AUT Journal of Modeling and Simulation

AUT J. Model. Simul., 57(2) (2025) 191-214
DOI: 10.22060/miscj.2026.25021.5447

Complete Automated Structure Discovery and Parameter Estimation for Piecewise Affine Models with Guaranteed Convergence

Mohammad Lotfi [1], Mohammad Bagher Menhaj [1], Mehdi Karrari [1], Mohammad Haeri [2]

[1] Department of Electrical Engineering, Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran.
[2] Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran.

ABSTRACT: This paper presents a novel, fully automated framework for the complete structure discovery and parameter estimation of Piecewise Affine (PWA) models. To the best of our knowledge, this is the first approach that simultaneously determines the number of submodels, their orders, parameter vectors, and polyhedral partitions from data, without any prior structural knowledge or the need for tuning parameters. The methodology integrates three key innovations: (1) Automated submodel order selection via Orthogonal Least Squares with an Error-to-Signal Ratio test; (2) A clustering-based algorithm for determining the number of submodels and generating a robust initial labeled dataset; and (3) An iterative algorithm that integrates a novel self-labeling support vector machine (SL-SVM) for estimating polyhedral partitions with a recursive least squares (RLS) scheme for refining submodel parameters, both with guaranteed convergence. Theoretical analysis demonstrates both computational efficiency and convergence properties, with the SL-SVM algorithm significantly reducing complexity compared to standard SVM. Extensive simulations validate the framework's performance across multiple benchmark systems, achieving Best Fit Rates exceeding 98% in scenarios of complete structural uncertainty. The approach consistently outperforms existing methods in accuracy while maintaining computational efficiency. Furthermore, we demonstrate the method's applicability to nonlinear system identification through PWARX approximation, showcasing its versatility for practical engineering applications. The proposed framework represents a significant advancement in automated system identification, providing a comprehensive solution for black-box modeling of hybrid and nonlinear systems.

Review History:
Received: Oct. 31, 2025
Revised: Nov. 04, 2025
Accepted: Dec. 26, 2025
Available Online: Dec. 31, 2025

Keywords:
Piecewise Affine Models
Structure Discovery
Parameter Estimation
Hyperspherical Partitioning
Self-Labeling SVM
Convergence Analysis

1- Introduction

Piecewise Affine (PWA) models provide a powerful and flexible framework for modeling hybrid and nonlinear dynamical systems. By combining multiple local affine submodels, they can accurately capture complex system behaviors [1, 2]. Such models have been extensively employed in a variety of domains, including environmental systems [3, 4], industrial process control [5], and power electronics [6]. They are particularly valuable in safety-critical applications, such as the supervisory control of complex nuclear reactor systems [7, 8]. For instance, PWA models are instrumental in fault diagnosis of electromechanical actuators [9], modeling of hybrid dynamics in automotive systems [10], and piecewise affine modeling of building thermal dynamics for temperature control in smart buildings [11]. Within this framework, the Piecewise Affine AutoRegressive Exogenous (PWARX) model forms a particularly important subclass for input–output system identification [12]. Accurate identification of PWARX models from data is essential for model-based analysis [13], controller synthesis [14, 15], and fault detection [16]. However, this task is inherently challenging, as it requires the simultaneous solution of three interdependent problems. These include: (i) structure discovery, i.e., determining the number and orders of submodels; (ii) submodel parameter estimation; and (iii) polyhedral partition estimation that defines the operational regions in the regression space.

Existing methods for PWARX identification can be broadly classified into clustering-based [5], bounded-error [17], Bayesian [18], algebraic [19], and machine learning–based approaches [20]. The pursuit of accurate data-driven models for complex systems lies at the intersection of classical control theory and modern intelligent systems [21]. Despite their methodological diversity, these approaches share key limitations. Many clustering and statistical methods, such as those in [22] and [23], assume prior knowledge of the model structure—specifically, the number and order of submodels. This assumption severely limits their applicability to black-box identification problems. In addition, Support Vector Machine (SVM)-based techniques used for partition estimation typically require fully labeled datasets, resulting in high computational costs and poor scalability when applied to large datasets. While contemporary methods like physics-

*Corresponding author's email: menhaj@aut.ac.ir

aware neural networks demonstrate significant potential for system management [24], the models they produce are often black-boxes, limiting their interpretability for tasks requiring structural transparency.

Recent studies have attempted to address the structure discovery problem, although often under the assumption that submodel orders are known. For example, [25] employed the DBSCAN algorithm to estimate the number of submodels, while [26] and [20] proposed regularization-based shrinking strategies and clustering-based algorithms, respectively, for submodel number estimation. Nevertheless, to the best of our knowledge, no existing work has tackled the complete structure discovery problem—jointly determining both the number and the orders of submodels directly from measured data.

To overcome these challenges, this paper proposes a fully automated framework for complete PWARX system identification without requiring any prior structural information. The proposed method jointly estimates the number of submodels, their orders, parameter vectors, and polyhedral partitions. The main innovations of this work can be summarized in three key aspects, as follows. First, an automated submodel order selection procedure is developed, combining Orthogonal Least Squares (OLS) with the Error-to-Signal Ratio (ESR) test, which can also be effectively integrated into other PWARX identification schemes. Second, a clustering-based algorithm is introduced to determine the number of submodels and provide robust initial parameter estimates. Finally, an iterative estimation algorithm is developed, which incorporates a computationally efficient Self-Labeling Support Vector Machine (SL-SVM) for partition estimation, coupled with a recursive least squares scheme for parameter refinement. Theoretical analysis rigorously proves the convergence of both the submodel parameters and the classifiers defining the partitions. Moreover, simulation studies demonstrate that the proposed framework achieves higher modeling accuracy—measured by the Best Fit Rate (BFR)—compared to existing identification methods, without the need for sensitive tuning parameters.

The remainder of this paper is organized as follows. Section 2 formulates the identification problem. Section 3 details the proposed three-stage approach. Simulation results are presented in Section 4. Finally, Section 5 concludes the paper. Appendices include computational complexity analysis and convergence proofs.

For clarity and ease of reference, the main notations used throughout the paper are summarized in Table 1.

2- Problem Formulation

2- 1- PWARX Model

A PWARX model with n_u inputs ($\underline{u} = [u_1 \cdots u_{n_u}]^T \in \mathbb{R}^{n_u}$), an output $y \in \mathbb{R}$, and s submodels is formulated as:

$$y(k) = \begin{cases} \begin{bmatrix} \underline{\varphi}(k) \\ 1 \end{bmatrix}^T \underline{\theta}_1 + e(k) & \text{if } \underline{\varphi}(k) \in \Phi_1 \\ \vdots \qquad\qquad\qquad \vdots & \\ \begin{bmatrix} \underline{\varphi}(k) \\ 1 \end{bmatrix}^T \underline{\theta}_s + e(k) & \text{if } \underline{\varphi}(k) \in \Phi_s \end{cases}, \tag{1}$$

where $e(k) \in \mathbb{R}$ is the noise signal, and $\{\underline{\theta}_i\}_{i=1}^s$ are the parameter vectors of the submodels. The regression vector $\underline{\varphi}(k)$ is defined as:

$$\underline{\varphi}(k) = [y(k-1) \cdots y(k-n_a)\ \ \underline{u}^T(k-1) \cdots \underline{u}^T(k-n_b)]^T, \tag{2}$$

which depends on the past n_a outputs and n_b inputs. Thus, $\underline{\varphi}(k) \in \mathbb{R}^{n_\varphi}$ with $n_\varphi = n_a + n_u \times n_b$. For simplicity, Defining the extended regression vector as $\underline{\tilde{\varphi}}(k) \triangleq [\underline{\varphi}^T(k)\ \ 1]^T$ simplifies the PWARX model to:

$$y(k) = \begin{cases} \underline{\tilde{\varphi}}^T(k)\underline{\theta}_1 + e(k) & \text{if } \underline{\varphi}(k) \in \Phi_1 \\ \vdots \qquad\qquad\quad \vdots & \\ \underline{\tilde{\varphi}}^T(k)\underline{\theta}_s + e(k) & \text{if } \underline{\varphi}(k) \in \Phi_s \end{cases}. \tag{3}$$

