2024 IEEE 63rd Conference on Decision and Control (CDC)
December 16-19, 2024. MiCo, Milan, Italy

# Piecewise System Modeling Algorithm Based on Simplified Fuzzy Inference Reasoning and Particle Swarm Optimization

Tadanari Taniguchi

*Abstract*— **This paper proposes a piecewise modeling method based on fuzzy if-then rules using particle swarm optimization. The piecewise model has the shape of a rectangular partition of the state-space; the model can be represented as a fuzzy if-then rule with singleton consequents. The vertex values of the rectangular regions are determined using particle swarm optimization because the optimal solution is a nonlinear programming problem. In order to determine optimal vertex values of the piecewise regions with minimal modeling errors, this paper proposes a learning algorithm based on simplified fuzzy inference reasoning and particle swarm optimization methods. Some numerical simulation results show the effectiveness of the learning algorithm.**

## I. INTRODUCTION

In recent years, nonlinear systems have been modeled using neural networks as data-driven modeling in [1], [2]. Although the modeling performance is high, stability analysis of the model can be difficult due to the high nonlinearity of the model. Piecewise model studies [3], [4] have also been conducted. Most studies dealt with piecewise linear models which are easier for stability analysis and control system design. Fuzzy models [5] were also used for control system design, but most studies were conducted for target systems with known physical structures. In [6], an improved particle swarm optimization (PSO) method was used for training the parameters of adaptive neural network based fuzzy inference systems. A PSO algorithm was utilized to optimize the fuzzy set parameters in [7].

This study uses a piecewise multilinear (PML) system [8] as a piecewise model. The system has a simple nonlinearity and a universal approximator property. In addition, it is fully parametric. The model has a rectangular shape of the state-space. It is formed by convex combinations of the vertices of the piecewise regions and the adjacent regions have continuity. Because we can represent the models as a fuzzy if-then rule with singleton consequents, the PML system is also regarded as a fuzzy system [5]. The stabilizing conditions of the PML system for continuous and discrete-time nonlinear systems have been derived in [9], [10].

The author has been conducting research on modeling methods using piecewise models. Three variables, vertex values of the piecewise models, dividing points in the state-space, and the number of piecewise regions, need to be derived for the PML model construction. In [11], the dividing points were determined by the data smoothing method of the spline function, and the vertex values of the piecewise regions were determined by simplified fuzzy reasoning. However, the solution region of the dividing points is narrow with spline function data smoothing. Therefore, they may fall into a local solution. In [12], the dividing points were determined by PSO algorithm [13], [14] to widen the solution region of these points, and the optimization of the vertex values for piecewise models was not performed. This study realizes the optimization of the dividing points by PSO in [12] and the vertex values of piecewise regions by simplified fuzzy reasoning in [11], and achieves the construction of a piecewise model with a small modeling error.

This study develops a piecewise modeling method based on the PSO algorithm using numerical data. It is an optimization method that searches for optimal values by simulating the behavior of living animals such as school of fish or flock of birds. The PSO algorithm searches for optimal solutions by having multiple particles fly around the search space in where the solution needs to be searched. In [12], the dividing points of the piecewise regions were determined using the PSO algorithm because finding the optimal solution is a nonlinear programming problem. Because the piecewise model is regarded as a fuzzy if-then rule with singleton consequents, this study applies a learning algorithm based on a simplified fuzzy inference reasoning [15] to find the vertex values of the rectangular region. The optimal vertex values and dividing points of the piecewise regions with minimal modeling errors can be found by the proposed modeling algorithm. Finally, numerical simulations are performed to demonstrate the effectiveness of the proposed method.

*Contributions:* This study achieved optimal dividing points in the state-space and vertex values of the piecewise model that minimizes modeling error. The piecewise model used in this study is a nonlinear model to which stability analysis and stabilization methods can be applied. The model can be constructed using only vertex value information. While it is easy to model, it is difficult to determine the optimal domain division points. This study proposes a method for constructing a classification model with low modeling error.

## II. PML MODEL

### A. *If-then Rule Expression*

The PML model in $R_{r_1\dots r_n}$ can be transformed into an if-then rule expression based on a fuzzy reasoning.

