(2)} = y_2 + \frac{df(x_2)}{dx}(x - x_2) \quad (13)$$

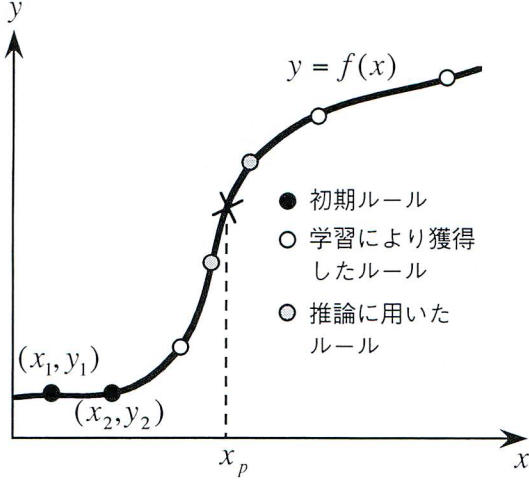


図1 1変数のモデリング

このルールからx方向に対してモデリングを行う。なお、学習により獲得されたルールは次式となる。

$$R^k: \text{If } x \text{ is } x_t \text{ then } y_{(k)} = y_t + \frac{df(x_t)}{dx}(x - x_t) \quad (14)$$

ただし、 $y_t = f(x_t)$

得られたファジィモデルから任意座標の推論値を求めるには、図1のようにその座標と直近のルールを採用して推論を行う。

3.3 2変数のモデリング

次式で示される2変数の非線形関数のモデリングは、以下の手順で行う。

$$y = f(x_1, x_2) \quad (15)$$

(1) x_1 方向のモデリング (図2の赤色部分)

Step1: 図2のように点 (x_{11}, x_2^k) 、 (x_{12}, x_2^k) から初期ルールを作成する。ただし、 $k=0, 1, 2, \dots$

$$R^1: \text{If } x_1 \text{ is } x_{11} \text{ and } x_2 \text{ is } x_2^k \text{ then } y_{(1)} = f(x_{11}, x_2^k) + \frac{\partial f(x_{11}, x_2^k)}{\partial x_1}(x_1 - x_{11}) + \frac{\partial f(x_{11}, x_2^k)}{\partial x_2}(x_2 - x_2^k) \quad (16)$$

$$R^2: \text{If } x_1 \text{ is } x_{12} \text{ and } x_2 \text{ is } x_2^k \text{ then } y_{(2)} = f(x_{12}, x_2^k) + \frac{\partial f(x_{12}, x_2^k)}{\partial x_1}(x_1 - x_{12}) + \frac{\partial f(x_{12}, x_2^k)}{\partial x_2}(x_2 - x_2^k) \quad (17)$$

Step2: x_2 座標を固定し、 x_1 方向について、指定された範囲まで学習を行う。学習により獲得され

たルールは次式となる。

$$R^q: \text{If } x_1 \text{ is } x_{1q} \text{ and } x_2 \text{ is } x_2^k \text{ then } y_{(q)} = f(x_{1q}, x_2^k) + \frac{\partial f(x_{1q}, x_2^k)}{\partial x_1}(x_1 - x_{1q}) + \frac{\partial f(x_{1q}, x_2^k)}{\partial x_2}(x_2 - x_2^k) \quad (18)$$

Step3: x_2 座標を一定幅 h_2 移動し、Step1,2を繰り返す。

(2) x_2 方向のモデリング (図2の青色部分)

x_1 座標を固定し、 x_2 方向に対してStep1~3を同様に行う。なお、初期ルールおよび獲得されたルールはそれぞれ、(19),(20),(21)式で与えられる。

$$R^1: \text{If } x_1 \text{ is } x_1^k \text{ and } x_2 \text{ is } x_{21} \text{ then } y_{(1)} = f(x_1^k, x_{21}) + \frac{\partial f(x_1^k, x_{21})}{\partial x_1}(x_1 - x_1^k) + \frac{\partial f(x_1^k, x_{21})}{\partial x_2}(x_2 - x_{21}) \quad (19)$$

$$R^2: \text{If } x_1 \text{ is } x_1^k \text{ and } x_2 \text{ is } x_{22} \text{ then } y_{(2)} = f(x_1^k, x_{22}) + \frac{\partial f(x_1^k, x_{22})}{\partial x_1}(x_1 - x_1^k) + \frac{\partial f(x_1^k, x_{22})}{\partial x_2}(x_2 - x_{22}) \quad (20)$$

$$R^q: \text{If } x_1 \text{ is } x_1^k \text{ and } x_2 \text{ is } x_{2q} \text{ then } y_{(q)} = f(x_1^k, x_{2q}) + \frac{\partial f(x_1^k, x_{2q})}{\partial x_1}(x_1 - x_1^k) + \frac{\partial f(x_1^k, x_{2q})}{\partial x_2}(x_2 - x_{2q}) \quad (21)$$