The regions $\{\Phi_i\}_{i=1}^s$ form a complete partition of the regression space Φ, satisfying $\bigcup_{i=1}^s \Phi_i = \Phi$ and $\Phi_i \cap \Phi_j = \varnothing$ for $i \neq j$. Each region Φ_i is a convex polyhedron [17], defined by:

$$\Phi_i = \{\underline{\varphi}(k) \in \mathbb{R}^{n_\varphi} | H_i \underline{\tilde{\varphi}}(k) \preccurlyeq_{[i]} \underline{0}\}, \tag{4}$$

where $H_i \in \mathbb{R}^{p_i \times (n_\varphi + 1)}$ is the hyperplane matrix. Each row of H_i defines a separating hyperplane:

$$H_i = [\underline{h}_{1,i}^T \quad \cdots \quad \underline{h}_{p_i,i}^T], \tag{5}$$

and $p_i \in \mathbb{N}^+$ is the number of separating hyperplanes enclosing Φ_i, with a maximum of $s-1$. The symbol "$\preccurlyeq_{[i]}$" denotes a componentwise inequality (combining $\leq$ and $<$) to ensure strict boundaries and prevent overlapping regions.

2- 2- Problem Statement and Assumptions

The identification problem in this paper is based on the following assumptions:

Assumption 1: The submodel orders are unknown.

Assumption 2: The number of submodels s is unknown.

Assumption 3: The noise $e(k)$ is zero-mean Gaussian white noise with finite variance σ^2.

Assumption 4: The input signal $\underline{u}(k)$ is persistently exciting .

Under these assumptions, the problem is stated as follows:

Problem 1: Given a dataset $\mathcal{D} = \{(\underline{u}(k), y(k))\}_{k=1}^N$, estimate the number of submodels s , the order of submodels

Table 1. Main notations used in the paper.

Scalars, vectors, and matrices			
Notation	**Description**	**Notation**	**Description**
x	A scalar variable	$\underline{x}$	A vector composed of elements $x_1, x_2, \dots, x_n$
X	A matrix	$\mathbf{1}_N$	A vector of ones with length N
I	Identity matrix		
Operations			
Notation	**Description**		
$\underline{x}^T, X^T$	The transpose of the vector $\underline{x}$ or the matrix X		
$\|\underline{x}\|$	Euclidean ($\mathcal{L}_2$) norm of vector $\underline{x}$, $\|\underline{x}\| = \sqrt{\sum_{i=1}^{n} {x_i}^2}$		
Specific notations			
Notation	**Description**		
$\mathcal{D}, N$	Collected dataset for system identification and the number of its samples		
$\underline{\theta}_i, \underline{\hat{\theta}}_i, \underline{\hat{\theta}}_i^{(0)}$	Parameter vector of the i-th submodel, its estimation, and its initial value		
P_i	Covariance matrix of $\underline{\theta}_i$		
s, n	Number and order of submodels		
u, y	System input and output		
e	Measurement noise		
k	Discrete time step		
q	Discrete variable or mode		
$\underline{\varphi}, \underline{\tilde{\varphi}}$	Regression and extended regression vectors		
H_i	Hyperplane matrix of the i-th submodel		
Φ_i	Polyhedron partion of the i-th submodel		
J	Cost function		
$\mathcal{HC}_i$	Hyperspherical cluster with the center $(\underline{\varphi}(k_{c_i}), y(k_{c_i}))$		
$\mathcal{HC}_i^{\mathcal{K}}$	Key hyperspherical cluster with the center $(\underline{\varphi}(k_{c_i}), y(k_{c_i}))$		
r	Radius of hyperspherical clusters		
l	Number of hyperspherical clusters		
$R^2(\mathcal{HC}_i)$	Coefficient of determination (R^2) of he i-th hyperspherical cluster		
$\mathcal{D}_P^{(i)}$	Labeled data set in the i -th iteration		
Φ	Regression space		
$\|\mathcal{D}\|$	Cardinality of the dataset $\mathcal{D}$		

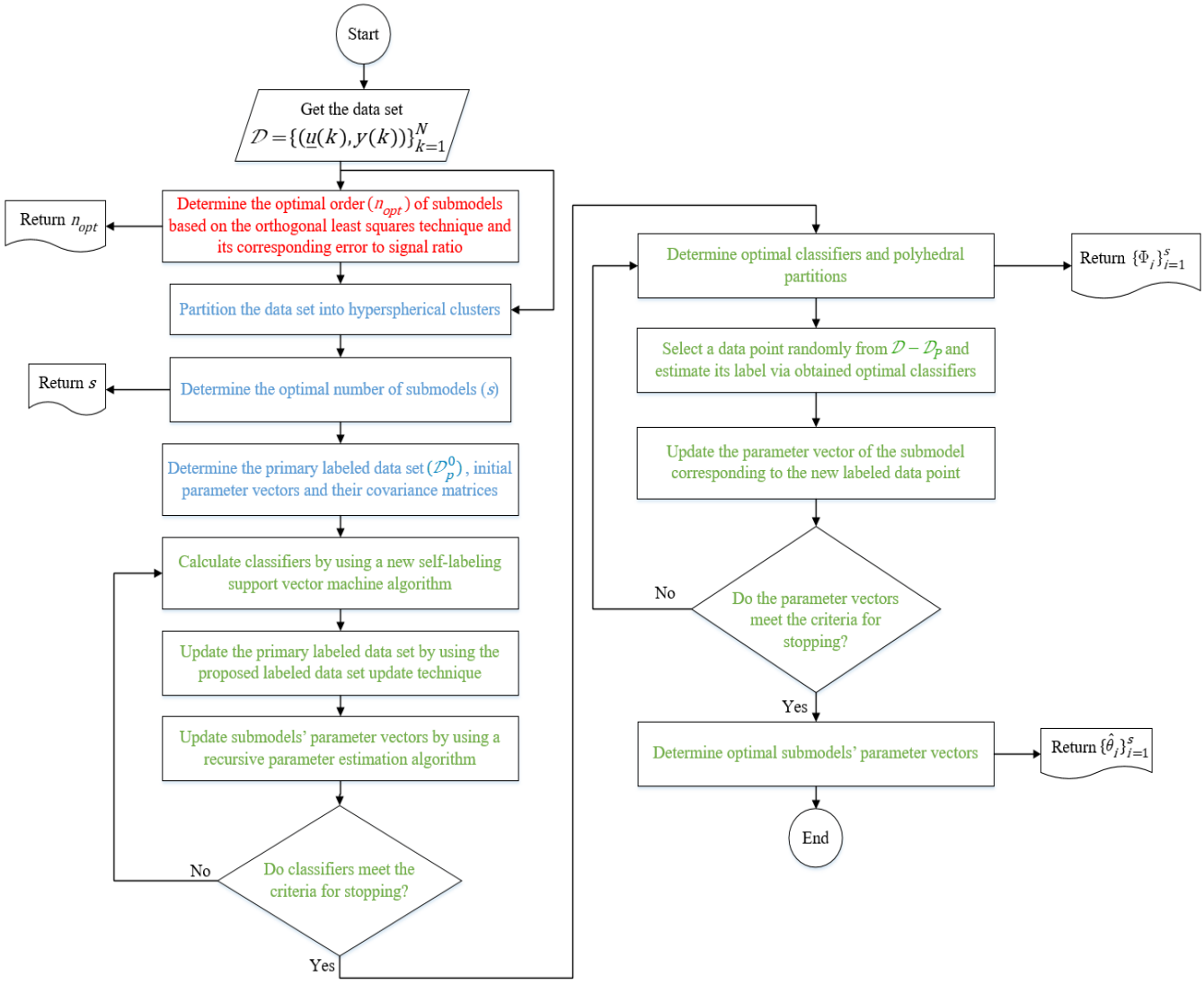

Fig. 1. The proposed identification approach flowchart (the first, second, and third algorithms are shown by the red, blue, and green blocks, respectively).