T. Taniguchi is with STEM Education Center, Tokai University, Kanagawa 2591292, Japan. taniguchi@tokai.ac.jp

$$\begin{aligned}
\text{Rule } 1: &\text{ If } x_1 \text{ is } A_1^{r_1} \text{ and } \cdots \text{ and } x_n \text{ is } A_n^{r_n}, \\
&\text{ Then } x^+ \text{ is } f(r_1,\ldots,r_n). \\
&\vdots \\
\text{Rule } 2^n: &\text{ If } x_1 \text{ is } A_1^{r_1+1} \text{ and } \cdots \text{ and } x_n \text{ is } A_n^{r_n+1}, \\
&\text{ Then } x^+ \text{ is } f(r_1+1,\ldots,r_n+1).
\end{aligned} \tag{1}$$

where $A_j^{r_j}$ is the fuzzy set, and $\omega_j^{r_j}(x_j)$, $j=1,\ldots,n$, is the membership function. The degree of the rules is denoted by

$$\begin{aligned}
\mu_k(x) =& \Pi_{j=1}^n \omega_j^{i_j}(x_j), \\
\omega_j^{r_j}(x_j) =& \Pi_{j=1}^n A_j^{r_j},
\end{aligned}$$

where $i_1 = r_1, r_1+1, \ \ldots, \ i_n = r_n, r_n+1$, and $k = \sum_{j=1}^n 2^{j-1}(i_j - r_j + 1)$.

Region $R_(r_1 \cdot r_n)$ has $2^n$ if-then rules. The PML model has $\Pi_{i=1}^n(m_i-1)$ regions and the entire system has $2^n\Pi_{i=1}^n(m_i-1)$ rules. The fuzzy inference system $x^+$ is denoted by

$$\begin{aligned}
x^+ =& \sum_{k=1}^{2^n} \mu_k(x) f^k \\
=& \sum_{i_1=r_1}^{r_1+1} \omega_1^{i_1}(x_1) \cdots \sum_{i_n=r_n}^{r_n+1} \omega_n^{i_n}(x_n) f(i_1,\ldots,i_n),
\end{aligned} \tag{2}$$

where $k = \sum_{i=1}^n 2^{i-1}(i_j - r_j + 1)$ and $f^k = f(i_1,\cdots,i_n)$. The fuzzy inference system in (2) is the same as that of the PML system (4). If the PML system is a continuous-time system, then $x^+$ means $x^+ = dx/dt$ ; if it is a discrete-time system, then $x^+$ means $x(t+1)$; if it is a static system, then $x^+$ means $z$.

Note that the PML system (2) has a fuzzy if-then rule expression with singleton consequents. On the other hand, the consequent terms of fuzzy if-then rules in T-S fuzzy systems [5] are the state-space representation of linear systems.

### B. State-Space Expression

This study deals with the PML system in $n$-dimensional case. The state-space is divided by the following points.

$$\begin{aligned}
x_1 \in& \{d_1(1),\ldots,d_1(r_1),\ldots,d_1(m_1)\}, \\
&\vdots \\
x_n \in& \{d_n(1),\ldots,d_n(r_n),\ldots,d_n(m_n)\},
\end{aligned} \tag{3}$$

where $d_j(i_j)$ is the vertex of $x_j$, and $n_j$ is the number of vertices of $x_j$; $j=1,\ldots,n$. $x \in S$, where $S$ is the bounded region. The number of piecewise regions is $\Pi_{i=1}^n(m_i-1)$. The PML model in region

$$\begin{aligned}
R_{r_1\ldots r_n} =& \{(x_1,\ldots,x_n) | d_1(r_1) \le x_1 \le d_1(r_1+1), \\
& \ldots, d_n(r_n) \le x_n \le d_n(r_n+1)\}
\end{aligned}$$

is denoted by

$$\begin{cases}
x^+ = \sum_{i_1=r_1}^{r_1+1} \omega_1^{i_1}(x_1) \cdots \sum_{i_n=r_n}^{r_n+1} \omega_n^{i_n}(x_n) f(i_1,\ldots,i_n), \\
x = \sum_{i_1=r_1}^{r_1+1} \omega_1^{i_1}(x_1) \cdots \sum_{i_n=r_n}^{r_n+1} \omega_n^{i_n}(x_n) d(i_1,\ldots,i_n),
\end{cases} \tag{4}$$

where

$$\begin{aligned}
f(r_1,\ldots,r_n) =& \big(f_1(r_1,\ldots,r_n),\ldots,f_n(r_1,\ldots,r_n)\big)^T, \\
d(r_1,\ldots,r_n) =& \big(d_1(r_1,\ldots,r_n),\ldots,d_n(r_1,\ldots,r_n)\big)^T.
\end{aligned}$$