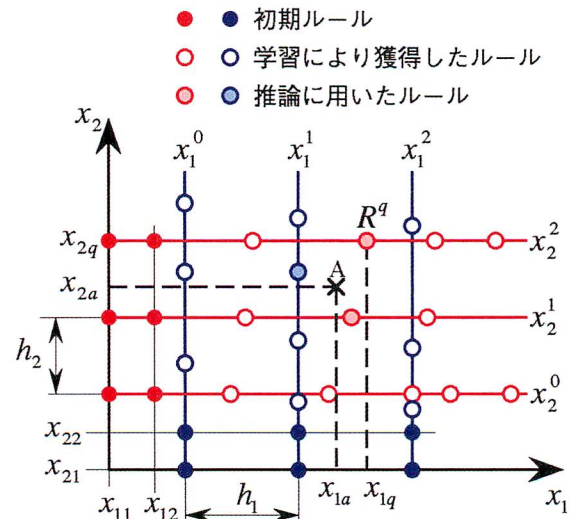


図2 2変数のモデリング

(3) 任意座標の推論

任意座標の推論値は、得られたファジィモデルから次の手順で求められる。

例として、図2において座標 $A(x_{1a}, x_{2a})$ の推論値を求めるものとする。

Step1: 任意座標と学習されたルール座標との距離 d を計算する。例えばルール R^q の場合、

$$d = \sqrt{(x_{1a} - x_{1q})^2 + (x_{2a} - x_{2q})^2} \quad (22)$$

Step2: 距離 d の小さいルールから順に、指定した個数のルールを採用する。ただし、 $d=0$ となるルールは1つのみとする。

Step3: 採用されたルールから推論を行う。

4. 自由システムへの適用

本章では、提案したモデリング方法を自由システムへ適用し、非線形な状態方程式のファジィモデル化について述べる。

いま、次式で示される2次の自由システムが与えられたとする。

$$\text{状態変数: } \mathbf{x} = [x_1, x_2]^T \quad (23)$$

$$\text{状態方程式: } \begin{cases} \dot{x}_1 = f_1(x_1, x_2) \\ \dot{x}_2 = f_2(x_1, x_2) \end{cases} \quad (24)$$

ただし、 f_1, f_2 は連続な非線形関数とする。

次に、 $\dot{x}_1, \dot{x}_2$ について以下の範囲でモデリングを行い、学習アルゴリズムによりファジィルールを獲得する。

$$\begin{cases} x_{1a} \leq x_1 \leq x_{1b} \\ x_{2a} \leq x_2 \leq x_{2b} \end{cases} \quad (25)$$

ここで、 $x_{1a}, x_{1b}, x_{2a}, x_{2b} \in R$ とする。

モデリングによって得られたif-thenルールは(26), (27)式で表される。

[$\dot{x}_1$ のルール]

$$\begin{aligned} R^i: & \text{ If } x_1 \text{ is } x_{1i}^1 \text{ and } x_2 \text{ is } x_{2i}^1 \\ & \text{ then } \dot{x}_{1(i)} = f_1(\mathbf{x}_i^1) + \frac{\partial f_1(\mathbf{x}_i^1)}{\partial x_1} (x_1 - x_{1i}^1) \\ & \quad + \frac{\partial f_1(\mathbf{x}_i^1)}{\partial x_2} (x_2 - x_{2i}^1) \end{aligned} \quad (26)$$

ただし、 $\mathbf{x}_i^1 = [x_{1i}^1, x_{2i}^1]$

[$\dot{x}_2$ のルール]

$$\begin{aligned} R^i: & \text{ If } x_1 \text{ is } x_{1i}^2 \text{ and } x_2 \text{ is } x_{2i}^2 \\ & \text{ then } \dot{x}_{2(i)} = f_2(\mathbf{x}_i^2) + \frac{\partial f_2(\mathbf{x}_i^2)}{\partial x_1} (x_1 - x_{1i}^2) \\ & \quad + \frac{\partial f_2(\mathbf{x}_i^2)}{\partial x_2} (x_2 - x_{2i}^2) \end{aligned} \quad (27)$$