(n_a, n_b), the parameter vectors $\{\underline{\theta}_i\}_{i=1}^{s}$, and the polyhedron partitions $\{\Phi_i\}_{i=1}^{s}$.

3- The proposed offline identification approach

This section presents the proposed framework for addressing Problem 1, with the overall workflow illustrated in Figure 1. The methodology integrates three consecutive algorithms into a coherent identification pipeline. The first algorithm determines the optimal order of the submodels through Orthogonal Least Squares (OLS) regression coupled with an Error-to-Signal Ratio (ESR) test. Building on this, the second algorithm performs initial structural identification using a clustering-based approach to estimate the number of submodels. This stage also provides initial parameter vectors and generates a primary labeled dataset. These outputs serve as inputs to the third algorithm, which executes an iterative refinement process. Utilizing the novel Self-Labeling Support Vector Machine (SL-SVM), this final stage simultaneously optimizes classifier parameters to obtain separating hyperplanes while refining the submodels' parameter vectors. The following subsections detail the implementation of each algorithm.

3- 1- First Algorithm: Submodel Order Selection via OLS and ESR Test

Accurate model order selection is particularly critical in PWARX model identification. This algorithm determines the optimal submodel order from dataset $\mathcal{D}=\{(\underline{\varphi}(k),y(k))\}_{k=1}^{N}$. using Orthogonal Least Squares (OLS) with an Error-to-Signal Ratio (ESR) test. The procedure consists of three stages: orthogonal regressor determination, ESR calculation, and optimal order evaluation. We begin by defining the model's dictionary set:

Definition 1 (Model Dictionary Set): The dictionary set

DIC_m for a model with m regressors contains all regressors present in the model.

From equation (3), each PWARX submodel follows the linear regression form:

$$y(k) = \underline{\tilde{\varphi}}^T(k)\underline{\theta}' + e(k) = \sum_{i=1}^{m} \theta_i' \tilde{\varphi}_i(k) + e(k), \quad (6)$$

where $m = n_\varphi + 1 = n_a + n_b + 1$, $\underline{\theta}' = [\theta_1' \cdots \theta_m']^T$, and

$$\begin{aligned} \underline{\tilde{\varphi}}(k) &= [\tilde{\varphi}_1(k) \ \cdots \ \tilde{\varphi}_{n_a}(k) \ \tilde{\varphi}_{n_a+1}(k) \ \cdots \ \tilde{\varphi}_{n_a+n_b}(k) \ \tilde{\varphi}_m(k)]^T, \\ &= [y(k-1) \ \cdots \ y(k-n_a) \ u(k-1) \ \cdots \ u(k-n_b) \ 1]^T. \end{aligned} \quad (7)$$

we set $n_b = n_a = n \in \mathbb{N}$ without loss of generality, yielding $m = 2n + 1$. The dictionary set becomes:

$$\begin{aligned} \mathrm{DIC}_m &= \{\tilde{\varphi}_1(k), \cdots, \tilde{\varphi}_n(k), \tilde{\varphi}_{n+1}(k), \cdots, \tilde{\varphi}_{2n}(k), \tilde{\varphi}_{m=2n+1}(k)\}, \\ &= \{y(k-1), \cdots, y(k-n), u(k-1), \cdots, u(k-n), 1\}. \end{aligned} \quad (8)$$

To apply OLS, we transform equation (6) into an orthogonal regressor form:

$$y(k) = \sum_{i=1}^{m} \beta_i \xi_i(k) + e(k). \quad (9)$$

where orthogonal regressors $\{\underline{\xi}_i(k)\}_{i=1}^m$ satisfy:

$$\sum_{k=1}^{N} \xi_i(k)\,\xi_j(k) = \begin{cases} 0 & \text{if } i \neq j \\ d_i > 0 & \text{if } i = j \end{cases}. \quad (10)$$

Using the Gram-Schmidt procedure inspired by [27], we compute orthogonal regressors recursively:

$$\begin{cases} \xi_1(k) = \tilde{\varphi}_1(k) \\ \xi_2(k) = \tilde{\varphi}_2(k) + \gamma_{1,2}\xi_1(k) \\ \xi_3(k) = \tilde{\varphi}_3(k) + \gamma_{1,3}\xi_1(k) + \gamma_{2,3}\xi_2(k) \\ \vdots \\ \xi_M(k) = \tilde{\varphi}_M(k) + \sum_{j=1}^{M-1} \gamma_{j,M}\xi_j(k) \\ \vdots \\ \xi_m(k) = \tilde{\varphi}_m(k) + \sum_{j=1}^{m-1} \gamma_{j,m}\xi_j(k) \end{cases}, \quad (11)$$

where

$$\gamma_{j,M} = -\frac{\sum_{k=1}^{N} \tilde{\varphi}_M(k)\xi_j(k)}{\sum_{k=1}^{N} \xi_j^2(k)}; \ M = 2,3,\cdots,m; \ 1 \le j \le M-1. \quad (12)$$

The orthogonal coefficients are determined as:

$$\beta_i = \frac{\sum_{k=1}^{N} y(k)\xi_i(k)}{\sum_{k=1}^{N} \xi_i^2(k)}, \ i \in \{1,2,\cdots,m\}. \quad (13)$$

The Error-to-Signal Ratio is defined as:

$$\mathrm{ESR} \triangleq \frac{\sum_{k=1}^{N} e^2(k)}{\sum_{k=1}^{N} y^2(k)}. \quad (14)$$

Defining vectors $\underline{\xi}_i \triangleq [\xi_i(1)\ \xi_i(2)\cdots\xi_i(N)]^T$ for $i = 1,2,\cdots,m$, $\underline{y} \triangleq [y(1)y(2)\cdots y(N)]^T$, $\underline{e} \triangleq [e(1)e(2)\cdots e(N)]^T$ using the dataset $\mathcal{D}$, and using the whiteness of $e(k)$ (Assumption 3), we derive the output variance decomposition:

$$\begin{aligned} &\mathrm{var}(\underline{y}) = \frac{1}{N}\underline{y}^T\underline{y} = \frac{1}{N}\sum_{k=1}^{N} y^2(k) = \\ &\frac{1}{N}\sum_{k=1}^{N}\left(\sum_{i=1}^{m} \beta_i\xi_i(k) + e(k)\right)^2 \\ &\mathrm{var}(\underline{y}) = \frac{1}{N}\sum_{i=1}^{m}\sum_{k=1}^{N} \beta_i^2\xi_i^2(k) + \frac{1}{N}\sum_{k=1}^{N} e^2(k) = \\ &\frac{1}{N}\sum_{i=1}^{m} \beta_i^2\,\underline{\xi}_i^T\underline{\xi}_i + \frac{1}{N}\underline{e}^T\underline{e} \end{aligned} \quad (15)$$

The i-th Error Reduction Ratio is defined as:

$$\mathrm{ERR}(i) \triangleq \frac{\beta_i^2\,\underline{\xi}_i^T\underline{\xi}_i}{\underline{y}^T\underline{y}}, \quad i = 1,2,\ldots,m. \quad (16)$$

This leads to the key relationships:

$$\underline{e}^T\underline{e} = \underline{y}^T\underline{y} + \sum_{i=1}^{m} \beta_i^2\,\underline{\xi}_i^T\underline{\xi}_i, \quad (17)$$

$$\frac{\underline{e}^T\underline{e}}{\underline{y}^T\underline{y}} = 1 - \sum_{i=1}^{m} \mathrm{ERR}(i). \quad (18)$$

Therefore, the ESR can be expressed as:

$$\mathrm{ESR} = \frac{\sum_{k=1}^{N} e^2(k)}{\sum_{k=1}^{N} y^2(k)} = \frac{\underline{e}^T\underline{e}}{\underline{y}^T\underline{y}} = 1 - \sum_{i=1}^{m} \mathrm{ERR}(i). \quad (19)$$

3- 1- 1- Algorithm Execution

The algorithm iterates over candidate orders $n = 0,1,2,\ldots,n_{\max}$. For each candidate order, the procedure follows these steps:

1. Construct the dictionary set $\mathrm{DIC}_{m=2n+1}$ using equation (8).
2. Compute orthogonal regressors $\{\underline{\xi}_i(k)\}_{i=1}^m$ via the Gram-Schmidt procedure in equation (11).