The triangular membership functions are as follows:

$$\begin{aligned}
&\omega_1^{r_1}(x_1) \\
&= \begin{cases} \dfrac{d_1(r_1+1)-x_1}{d_1(r_1+1)-d_1(r_1)}, & d_1(r_1) \le x_1 \le d_1(r_1+1) \\ 0, & \text{otherwise} \end{cases} \\
&\omega_1^{r_1+1}(x_1) \\
&= \begin{cases} \dfrac{x_1-d_1(r_1)}{d_1(r_1+1)-d_1(r_1)}, & d_1(r_1) \le x_1 \le d_1(r_1+1) \\ 0, & \text{otherwise} \end{cases} \\
&\vdots \\
&\omega_n^{r_n}(x_n) \\
&= \begin{cases} \dfrac{d_n(r_n+1)-x_n}{d_n(r_n+1)-d_n(r_n)}, & d_n(r_n) \le x_n \le d_n(r_n+1) \\ 0, & \text{otherwise} \end{cases} \\
&\omega_n^{r_n+1}(x_n) \\
&= \begin{cases} \dfrac{x_n-d_n(r_n)}{d_n(r_n+1)-d_n(r_n)}, & d_n(r_n) \le x_n \le d_n(r_n+1) \\ 0, & \text{otherwise.} \end{cases}
\end{aligned}$$

Fig. 1 shows a PML model in two-dimensional case.

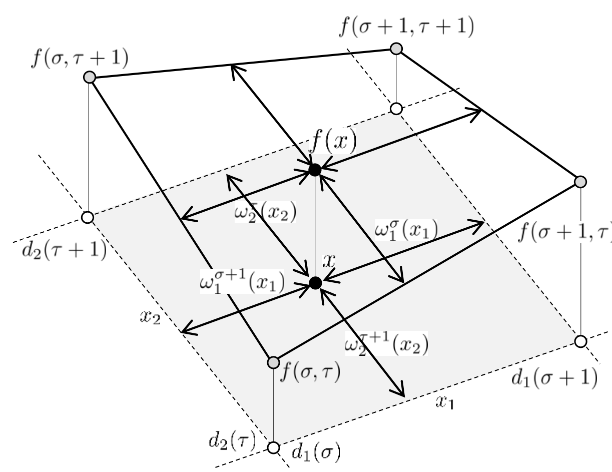


Fig. 1. PML model in two-dimensional case

## III. Piecewise Modeling

This section presents the PML optimal modeling with three variables, vertex values $f(i_1,\ldots,i_n)$ of the PML model in (4), dividing points $d_j(i_j)$ in the state-space in (3), and the number of piecewise regions, $\Pi_{i=1}^n(m_i-1)$.

In [11], two variables $f(i_1,\ldots,i_n)$ and $d_j(i_j)$ were optimized by fixing the variable $\Pi_{i=1}^n(m_i-1)$. This study proposes a modeling algorithm for constructing a PML system using numerical data. Because the PML system can also be expressed as fuzzy if-then rules, the vertex values of the PML model are determined based on a modeling algorithm using simplified fuzzy inference reasoning. This method derives the vertex values and dividing points of a piecewise model by combining simplified fuzzy inference reasoning [15] with a spline function minimization algorithm [16]. Because the calculations of the vertex values and dividing points are nonlinear programming problems, finding a globally optimal solution is difficult. In [12], the dividing points $d_j(i_j)$ of the piecewise regions were determined using the PSO algorithm because finding the optimal solution is not a linear programming problem. The proposed method can determine the optimal vertex values of the piecewise regions with minimal modeling errors.

This paper proposes a PML modeling method that combines simplified fuzzy inference reasoning and the PSO algorithm. The proposed method can optimize two variables $f(i_1,\ldots,i_n)$ and $d_j(i_j)$.

### A. PSO Algorithm

In this study, the PSO algorithm [13] was used to determine the dividing points of the PML model in the state-space. PSO is an optimization method that searches for optimal values by simulating the behavior of living animals such as school of fish or flock of birds. The PSO algorithm is simple and has several features. The first is that it is a multi-point search algorithm with multiple search points. Another feature is that it shares information about the best solution among multiple points and searches for solution space based on that information.

Particle position is represented as $x_i = (x_{i1},\ldots,x_{ij},\ldots,x_{in})^T$ in $R^n$ , where i is the particle number and $j$ is the number of dimensions. The particle positions include position $x_i$ and velocity $v_i = (v_{i1},\ldots,v_{ij},\ldots,v_{in})^T$. It has the best position data $bp = (bp_{i1},\ldots,bp_{ij},\ldots,bp_{in})^T$ and evaluation value $E(bp)$ for each particle. In addition, it has the best position data $gbp = (gbp_{i1},\ldots,gbp_{ij},\ldots,gbp_{in})^T$ and evaluation value $E(gbp)$ for all the data of the swarm group.