ただし、 $\mathbf{x}_i^2 = [x_{1i}^2, x_{2i}^2]$

ここで x_{1i}^1, x_{2i}^1 は、それぞれ i 番目ルールにおける変数 x_1 および x_2 のシングルトンメンバーシップ関数を表す。また、肩文字の“1”は $\dot{x}_1$ についてのモデリングを意味する。 x_{1i}^2, x_{2i}^2 についても同様である。

次に、(26), (27)式のルールについて推論結果を求めると、(28)~(31)式となる。ただし、 $n1$ と $n2$ は $\dot{x}_1$ および $\dot{x}_2$ でのモデリングによって得られたルール数である。

$$\dot{x}_1 = \frac{\sum_{i=1}^{n1} \left\{ \dot{x}_{1(i)} \prod_{j=1, j \neq i}^{n1} d_j^1 \right\}}{\sum_{i=1}^{n1} \prod_{j=1, j \neq i}^{n1} d_j^1} \quad (28)$$

$$d_i^1 = \sum_{j=1}^2 d(x_j, x_{ji}^1) \quad (29)$$

$$\dot{x}_2 = \frac{\sum_{i=1}^{n2} \left\{ \dot{x}_{2(i)} \prod_{j=1, j \neq i}^{n2} d_j^2 \right\}}{\sum_{i=1}^{n2} \prod_{j=1, j \neq i}^{n2} d_j^2} \quad (30)$$

$$d_i^2 = \sum_{j=1}^2 d(x_j, x_{ji}^2) \quad (31)$$

さらに、(28), (30)式にルール後件部の線形式を代入することにより次式を得る。

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{D}$$

$$\mathbf{A} = \begin{bmatrix} \frac{\sum_{i=1}^{n1} \left\{ \frac{\partial f_1(\mathbf{x}_i^1)}{\partial x_1} \right\} \prod_{j=1, j \neq i}^{n1} d_j^1}{\Delta^1} & \frac{\sum_{i=1}^{n1} \left\{ \frac{\partial f_1(\mathbf{x}_i^1)}{\partial x_2} \right\} \prod_{j=1, j \neq i}^{n1} d_j^1}{\Delta^1} \\ \frac{\sum_{i=1}^{n2} \left\{ \frac{\partial f_2(\mathbf{x}_i^2)}{\partial x_1} \right\} \prod_{j=1, j \neq i}^{n2} d_j^2}{\Delta^2} & \frac{\sum_{i=1}^{n2} \left\{ \frac{\partial f_2(\mathbf{x}_i^2)}{\partial x_2} \right\} \prod_{j=1, j \neq i}^{n2} d_j^2}{\Delta^2} \end{bmatrix} \quad (32)$$

$$\mathbf{D} = \begin{bmatrix} \frac{\sum_{i=1}^{n1} \left\{ f_1(\mathbf{x}_i^1) - \frac{\partial f_1(\mathbf{x}_i^1)}{\partial x_1} x_{1i}^1 - \frac{\partial f_1(\mathbf{x}_i^1)}{\partial x_2} x_{2i}^1 \right\} \prod_{j=1, j \neq i}^{n1} d_j^1}{\Delta^1} \\ \frac{\sum_{i=1}^{n2} \left\{ f_2(\mathbf{x}_i^2) - \frac{\partial f_2(\mathbf{x}_i^2)}{\partial x_1} x_{1i}^2 - \frac{\partial f_2(\mathbf{x}_i^2)}{\partial x_2} x_{2i}^2 \right\} \prod_{j=1, j \neq i}^{n2} d_j^2}{\Delta^2} \end{bmatrix} \quad (34)$$

$$\text{ただし、} \Delta^1 = \sum_{i=1}^{n1} \prod_{j=1, j \neq i}^{n1} d_j^1 \quad \Delta^2 = \sum_{i=1}^{n2} \prod_{j=1, j \neq i}^{n2} d_j^2$$

上記の行列 $\mathbf{A}$ 、 $\mathbf{D}$ の各要素は実数値となるため、本手法によって非線形システムをファジィモデルで表現することが可能である。