3. Calculate the error-to-signal ratio $esr\left(n\right)$ using equation (19).

The optimal order (n_{opt}) is selected as the smallest order n where the variance of subsequent ESR values stabilizes below the threshold $\rho = 10^{-2}$, satisfying the condition:

$$\mathrm{var}\big(\mathrm{esr}(n), \mathrm{esr}(n+1), \dots, \mathrm{esr}(n_{max})\big) \leq \rho. \quad (20)$$

This selection criterion effectively balances model parsimony with accuracy, accommodating practical scenarios where ESR approaches a small nonzero constant due to measurement noise rather than reaching zero. The complete order selection procedure is formalized in Algorithm 1, and this methodology can be readily incorporated into various PWARX identification frameworks, including bounded error, algebraic, and Bayesian approaches.

3- 2- The second algorithm: Clustering-based algorithm

The second algorithm estimates the optimal number of submodels ss, generates an initial labeled dataset, and provides initial estimates for the submodel parameter vectors and their covariance matrices. Based on clustering, this algorithm begins by partitioning the dataset set $\mathcal{D}$ into hyperspherical clusters (HCs). The following definitions formalize key concepts:

Definition 2 (Hyperspherical Cluster): A hyperspherical cluster $\mathcal{HC}$ centered at $\left(\underline{\varphi}\left(k_c\right), y\left(k_c\right)\right) \in \mathcal{D}$ with $k_c \in \{1,2,\dots,N\}$ is defined as:

$$\mathcal{HC} = \{(\underline{\varphi}(k), y(k)) | d(\underline{\varphi}(k_c), \underline{\varphi}(k)) < r, \forall (\underline{\varphi}(k), y(k)) \in \mathcal{D}\} \quad (21)$$

where $d\left(.,.\right)$ denotes the Euclidean distance between two vectors, defined as $d\left(\underline{x}, \underline{x}'\right) = \sqrt{\sum_{i=1}^{n_x} (x_i - x_i')^2}$ for $\underline{x} = [x_1 \cdots x_{n_x}]^T$ and $\underline{x}' = [x_1' \cdots x_{n_x}']^T$.

Definition 3 (Pure and Mixed Hyperspherical Cluster): A hyperspherical cluster is termed pure if all its data points belong to a single subsystem; otherwise, it is mixed.

Definition 4 (Key Hyperspherical Cluster set): A key hyperspherical cluster set (KHCS) is a collection of pure hyperspherical clusters, each associated with a distinct subsystem. Mathematically:

$$\text{KHCS: } \{\mathcal{HC}_z | \mathcal{HC}_z \subset \Phi_z \ , z = 1,2,\dots,s\} \quad (22)$$

Algorithm 1: The first algorithm to determine the optimal submodels' order

Inputs:

Data set $\mathcal{D} = \{(\underline{\varphi}(k), y(k))\}_{k=1}^{N}$, coefficient ρ, upper bound n_{max}

Output:

optimal submodels' order (n_{opt})

1: Set $n = 0$
2: **for** $n = 0,1,\dots,n_{max}$ **do**
3: Let $m = 2n + 1$
4: Determine the dictionary set of the model, DIC_m, according to (8).
5: By using (11) and (12), obtain the orthogonal regressors corresponding to the regressors in DIC_m.
6: **for** $i = 1,2,\dots,m$ **do**
7: Calculate β_i using (13).
8: Calculate $\mathrm{ERR}(i)$using (16).
9: **end for**
10: Calculate the ESR value according to (19).
11: Let $\mathrm{esr}(n) = \mathrm{ESR}$
12: **end for**
13: **for** $n = 0,1,\dots,n_{max}$ **do**
14: Calculate $\mathrm{var}\big(\mathrm{esr}(n), \mathrm{esr}(n+1), \dots, \mathrm{esr}(n_{max})\big)$
15: If condition (20) is met, exit the loop
16: **end for**
17: Select $n_{opt} = n$
18: **return** $\boldsymbol{n_{opt}}$

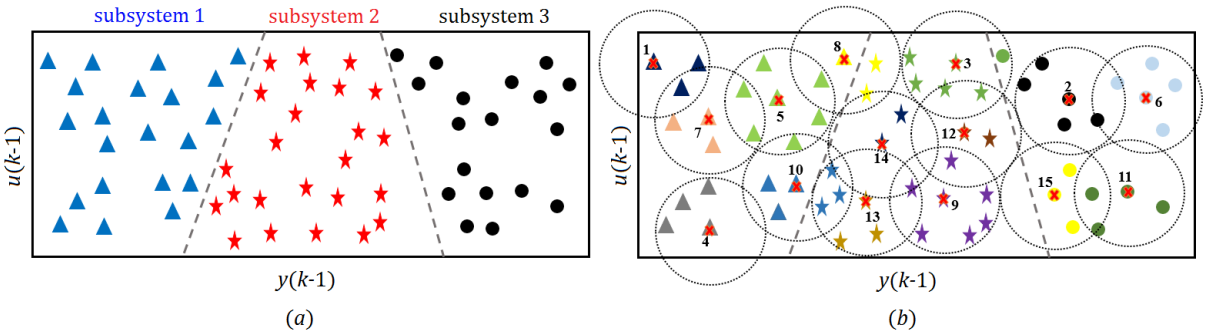

Fig. 2. Hyperspherical clustering of PWARX system data: (a) original dataset, (b) obtained clusters.

The z-th member of the KHCS set is indicated as $\mathcal{HC}_z^k$.

The algorithm comprises five steps:

1. Partition the dataset $\mathcal{D}$ into hyperspherical clusters.
2. Fit a linear regression model to each hyperspherical cluster.
3. Identify a key hyperspherical cluster set and estimate the optimal number of submodels.
4. Compute initial values for the submodels' parameter vectors and their covariance matrices.
5. Generate a primary labeled dataset.

Each step is detailed in the subsequent subsections.

3- 2- 1- Data set partitioning into hyperspherical clusters

The first step of the second algorithm partitions the dataset $\mathcal{D}$ into hyperspherical clusters. This iterative process yields, at the i-th iteration, a cluster denoted as $\mathcal{HC}_i = \{(\breve{\varphi}(k), y(k)\}_{k=1}^{N_{c_i}}$, where N_{c_i} is the number of data points in the hyperspherical cluster.

For a fixed radius r, the partitioning procedure at each iteration $i = 1,2,3,\ldots$ involves four sequential steps:

1. A center point $\left(\varphi(k_{c_i}), y(k_{c_i})\right) \in \mathcal{D}$ is randomly selected from the current dataset $\mathcal{D}$.

2. Using Equation (21), the hyperspherical cluster $\mathcal{HC}_i$ is constructed, comprising all data points within the hypersphere of radius r centered at the selected point.

3. The dataset is updated by removing the points in $\mathcal{HC}_i$, i. e., as $\mathcal{D} \setminus \mathcal{HC}_i$.

4. The process repeats until the dataset is empty.

Upon completion, the procedure yields a set of hyperspherical clusters $\{\mathcal{HC}_i\}_{i=1}^{l}$, where $l \in \mathbb{N}$ is the total number of clusters generated. Figure 2 provides an illustrative example for a PWARX system with three first-order subsystems. Figure 2(a) displays the data points of the original dataset $\mathcal{D}$ in the regression space, with dashed lines indicating subsystem boundaries (separating hyperplanes). Figure 2(b) shows the resulting hyperspherical clusters after partitioning, where each cluster's points are enclosed in a circle (a 2D projection of the hypersphere) of the same color. Centers of the hyperspherical clusters are marked with cross symbols. According to Definition 3, clusters 3, 8, and 10 are mixed, while the remaining clusters are pure.