Moving vector $v_{ij}^{k+1}$ is generated from the weighted linear combination of present position $x_{ij}^k$, previous move vector $v_{ij}^k$, and best positions $(bp^k, gbp^k)$. Next position $x_{ij}^{k+1}$ is calculated as the sum of $x_{ij}^k$ and $v_{ij}^{k+1}$.

Particle positions and velocities in [14] are represented as

$$\begin{aligned} v_{ij}^{k+1} &= wv_{ij}^k + c_1 r_1 (bp_{ij}^k - x_{ij}^k) + c_2 r_2 (gbp_{ij}^k - x_{ij}^k), \\ x_{ij}^{k+1} &= x_{ij}^k + v_{ij}^{k+1}, \end{aligned}$$

where $r_1$ and $r_2$ are uniform random numbers between 0 and 1, $w$ is the inertia constant, and $c_1$ and $c_2$ are the local and global weights, respectively; $k$ is the number of calculations.

The PSO algorithm can be applied to a wide range of problems, including no differentiable systems because it does not require gradient information.

**Algorithm 1** PSO

1) Set $m$, $w$, $k = 0$
2) Set initial particle position $x_i^0$ and velocity $v_i^0$.

$$bp_i^0 \leftarrow x_i^0, gbp_l^0 \leftarrow bp_l^0,$$

where $l = \arg\min_i E(bp_i^0)$.

3) Calculate the velocity and position

$$\begin{aligned} &v_{ij}^{k+1} \leftarrow wv_{ij}^k + c_1 r_1 (bp_{ij}^k - x_{ij}^k) + c_2 r_2 (gbp_{ij}^k - x_{ij}^k) \\ &x_{ij}^{k+1} \leftarrow x_{ij}^k + v_{ij}^{k+1} \end{aligned}$$

4) Update $bp_i$ and $gbp$
   **if** $E(x_i^{k+1}) < E(bp_i^k)$, **then** $bp_i^{k+1} \leftarrow x_i^{k+1}$
   **else** $bp_i^{k+1} \leftarrow bp_i^k$
   **end if**
   $gbp^{k+1} \leftarrow bp_l^k$, where $l = \arg\min_i E(bp_i^k)$.
5) Exit sequence
   **if** $k \neq T$, **then** $k \leftarrow k+1$, **return** to 3)
   **end if**

### B. PML Modeling Using the PSO Algorithm

In [12], the PSO algorithm was applied to construct PML models of nonlinear systems with known model dynamics. PML modeling requires determining the dividing points to construct the piecewise regions. To construct a PML model with $4 \times 4$ piecewise regions, two optimal dividing points are required to be found in the state-space. Fig. 2 illustrates two particle positions, that is, $\times : (d_1^*(\sigma+1), d_2^*(\tau+1))$ in the first quadrant and $+ : (d_1^*(\sigma), d_2^*(\tau))$ in the third quadrant.

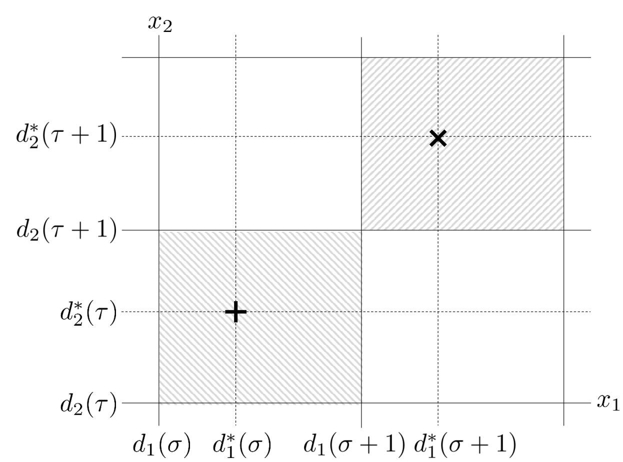


Fig. 2. Two dividing points in the first and third quadrants

The evaluation value of modeling based on the PSO algorithm is the error between the PML model and the nonlinear system. Two optimal dividing points are determined to minimize the evaluation value of the PML model. The evaluation value of the PML model is

$$E(x) = \int_{d_2(\tau)}^{d_2(\tau+1)} \int_{d_1(\sigma)}^{d_1(\sigma+1)} F(x)^T F(x) dx_1 dx_2 \qquad (5)$$

for region $R_{\sigma\tau}$, where $x = (x_1, x_2)^T$,

$$F(x) = \sum_{i_1=\sigma}^{\sigma+1} \sum_{i_2=\tau}^{\tau+1} w_1^{i_1}(x_1) w_2^{i_2}(x_2) f(i_1, i_2) - f(x)$$