5. 例 題

提案した手法をVan der Polの方程式に適用した結果を以下に示す。

ファジィモデル

(35)式で与えられるVan der Pol方程式について $\mu=1$ とし、状態変数を $x_1 = x, x_2 = \dot{x}$ とすると、

$$\ddot{x} + \mu(x^2 - 1)\dot{x} + x = 0 \quad (35)$$

(36)式を得る。

$$\dot{x}_2 = -x_1 + x_2 - x_1^2 x_2 \quad (36)$$

上式の右辺第3項は非線形となっているため、(35)式は線形の状態方程式で表すことができない。そこで、非線形項を次の関数としてモデリングを行う。

$$y = -x_1^2 x_2 \quad (37)$$

モデリングで得られたif-thenルールは次のようになり、さらに推論結果を求めると(39)式となる。

$$\begin{aligned} R' : & \text{If } x_1 \text{ is } x_{i1} \text{ and } x_2 \text{ is } x_{i2} \\ & \text{then } y_{(i)} = -x_{i1}^2 x_{i2} - 2x_{i1} x_{i2} (x_1 - x_{i1}) - x_{i1}^2 (x_2 - x_{i2}) \end{aligned} \quad (38)$$

$$y = \frac{1}{\Delta} \sum_{i=1}^n \left\{ -x_{i1}^2 x_{i2} - 2x_{i1} x_{i2} (x_1 - x_{i1}) - x_{i1}^2 (x_2 - x_{i2}) \right\} \prod_{j=1, j \neq i}^n d_j \quad (39)$$

ただし、 $d_i = |x_1 - x_{i1}| + |x_2 - x_{i2}|$

この推論結果を(36)式に代入することにより、Van der Pol方程式を線形の状態方程式で表すことができる。

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ C_1 & C_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ C_3 \end{bmatrix} \quad (40)$$

ただし、

$$C_1 = -1 + \frac{1}{\Delta} \sum_{i=1}^n (-2x_{i1} x_{i2}) \prod_{j=1, j \neq i}^n d_j$$

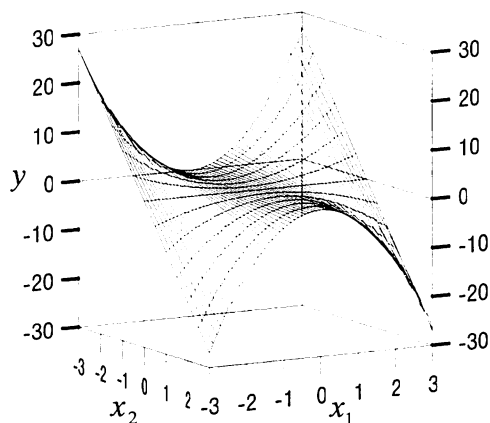
$$C_2 = 1 + \frac{1}{\Delta} \sum_{i=1}^n (-x_{i1}^2) \prod_{j=1, j \neq i}^n d_j$$

$$C_3 = \frac{1}{\Delta} \sum_{i=1}^n (2x_{i1}^2 x_{i2}) \prod_{j=1, j \neq i}^n d_j$$

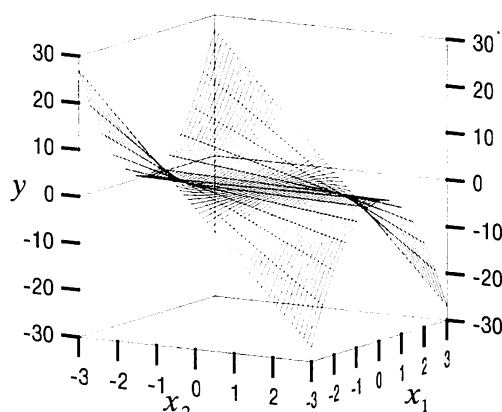
$$\Delta = \sum_{i=1}^n \prod_{j=1, j \neq i}^n d_j$$

シミュレーション結果

図3は非線形項を x_1 および x_2 方向についてモデリングを行った結果である。モデリングの条件は、教師信号との誤差を5%とし、学習のきざみ幅は両方向ともに0.1としている。また、図2における移動幅は h_1, h_2 とともに0.5とした。



(a) x_1 方向



(b) x_2 方向

図3 モデリング結果
($y = -x_1^2 x_2$)

(38)式のファジィモデルから非線形項を推論した結果を図4に示す。この結果から、後件部を線形関数としたif-thenルールによって、非線形関数を近似できることが確認できた。