Since cluster centers are chosen randomly, each execution of the partitioning procedure may produce a different set of clusters with a varying number l. The resulting clusters satisfy $\bigcup_{i=1}^{l} \mathcal{HC}_i = \mathcal{D}$, and $\mathcal{HC}_i \cap \mathcal{HC}_j = \varnothing$ for all $i \neq j$. The choice of the radius r in Equation (21) significantly impacts the algorithm's performance; guidelines for selecting an appropriate value are detailed in Section 3-2-4.

After obtaining the hyperspherical clusters, the next step is to determine a Key Hyperspherical Cluster Set (KHCS) and estimate the optimal number of submodels, as explained in the following subsection.

3- 2- 2- Determining a KHCS and the optimal number of submodels

A necessary condition for solving Problem 1 using the proposed approach is that at least one pure hyperspherical cluster exists for each subsystem of the PWARX system. This condition is readily satisfied by selecting an appropriate radius r. As defined in Definition 4, a Key Hyperspherical Cluster Set (KHCS) comprises pure hyperspherical clusters, each corresponding to a distinct subsystem. Multiple KHCS may exist for a given set of hyperspherical clusters. For example, in Figure 2, possible KHCS include $\{\mathcal{HC}_1, \mathcal{HC}_2, \mathcal{HC}_{14}\}$, $\{\mathcal{HC}_6, \mathcal{HC}_7, \mathcal{HC}_{14}\}$, $\{\mathcal{HC}_4, \mathcal{HC}_{13}, \mathcal{HC}_{15}\}$.

Since determining a KHCS requires knowledge of the number of subsystems s, s and the KHCS must be identified simultaneously. To this end, hyperspherical clusters are ranked based on their likelihood of being key clusters, using a cost function that evaluates two characteristics: purity and membership in distinct subsystem

The cost function for each hyperspherical cluster $\mathcal{HC}_i$ is defined as:

$$J(\mathcal{HC}_i) = J_1(\mathcal{HC}_i) + J_2(\mathcal{HC}_i), \qquad i = 1,2,\cdots,l \tag{23}$$

where $J_1(\mathcal{HC}_i)$ assesses purity and $J_2(\mathcal{HC}_i)$ assesses distinct subsystem membership.

A. Cost function $J_1(\mathcal{HC}_i)$

This function evaluates purity by measuring how well a linear regression model fits the cluster data. For each $\mathcal{HC}_i$, a linear model:

$$\hat{\breve{y}}(k) = \breve{\underline{\varphi}}^T(k)\breve{\underline{\theta}}_i, \tag{24}$$

is fitted using least squares estimation:

$$\breve{\underline{\theta}}_i = R_i^{-1} f_i, \tag{25}$$

where

$$R_i = \frac{1}{N_{c_i}}\sum_{k=1}^{N_{c_i}} \tilde{\underline{\varphi}}(k)\tilde{\underline{\varphi}}^T(k),\ f_i = \frac{1}{N_{c_i}}\sum_{k=1}^{N_{c_i}} \tilde{\underline{\varphi}}(k)\breve{y}(k). \tag{26}$$

The covariance matrix is $P_i = R_i^{-1}$. Purity is quantified using the R-squared criterion:

$$R^2(\mathcal{HC}_i) = 1 - \frac{\text{SSE}\ (\mathcal{HC}_i)}{\text{SST}(\mathcal{HC}_i)}, \tag{27}$$

where

$$\text{SSE}\ (\mathcal{HC}_i) = \sum_{k=1}^{N_{c_i}}(\breve{y}(k) - \hat{\breve{y}}(k))^2,$$
$$\text{SST}(\mathcal{HC}_i) = \sum_{k=1}^{N_{c_i}}(\breve{y}(k) - \breve{y}_{\text{mean}})^2 \tag{28}$$

and $\breve{y}_{\text{mean}} = \frac{1}{N_{c_i}}\sum_{k=1}^{N_{c_i}} \breve{y}(k)$. The cost function $J_1(\mathcal{HC}_i)$ is then:

$$J_1(\mathcal{HC}_i) = 1 - R^2(\mathcal{HC}_i) \tag{29}$$

For a pure cluster, $R^2(\mathcal{HC}_i) \approx 1$, so $J_1(\mathcal{HC}_i) \approx 0$. Mixed clusters exhibit higher J_1 values due to inconsistent data points.

B. Cost function $J_2(\mathcal{HC}_i)$

This cost function ensures distinct subsystem membership by comparing the parameter vectors of the local linear regression models fitted to each hyperspherical cluster:

$$J_2(\mathcal{HC}_i) = \exp\left(-\delta.\min_{z=1,2,\cdots,\tau-1}\left\|\breve{\underline{\theta}}_i - \breve{\underline{\theta}}_z^{\mathcal{K}}\right\|\right),$$
$$\tau = 2,3,4,\cdots,l, \tag{30}$$

where τ denotes the iteration number in the KHCS identification process, $\breve{\underline{\theta}}_z^{\mathcal{K}}$ is the parameter vector of the linear regression model corresponding to the z-th key cluster identified in previous iterations, and δ is a scaling parameter that controls the sensitivity of the cost function to differences in parameter vectors. For the first iteration $\tau = 1$, $J_2(\mathcal{HC}_i) = 0$ since no previous key clusters exist for comparison. If $\mathcal{HC}_i$ belongs to the same subsystem as an existing key cluster, $J_2(\mathcal{HC}_i) \approx 1$; otherwise, $J_2(\mathcal{HC}_i) \approx 0$.

The scaling parameter $\delta = 16$ is used as a default value in this study, determined through empirical analysis on a range of test systems to provide strong discrimination between different subsystems. This value ensures that J_2 effectively approaches 1 for similar parameter vectors and 0 for distinct ones.

Both J_1 and J_2 range between 0 and 1, requiring no normalization. While J_1 remains constant throughout the process, J_2 is updated iteratively. The KHCS and optimal s are determined by iteratively solving $\min_{i=1,2,\cdots,l} J(\mathcal{HC}_i)$. At each iteration τ, the cluster with the minimal $J(\mathcal{HC}_i)$ is selected as $\mathcal{HC}_\tau^{\mathcal{K}}$, with corresponding parameter vector $\breve{\underline{\theta}}_\tau^{\mathcal{K}}$, covariance matrix $P_\tau^{\mathcal{K}}$, and cost value $J(\tau)$.

For a system with s subsystems, the first s key clusters correspond to distinct subsystems, and $J(1), J(2), \cdots, J(s)$ are relatively low. However, $J(s+1)$ increases sharply because $\mathcal{HC}_{s+1}^{\mathcal{K}}$ either belongs to a duplicate subsystem or is mixed. Therefore, the optimal number of submodels s is the smallest τ satisfying:

$$J(\tau+1) \gg \text{mean}\big(J(1), J(2), \cdots, J(\tau)\big), \tau = 2,3,\cdots,l \tag{31}$$

where $\gg$ denotes " at least 4 times greater". The KHCS is then $\left\{\mathcal{HC}_1^{\mathcal{K}}, \mathcal{HC}_2^{\mathcal{K}}, \ldots, \mathcal{HC}_s^{\mathcal{K}}\right\}$.