### C. Determining optimal vertex values $f(i_1, \ldots, i_n)$

This subsection focuses on finding the optimal solution [11] of $f(i_1, \ldots, i_n)$ when the number of piecewise regions and dividing points in the state-space are fixed. The rule expression (1) is used instead of the state-space model (4) for the PML modeling, in order to determine the optimal vertex values. A simplified fuzzy inference reasoning algorithm [15] is applied to the rules (1). Algorithm 2 shows the fuzzy inference reasoning-based algorithm. optimal values $f^k = f(i_1, \ldots, i_n)$ are determined by the gradient descent method. In this algorithm, $(x_*(t), x_*^+(t))$ is the training data, $\tau$ is the training rate. $M$ is the number of training data and $N$ is the number of iterations. $\delta(t)$ means the calculating error.

This paper proposes an optimal method for the PML modeling. The modeling method finds the optimal dividing points $d_j(i_j)$ and vertex values $f(i_1, \ldots, i_n)$ in Fig. 1 using Algorithms 1 and 2, respectively.

**Algorithm 2** Optimal vertex values

1) Set $(x_*(t), x_*^+(t))$, $M$, $N$, $f^k$, $\tau$, $i = 1$, and $t = 1$.
2) **for** $i \leq M$
3) **for** $t \leq N$
4) Substitute $x_*(t)$ into

$$x^+(t) = \sum_{k=1}^{2^n} \mu_k(x_*(t)) f^k \quad (6)$$

5) Calculate error $\delta(t) = x^+(t) - x_*^+(t)$
6) Derive $f^k$ from

$$f^k(t+1) = f^k(t) + \tau \mu_k(x_*(t)) \delta(t)$$

7) Update $t \leftarrow +1$
8) **end for**
9) Update $i \leftarrow +1$
10) **end for**

## IV. SIMULATION RESULTS

It is not difficult for nonlinear systems, such as convex functions (e.g. $z = x^2 + y^2$), to determine the parameters of a piecewise model. Due to demonstrating the effectiveness of the proposed method, this study constructs a piecewise model of the following multimodal function with two local minima, which has complex constructions.

$$z = f(x, y) = 4x^2 - 2.1x^4 + x^6/3 + xy - 4x^2 + 4y^4 \quad (7)$$

Figs. 3 and 4 show three-dimensional and contour plots in the bounded region of the state-space $-1 \leq x, y \leq 1$.

The 60 initial positions (30 positions per quadrant) of the optimal dividing points $(d_1(i), d_2(j))$ are shown on Fig. 5. The algorithm parameters are $m = 30$, $w = 0.3$, $c_1 = 1.2$, $c_2 = 1.2$, and $T = 20$. The initial particle positions $(x^0, y^0)$

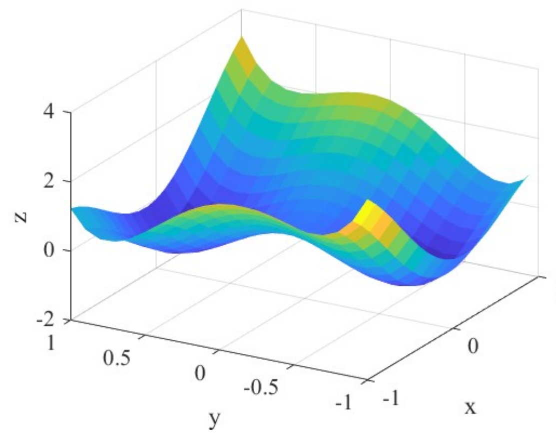


Fig. 3. Multimodal function

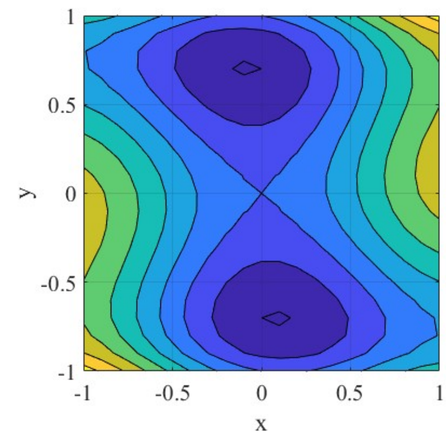


Fig. 4. Contour plots

are uniform random values bounded by $0 \leq (x^0, y^0) \leq 1$ and $-1 \leq (x^0, y^0) \leq 0$ in the first and third quadrants, respectively. $r_1$ and $r_2$ are uniform random values between 0 and 1. $v_{ij}$ is also a uniform random value bounded by $|v_{ij}| \leq 1$.