次に、Van der Pol方程式のシミュレーション結果を図5に示す。図5において(a)は $x_1 - x_2$ を図示したものであり、(b)は x_1 について理論値とファジィモデルからの結果とを比較したものである。なお、状態変数の初期値は次の値としている。
 $[x_1, x_2] = [0.25, 0]$

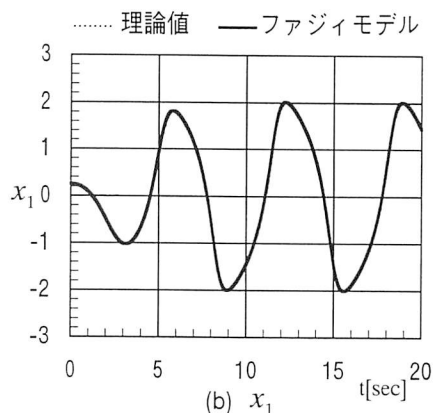
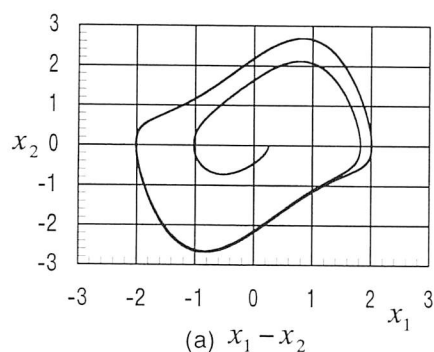
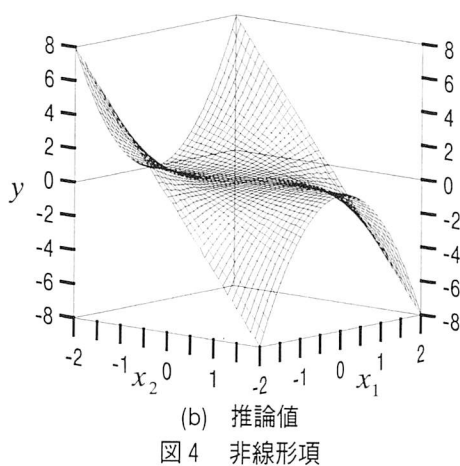
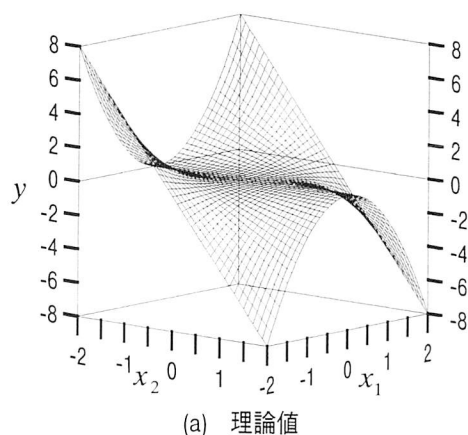


図5 シミュレーション結果
(Van der Pol方程式)

図4(a)を見ると、 $x_1 - x_2$ の挙動はVan der Pol方程式特有のリミットサイクルとなっており、また理論値ともよく一致していることがわかる。

6. おわりに

本稿では、距離型ファジィ推論法に基づいた学習アルゴリズムを用いて、非線形関数のファジィモデリング手法および自由システムへの適用方法について述べた。その結果、非線形関数の動作点やセクタを探索せずに、モデリングが可能であることが示された。また、非線形な自由システムを、本手法を適用したファジィモデルで表現できることが確認された。

今後は、強制系への拡張について検討し、距離型推論法を用いたモデルに基づくファジィ制御への適用を目指す。

参考文献

- 1) 田中：アドバンストファジィ制御, 共立出版 (1994)
- 2) 木村, 王, 土谷：距離型ファジィ推論法を用いたモデルに基づくファジィ制御, 第16回ファジィシステムシンポジウム講演論文集, 5/6 (2000)
- 3) 田中, 大竹：Sector nonlinearityの概念を用いたファジィモデルの同定, 計測自動制御学会論文集, Vol.37, No.4, 372/378 (2001)
- 4) 王, 土谷：距離型ファジィ推論法Part7, 第13回ファジィシステムシンポジウム講演論文集, 145/148 (1997)
- 5) 王, 土谷, 水本：距離型ファジィ推論法, バイオメディカルファジィシステム学会誌, Vol.1, No.1, 61/78 (1999)

(平成13年11月27日受理)