3- 2- 3- Determining the Initial Values of the Parameter Vectors and the Primary Labeled Dataset

The parameter vector and covariance matrix corresponding to each key hyperspherical cluster in the KHCS are selected as the initial values for the submodels' parameter vectors and their covariance matrices, respectively. Formally: $\left\{\hat{\underline{\theta}}_z^{(0)}\right\}_{z=1}^{s} = \left\{\breve{\underline{\theta}}_z^{k}\right\}_{z=1}^{s}, \left\{\underline{P}_z^{(0)}\right\}_{z=1}^{s} = \left\{\underline{P}_z^{k}\right\}_{z=1}^{s}$. Finally, for each $i = 1,2,\cdots,s$, the label $\hat{q}(i) = i$ is assigned to the center of $\mathcal{HC}_i^{\mathcal{K}}$. The primary labeled dataset is constructed by gathering these s labeled data points, defined as: $\mathcal{D}_P^{(0)} = \{(\underline{\psi}(k), \hat{q}(k))\}_{k=1}^{s}$, where $\underline{\psi}(k) \triangleq \left[\underline{\varphi}^T(k), y(k)\right]^T$.

The complete procedure for the second algorithm is summarized in Algorithm 2.

3- 2- 4- On the choice of radius *r*

The radius r in Equation (21) significantly influences the performance of the second algorithm. An appropriate selection of r must ensure that at least one pure hyperspherical cluster is obtained for each subsystem.

Generally, a smaller r yields more hyperspherical clusters, including more pure clusters. However, this reduces the number of data points per cluster, particularly in key hyperspherical clusters, which can degrade the accuracy of the local linear models and consequently the initial parameter estimates. Critically, r should not be so small that the number of data points in any cluster falls below the number

of unknown parameters ($m = 2n + 1$), as this leads to an underdetermined estimation problem with infinite solutions. Conversely, a larger r produces fewer clusters but with more data points per pure cluster, improving regression accuracy and initial parameter estimates. However, excessively large values of r may result in predominantly mixed clusters, making it impossible to form a valid KHCS and causing the identification approach to fail. Therefore, r must be carefully balanced. To facilitate appropriate tuning, it is recommended to first normalize the dataset to the interval $[-1,1]$ before performing hyperspherical clustering, and then select r based on the normalized data.

Algorithm 2: The second clustering-based algorithm

Inputs:

Dataset $\mathcal{D}$, radius of hyperspherical clusters (r), optimal order of the submodels (n_{opt})

Outputs:

Optimal number of submodels (s), initial values of the parameter vectors ($\{\hat{\underline{\theta}}_z^{(0)}\}_{z=1}^{s}$), initial values of the covariance matrices ($\{\underline{P}_z^{(0)}\}_{z=1}^{s}$), primary labeled data set ($\mathcal{D}_P^{(0)}$)

Set $j = 0$
repeat
  Let $j = j + 1$
  Select a center $(\underline{\varphi}(k_{c_j}), y(k_{c_j})) \epsilon \mathcal{D}$ randomly, and determine the hyperspherical cluster $\mathcal{HC}_j$ by using (21)
  Update the data set $\mathcal{D}$ as $\mathcal{D} - \mathcal{HC}_j$
until $\mathcal{D} = \emptyset$
Select $l = j$
for $i = 1,2, \dots, l$ **do**
  Calculate R_i, f_i, and $\check{\underline{\theta}}_i$ using (26) and (25), respectively.
  Calculate $P_i = R_i^{-1}$
  Calculate $R^2(\mathcal{HC}_i)$ using (27) and (28)
  Calculate $J_1(\mathcal{HC}_i)$ using (29)
end for
for $\tau = 1,2, \cdots, l$ **do**
  for $\boldsymbol{i} = 1,2, \cdots, l$ **do**
    if $\tau = 1$ then
      $J_2(\mathcal{HC}_i) = 0$
    else
      Calculate $J_2(\mathcal{HC}_i)$ using (30)
    end if
    Calculate $J(\mathcal{HC}_i)$ using (23)
  end for
  Solve $v = \underset{i=1,2,\cdots,l}{\operatorname{argmin}}(J(\mathcal{HC}_i))$
  Let $\mathcal{HC}_\tau^{\mathcal{K}} = \mathcal{HC}_v$, $\check{\underline{\theta}}_\tau^{\mathcal{K}} = \check{\underline{\theta}}_v$, $P_\tau^{\mathcal{K}} = P_v$, and $J(\tau) = J(\mathcal{HC}_v)$
end for
Let $\tau = 1$
repeat
  Let $\tau = \tau + 1$
  Calculate $\text{mean}(J(1), J(2), \cdots, J(\tau))$
until $J(\tau + 1) \gg (\text{mean}(J(1), J(2), \cdots, J(\tau)))$
Select $s = \tau$
Determine the KHCS set as $\{\mathcal{HC}_z^{\mathcal{K}}\}_{z=1}^{s}$
Let $\hat{\underline{\theta}}_1^{(0)} = \check{\underline{\theta}}_1^{\mathcal{K}}, \hat{\underline{\theta}}_2^{(0)} = \check{\underline{\theta}}_2^{\mathcal{K}}, \dots, \hat{\underline{\theta}}_s^{(0)} = \check{\underline{\theta}}_s^{\mathcal{K}}$
Let $P_1^{(0)} = P_1^{\mathcal{K}}, P_2^{(0)} = P_2^{\mathcal{K}}, \cdots, P_s^{(0)} = P_s^{\mathcal{K}}$
Designate labels $\hat{q}(1) = 1, \cdots, \hat{q}(s) = 1$ to the center of $\mathcal{HC}_1^{\mathcal{K}}, \cdots, \mathcal{HC}_s^{\mathcal{K}}$, respectively.
Build $\mathcal{D}_P^{(0)} = \{(\underline{\psi}(k), \hat{q}(k))\}_{k=1}^{s}$ by gathering the labeled data points obtained in step 35.
return s, $\{\hat{\underline{\theta}}_z^{(0)}\}_{z=1}^{s}$,$\{P_z^{(0)}\}_{z=1}^{s}$, $\mathcal{D}_P^{(0)}$

3- 3- Third Algorithm: Self-Labeling SVM-Based Approach

he third algorithm employs a novel Self-Labeling Support Vector Machine (SL-SVM) to determine optimal separating hyperplanes and a recursive estimation method to iteratively refine submodel parameters using initial values $\{\hat{\underline{\theta}}_z^{(0)}\}_{z=1}^{s}$, $\{P_z^{(0)}\}_{z=1}^{s}$, and $\mathcal{D}_P^{(0)}$. The estimated parameter vector and covariance matrix for the z-th submodel at iteration i are denoted as $\hat{\underline{\theta}}_z^{(i)}$ and $P_z^{(i)}$, respectively.

3- 3- 1- The Novel SL-SVM Algorithm for Determining Optimal Separating Hyperplanes (Classifiers)

This work introduces a Self-Labeling SVM algorithm to determine optimal separating hyperplanes $(\underline{w}_j^*)^T\underline{\varphi}+b_j^*=0\ (j=1,2,\dots,p_h)$ and classifiers $\hat{q}(k)=\text{sign}\left((\underline{w}_j^*)^T\underline{\varphi}+b_j^*\right)$, offering superior computational efficiency compared to standard SVM. Computational complexity analysis in Appendix A demonstrates the SL-SVM algorithm's significantly lower complexity.

For systems with two submodels ($p_h=1$), a binary classification problem is addressed. For multi-class problems ($s>2$), a decomposition approach employing multiple independent binary classifiers is utilized.