$\times$ and $+$ represent particle positions in the first and third quadrants, respectively. These particle positions represent optimal dividing point candidates. Fig. 6 shows particle positions after the fifth iteration. Fig. 7 shows particle positions after the 15th iteration. The optimal dividing points are $+ : (d_1^*(\sigma), d_2^*(\tau)) = (-0.397, -0.761)$ and $\times : (d_1^*(\sigma + 1), d_2^*(\tau+1)) = (0.312, 0.752)$.

Then, the optimal dividing points of the PML model are represented as

$$\begin{cases} x \in \{d_1(1), d_1(2), d_1(3), d_1(4), d_1(5)\} \\ \quad = \{-1.000,\ -0.397,\ 0,\ 0.312,\ 1.000\}, \\ y \in \{d_2(1), d_2(2), d_2(3), d_2(4), d_2(5)\} \\ \quad = \{-1.000,\ -0.761,\ 0,\ 0.752,\ 1.000\}, \end{cases} \quad (8)$$

Vertex values $f(i_1, i_2)$ in Table I are calculated by substituting the dividing points in (8) into the multimodal function in (7).

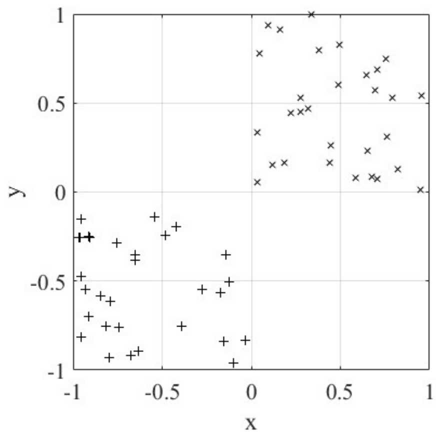


Fig. 5. Particle points initially

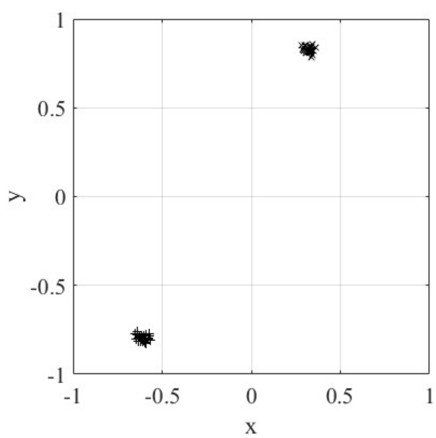


Fig. 6. Particle points after the fifth iteration

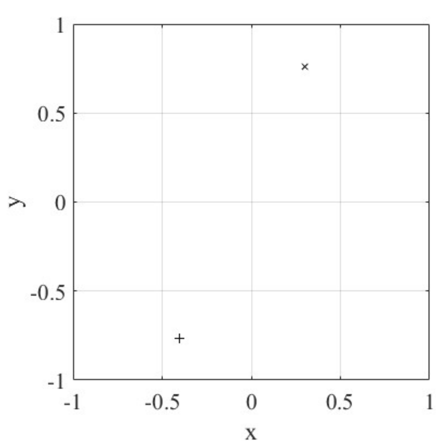


Fig. 7. Particle points after the 15th iteration

TABLE I

VERTEX VALUES $f(i_1, i_2)$ CALCULATED BY SUBSTITUTING (8) INTO (7)

| | $d_2(1)$ | $d_2(2)$ | $d_2(3)$ | $d_2(4)$ | $d_2(5)$ |
|---|---|---|---|---|---|
| $d_1(1)$ | 3.233 | 2.019 | 2.233 | 0.499 | 1.233 |
| $d_1(2)$ | 0.977 | -0.093 | 0.580 | -0.702 | 0.183 |
| $d_1(3)$ | 0 | -0.975 | 0 | -0.983 | 0 |
| $d_1(4)$ | 0.058 | -0.843 | 0.369 | -0.379 | 0.681 |
| $d_1(5)$ | 1.233 | 0.497 | 2.233 | 2.002 | 3.233 |

Optimal vertex values $f(i_1, i_2)$ using Algorithm 2 are listed in Table . In this example, the number of iterations is $M = 50$, training rate is $\tau = 1$, and initial data are the vertices in Table I.