Standard SVM solutions depend exclusively on support vectors [28], which maintain minimum margins to the separating hyperplane. The SL-SVM algorithm uses these properties through the following margin definition:

Definition 5 (Data Point Margin): For a classifier $\hat{q}(k)=\text{sign}((\underline{w}^*)^T\underline{\varphi}+b^*)$ trained on dataset $\mathcal{D}_P=\{(\underline{\psi}(k),\hat{q}(k))\}_{k=1}^{N_L}$, the margin for a data point $(\underline{\psi}(k),\hat{q}(k))$ is:

$$\Upsilon(\underline{\varphi},\underline{w}^*,b^*)=\left|(\underline{w}^*)^T\underline{\varphi}+b^*\right|/\|\underline{w}^*\|. \tag{32}$$

The SL-SVM optimization problem mirrors standard SVM formulation but incorporates dynamic dataset updates. Let $\mathcal{D}_P^{(i-1)}=\{(\underline{\psi}(k),\hat{q}(k))\}_{k=1}^{N_{L_{i-1}}}$ denote the labeled dataset at iteration i. For each classifier $=1,2,\dots,p_h$, we solve:

$$\begin{cases}\min\limits_{\underline{w}_j^{(i)},b_j^{(i)},\underline{\eta}_j^{(i)}} J_{\text{SLSVM}}(\underline{w}_j^{(i)},b_j^{(i)},\underline{\eta}_j^{(i)})= \\ \quad \min\limits_{\underline{w}_j^{(i)},b_j^{(i)},\underline{\eta}_j^{(i)}} 0.5\left\|\underline{w}_j^{(i)}\right\|^2+C\sum\limits_{k=1}^{N_{L_{i-1}}}\eta_{j,k}^{(i)} \\ s.t: q(k)((\underline{w}_j^{(i)})^T\underline{\varphi}(k)+b_j^{(i)})\geq 1-\eta_{j,k}^{(i)}, \\ \qquad k=1,2,\cdots,N_{L_{i-1}} \\ \eta_{j,k}^{(i)}\geq 0\ ,k=1,2,\cdots,N_{L_{i-1}}\end{cases}, \tag{33}$$

where $\underline{\eta}_j^{(i)}=[\eta_{j,1}^{(i)}\,\eta_{j,2}^{(i)}\cdots\eta_{j,N_{L_{i-1}}}^{(i)}]^T$, and C is the penalty parameter. Through Lagrange multipliers $\underline{\alpha}_j^{(i)}=[\alpha_{j,1}^{(i)}\,\alpha_{j,2}^{(i)}\dots\alpha_{j,N_{L_{i-1}}}^{(i)}]^T$ and $\underline{\mu}_j^{(i)}=[\mu_{j,1}^{(i)}\,\mu_{j,2}^{(i)}\dots\mu_{j,N_{L_{i-1}}}^{(i)}]^T$, we derive the QP formulation:

$$\begin{cases}\min\limits_{\underline{\alpha}_j^{(i)}} 0.5\sum_{k=1}^{N_{L_{i-1}}}\sum_{v=1}^{N_{L_{i-1}}}\alpha_{j,k}^{(i)}\alpha_{j,v}^{(i)}q(k)q(v)\underline{\varphi}^T(k)\underline{\varphi}(v)- \\ \qquad \sum_{k=1}^{N_{L_{i-1}}}\alpha_{j,k}^{(i)} \\ s.t.:\sum_{k=1}^{N_{L_{i-1}}}\alpha_{j,k}^{(i)}q(k)=0 \\ 0\leq\alpha_{j,k}^{(i)}\leq C, k=1,2,\cdots,N_{L_{i-1}}\end{cases} \tag{34}$$

The optimal weight vector is obtained as:

$$\underline{w}_j^{*(i)}=\sum_{k=1}^{N_{L_{i-1}}}\alpha_{j,k}^{*(i)}\,q(k)\underline{\varphi}(k). \tag{35}$$

Support vectors for the j-th classifier are identified as:

$$\mathcal{D}_{SV}^{j}=\left\{\underline{\varphi}(k)\in\mathcal{D}_P^{(i-1)}\middle|0<\alpha_{j,k}^{(i)}<C,k=1,\cdots,N_{L_{i-1}}\right\}. \tag{36}$$

The total support vector set combines all classifiers' support vectors:

$$\mathcal{D}_{TSV}^{(i)}=\cup_{j=1}^{p_h}\mathcal{D}_{SV}^{j}. \tag{37}$$

The bias term is computed via KKT conditions:

$$b_j^{*(i)}=\frac{1}{|\mathcal{D}_{SV}^{j}|}\sum_{\underline{\varphi}(k)\in\mathcal{D}_{SV}^{j}}[q(k)-(\underline{w}_j^{*(i)})^T\underline{\varphi}(k)]. \tag{38}$$

The labeled dataset update strategy achieves two objectives:

Retain support vectors while removing non-support vectors.

Identify and incorporate high-probability support vector (HPSV) candidates.

HPSV candidates are selected through margin minimization:

$$\underline{\psi}_{j,i}^{HPSV}=[\underline{\varphi}_{j,i}^{HPSV}\quad y_{j,i}^{HPSV}]^T=\underset{\underline{\psi}\in\mathcal{D}\setminus\mathcal{D}_P^{(i-1)}}{\text{argmin}}\Upsilon\left(\underline{\varphi},\underline{w}_j^{*(i)},b_j^{*(i)}\right). \tag{39}$$

Their labels are assigned via nearest submodel:

$$\begin{cases}z_j=\underset{z\in\{1,2,\cdots,s\}}{\text{argmin}}\left\|y_{j,i}^{HPSV}-(\underline{\varphi}_{j,i}^{HPSV})^T\hat{\underline{\theta}}_z^{(i-1)}\right\| \\ \hat{q}_{j,i}^{HPSV}=z_j\end{cases}. \tag{40}$$

The HPSV set $\mathcal{D}_{HPSV}^{(i)} = \{(\underline{\psi}_{j,i}^{HPSV}, \hat{q}_{j,i}^{HPSV})\}_{j=1}^{p_h}$ updates the labeled dataset as $\mathcal{D}_P^{(i)} = \mathcal{D}_{TSV}^{(i)} \cup \mathcal{D}_{HPSV}^{(i)}$.

Convergence is achieved when:

$$\max_{j\in\{1,2,\cdots,p_h\}} \left\| \left[\left(\underline{w}_j^{*(i)}\right)^T, b_j^{*(i)} \right]^T - \left[\left(\underline{w}_j^{*(i-1)}\right)^T, b_j^{*(i-1)} \right]^T \right\| < \varepsilon_1, \quad (41)$$

which ε_1 is a constant coefficient. Appendix B proves the algorithm's monotonic convergence. Final hyperplane parameters $\{(\underline{w}_j^*, b_j^*)\}_{j=1}^{p_h}$ form the hyperplane matrix H_i via Equation (5), yielding polyhedral partitions through Equation (4).

3- 3- 2- Iterative Parameter Estimation Algorithm

Submodel parameters are iteratively refined using the data points in $\mathcal{D}_{HPSV}^{(i)}$ through a recursive least squares scheme, starting from initial estimates $\{\underline{\hat{\theta}}_z^{(0)}\}_{z=1}^s$ and $\{P_z^{(0)}\}_{z=1}^s$:

$$\begin{cases} \underline{\hat{\theta}}_{z_j}^{(i)} = \underline{\hat{\theta}}_{z_j}^{(i-1)} + P_{z_j}^{(i)} \underline{\varphi}_{j,i}^{HPSV} \left(y_{j,i}^{HPSV} - (\underline{\varphi}_{j,i}^{HPSV})^T \underline{\hat{\theta}}_{z_j}^{(i-1)} \right) \\ P_{z_j}^{(i)} = P_{z_j}^{(i-1)} - \dfrac{P_{z_j}^{(i-1)} \underline{\varphi}_{j,i}^{HPSV} (\underline{\varphi}_{j,i}^{HPSV})^T P_{z_j}^{(i-1)}}{1 + (\underline{\varphi}_{j,i}^{HPSV})^T P_{z_j}^{(i-1)} \underline{\varphi}_{j,i}^{HPSV}} \end{cases}. \quad (42)$$

The convergence is achieved when:.

$$\max_{z\in\{1,2,\cdots,s\}} \left\| \underline{\hat{\theta}}_z^{(i)} - \underline{\hat{\theta}}_z^{(i-1)} \right\| < \varepsilon_2, \quad (43)$$

where ε_2 is a constant coefficient with a small value. Appendix C establishes asymptotic convergence to true parameters. Algorithm 3 summarizes the complete procedure.