TABLE II

VERTEX VALUES $z = f(i_1, i_2)$ USING ALGORITHM 2 AND (8)

| | $d_2(1)$ | $d_2(2)$ | $d_2(3)$ | $d_2(4)$ | $d_2(5)$ |
|---|---|---|---|---|---|
| $d_1(1)$ | 3.336 | 1.819 | 2.602 | 0.262 | 1.465 |
| $d_1(2)$ | 0.818 | -0.337 | 0.836 | -0.999 | 0.144 |
| $d_1(3)$ | -0.227 | -1.096 | 0 | -1.434 | -0.022 |
| $d_1(4)$ | 0.013 | -1.086 | 0.669 | -0.699 | 0.779 |
| $d_1(5)$ | 1.191 | 0.324 | 2.464 | 1.812 | 3.244 |

For comparison with the conventional method, the following vertices dividing x and y into equal intervals were considered.

$$\begin{cases} x \in \{d_1(1), d_1(2), d_1(3), d_1(4), d_1(5)\} \\ \quad = \{-1.000,\ -0.500,\ 0,\ 0.500,\ 1.000\}, \\ y \in \{d_2(1), d_2(2), d_2(3), d_2(4), d_2(5)\} \\ \quad = \{-1.000,\ -0.500,\ 0,\ 0.500,\ 1.000\} \end{cases} \tag{9}$$

Vertex values $f(i_1, i_2)$ in Table III are calculated by substituting the dividing points in (9) into the multimodal function in (7).

TABLE III

VERTEX VALUES $z = f(i_1, i_2)$ CALCULATED BY SUBSTITUTING (9) INTO (7)

| | $d_2(1)$ | $d_2(2)$ | $d_2(3)$ | $d_2(4)$ | $d_2(5)$ |
|---|---|---|---|---|---|
| $d_1(1)$ | 3.233 | 1.983 | 2.233 | 0.983 | 3.233 |
| $d_1(2)$ | 1.374 | 0.374 | 0.874 | -0.126 | 1.374 |
| $d_1(3)$ | 0 | -0.750 | 0 | -0.750 | 0 |
| $d_1(4)$ | 0.374 | -0.126 | 0.874 | 0.374 | 0.374 |
| $d_1(5)$ | 1.233 | 0.983 | 2.233 | 1.983 | 1.233 |

Figs 8 and 9 show PML models with $4 \times 4$ regions. Fig. 8 shows a PML model constructed using the dividing points in (8) and vertex values in Table II calculated by PSO and learning methods described in Algorithms 1 and 2. Fig. 9 shows a PML model constructed using the vertices in (9) and Table III. Table IV shows the modeling errors $E(x)$ in (5) of the three PML models with the dividing points and vertex values. Therefore, the simulation results indicate that the PML model constructed using Algorithms 1 and 2 achieves good modeling performance in terms of modeling error.

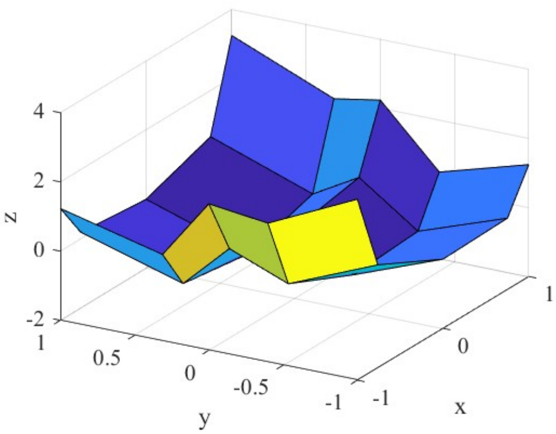


Fig. 8. PML model with the dividing points (8) and vertex values in Table II

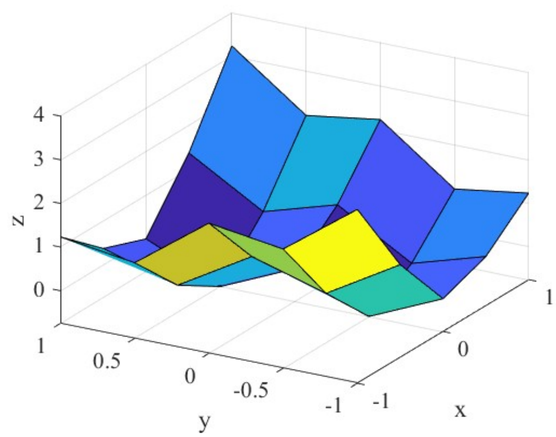


Fig. 9. PML model with the dividing points at equal intervals (9) and vertex values in Table III

The objective of this study is to construct a piecewise model with as few modeling errors as possible with a small number of piecewise models, using PML models that can perform stability analysis and stabilization. This study has no particular indicator that the modeling error should be reduced. However, it is important to use the value of modeling error as an indicator. I plan to conduct future research on robust stability and modeling errors.

TABLE IV
MODELING ERRORS OF THE PML MODELS

| Dividing points | Vertex values | Error |
|---|---|---|
| (7) by Algorithm 1 | Table II by Algorithm 2 | 5.101 |
| (7) by Algorithm 1 | Table I | 7.157 |
| (8) at equal intervals | Table III | 53.773 |

## V. CONCLUSION

This study developed a piecewise modeling method using PSO and a learning algorithm based on simplified fuzzy inference reasoning. The piecewise model has the shape of a rectangular partition of the state-space The proposed algorithm determined optimal vertices of the piecewise regions with minimal modeling errors. The numerical simulation results were performed to demonstrate the effectiveness of the algorithm.

I will consider PML models with many piecewise regions for complex nonlinear systems and apply them to multi-dimensional nonlinear systems and conduct some researches on robust stability and modeling errors in the future.

## VI. ACKNOWLEDGMENTS

This study was supported by Japan Society Promotion of Science KAKENHI Grant No. JP21K12051.

## REFERENCES


[1] F. He and Y. Yang, "Nonlinear system identification of neural systems from neurophysiological signals," *Neuroscience*, vol. 458, pp. 213–228, 2021.

[2] H. Eivazi, H. Veisi, M. H. Naderi, and V. Esfahanian, "Deep neural networks for nonlinear model order reduction of unsteady flows," *Physics of Fluids*, vol. 32, no. 10, p. 105104, 10 2020.

[3] T. Tao, G. Zhao, J. Zhai, and S. Ren, "A strong adaptive piecewise model order reduction method for large-scale dynamical systems with viscoelastic damping," *Mechanical Systems and Signal Processing*, vol. 164, p. 108203, 2022.

[4] C. Tan, J. Zhang, H. Yang, and L. Zhang, "Piecewise model-based adaptive compensation control for high-speed trains with unknown actuator failures," *Transportation Safety and Environment*, vol. 5, no. 4, p. tdad003, 02 2023.

[5] A.-T. Nguyen, T. Taniguchi, L. Eciolaza, V. Campos, R. Palhares, and M. Sugeno, "Fuzzy control systems: Past, present and future," *IEEE Computational Intelligence Magazine*, vol. 14, no. 1, pp. 56–68, 2019.

[6] P. Liu, W. Leng, and W. Fang, "Training anfis model with an improved quantum-behaved particle swarm optimization algorithm," *Mathematical Problems in Engineering*, vol. 2013, no. 1, pp. 595–639, 2013.

[7] X. Wang, Y. Chen, J. Jin, and B. Zhang, "Fuzzy-clustering and fuzzy network based interpretable fuzzy model for prediction," *Scientific Reports*, vol. 12, no. 16279, 2022.

[8] M. Sugeno and T. Taniguchi, "On improvement of stability conditions for continuous mamdani-like fuzzy systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 1, pp. 120–131, 2004.

[9] T. Taniguchi and M. Sugeno, "Stabilization of nonlinear systems with piecewise bilinear models derived from fuzzy if-then rules with singletons," in *International Conference on Fuzzy Systems*, 2010, pp. 1–6.

[10] ——, "Stabilization of discrete-time nonlinear systems using piecewise multi-linear control based on input-output feedback linearization," in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 2019, pp. 1582–1587.

[11] T. Taniguchi, L. Eciolaza, and M. Sugeno, "Piecewise control system design based on modeling algorithms using numerical data," in *Proceedings of the 2023 6th International Conference on Electronics, Communications and Control Engineering*, 2023, p. 290–294.

[12] T. Taniguchi and L. Eciolaza, "Piecewise modeling based on particle swarm optimization algorithms using numerical data," in *IEEE WCCI 2024*, 2024, (Accepted).

[13] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, 1995, pp. 1942–1948 vol.4.

[14] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)*, 1998, pp. 69–73.

[15] H. Ichihashi and T. Watanabe, "Learning control by fuzzy models using a simplified fuzzy reasoning," *Journal of Japan Society for Fuzzy Theory and Systems*, vol. 2, pp. 429–437, 1990, (in Japanese).

[16] K. Ichida and F. Yoshimoto, *Spline functions and their applications*. Kyoiku-shuppan, 1979, (in Japanese).