4- Simulation Results and Discussion

This section evaluates the performance and efficiency of the proposed identification approach for offline identification of both PWARX hybrid systems and nonlinear systems represented by PWARX models. All simulations are conducted using MATLAB R2021a on a computer with an Intel Core i5 3.10 GHz processor and 8 GB of RAM.

The Best Fit Rate (BFR) criterion [29] is employed to validate the identified PWARX models across all examples. The BFR is defined as:

$$\text{BFR} = \max\left\{0, 1 - \frac{\left\|\underline{y} - \underline{\hat{y}}\right\|}{\left\|\underline{y} - \text{mean}(\underline{y}).\underline{1}_N\right\|}\right\} \times 100\%, \quad \text{mean}(\underline{y}) = \frac{1}{N}\sum_{k=1}^{N} y(k), \quad (44)$$

where $\underline{y} \in \mathbb{R}^N$ is the vector of actual measured outputs, $\underline{\hat{y}} \in \mathbb{R}^N$ is the vector of model outputs, and $\underline{1}_N = [1,1,\cdots,1]^T \in \mathbb{R}^N$ is a vector of ones.

Example 1. Consider the following PWARX system [20]:

$$y(k) = \begin{cases} \underline{\tilde{\varphi}}^T(k)\underline{\theta}_1 + e(k) & \text{if } \underline{\varphi}(k) \in \Phi_1 \\ \underline{\tilde{\varphi}}^T(k)\underline{\theta}_2 + e(k) & \text{if } \underline{\varphi}(k) \in \Phi_2 \end{cases}, \quad (45)$$

which consists of two first-order subsystems, where $\underline{\tilde{\varphi}}(k) = [y(k-1), u(k-1), 1]^T$, $\underline{\theta}_1 = [0.5, 0.2, 0.1]^T$, $\underline{\theta}_2 = [0.6, 0.1, 0.3]^T$. The polyhedral partitions are defined as:

$$\Phi_1 = \{\underline{\varphi}(k) \in \mathbb{R}^2 | [0 \quad 1 \quad 0]\underline{\tilde{\varphi}}(k) < 0\},$$
$$\Phi_2 = \{\underline{\varphi}(k) \in \mathbb{R}^2 | [0 \quad -1 \quad 0]\underline{\tilde{\varphi}}(k) \le 0\}. \quad (46)$$

For identification purposes, an input signal with uniform distribution in $[-0.5, 0.5]$ is generated and applied to the system. A dataset $\mathcal{D}$ of $N = 1000$ data points is collected. While [20] assumes zero noise, we introduce Gaussian noise $e(k)$ in the interval $[-0.04, 0.04]$ with zero mean and variance 1×10^{-4}. This noise level is reasonable given that the Signal-to-Noise Ratio (SNR) of 25 dB.

Figure 3 displays the collected data points in the regression space. Implementing the first algorithm with $n_{\max} = 10$ and $\rho = 10^{-2}$ yields the optimal submodel order $n_{opt} = 1$. Figure 4 shows the ESR values for orders $n = 0,1,\cdots,10$. The second algorithm with radius $r = 0.2$ generates 16 hyperspherical clusters, shown in Figure 5, and determines the optimal number of submodels as $s = 2$. The resulting KHCS comprises two key hyperspherical clusters, illustrated in Figure 6. The initial parameter vectors are estimated as: $\underline{\hat{\theta}}_1^{(0)} = [0.4951, 0.1990, 0.1019]^T$, $\underline{\hat{\theta}}_2^{(0)} = [0.5812, 0.0959, 0.3079]^T$.

Executing the third algorithm with parameters $C = 1000$, $\varepsilon_1 = 10^{-5}$, and $\varepsilon_2 = 10^{-3}$ converges after 27 iterations ($i_{\text{final}} = 27$), yielding the optimal separating hyperplane with parameters $\underline{w}^* = [-0.0032, 1]^T$ and $b^* = 0.0007$. The final submodel parameters are: $\underline{\hat{\theta}}_1^* = [0.5047, 0.2009, 0.0983]^T$ and $\underline{\hat{\theta}}_2^* = [0.6037, 0.1066, 0.2969]^T$.

Figures 7 and 8 demonstrate the convergence of the classifier parameters and submodel parameters, respectively. Figure 9 displays the optimal separating hyperplane along with the data points in the regression space. The identified PWARX model achieves a BFR value of 98.2493%. Figure 10 compares the actual output, estimated output, and error for a validation dataset of $N = 200$ samples.

Example 2. Consider a PWARX system with three subsystems as follows:

$$y(k) = \begin{cases} \underline{\tilde{\varphi}}^T(k)\underline{\theta}_1 + e(k) & \text{if } \underline{\varphi}(k) \in \Phi_1 \\ \underline{\tilde{\varphi}}^T(k)\underline{\theta}_2 + e(k) & \text{if } \underline{\varphi}(k) \in \Phi_2 \\ \underline{\tilde{\varphi}}^T(k)\underline{\theta}_3 + e(k) & \text{if } \underline{\varphi}(k) \in \Phi_3 \end{cases}, \quad (47)$$

Algorithm 3: The SL-SVM-based algorithm

Inputs:

Number of submodels (s), dataset $\mathcal{D}$, initial values of the parameter vectors ($\{\underline{\hat{\theta}}_z^0\}_{z=1}^s$) and covariance matrices ($\{P_z^0\}_{z=1}^s$), primary labeled data set ($\mathcal{D}_P^0$), coefficients $\varepsilon_1, \varepsilon_2$ and C

Outputs:

Parameters of the optimal separating hyperplanes ($\{(\underline{w}_j^*, b_j^*)\}_{j=1}^{p_h}$), polyhedron partitions ($\{\Phi_i\}_{i=1}^s$), optimal parameter vectors ($\{\underline{\hat{\theta}}_z^*\}_{z=1}^s$)

Set $i = 0$
repeat
  Let $i = i + 1$
  for $j = 1,2,\cdots,p_h$ **do**
    solve the QP optimization problem in (34) for the j-th classifier with the labeled data set $\mathcal{D}_P^{i-1}$, and determine $\underline{\alpha}_j^{i^*}$.
    Calculate the parameters of the j-th classifier ($\underline{w}_j^{i^*}, b_j^{i^*}$) by using (35) and (38).
    Obtain the support vector set corresponding to the j-th optimal separating hyperplane, $\mathcal{D}_{SV}^j$, using (36)
  end for
  Let $\mathcal{D}_{TSV}^i = \bigcup_{j=1}^{p_h} \mathcal{D}_{SV}^j$
  for $j = 1,2,\ldots,p_h$ **do**
    Calculate $\underline{\psi}_{j,i}^{HPSV}$ by solving the optimization problem in (39)
    Calculate z_j and $\hat{q}_{j,i}^{HPSV}$ according to (40)
    Calculate $\underline{\hat{\theta}}_{z_j}^i$ and $P_{z_j}^i$ according to (42)
  end for
  Let $\mathcal{D}_{HPSV}^i = \{(\underline{\psi}_{j,i}^{HPSV}, \hat{q}_{j,i}^{HPSV})\}_{j=1}^{p_h}$
  Let $\mathcal{D}_P^i = \mathcal{D}_{TSV}^i \cup \mathcal{D}_{HPSV}^i$
  until $\max_{j\in\{1,2,\ldots,p_h\}} (\left\| [\underline{w}_j^{i^*} \;\; b_j^{i^*}]^T - [\underline{w}_j^{i-1^*} \;\; b_j^{i-1^*}]^T \right\|) < \varepsilon_1$
  Let $i_{\text{final}} = i$, $\{(\underline{w}_j^*, b_j^*)\}_{j=1}^{p_h} = \{(\underline{w}_j^{i_{\text{final}}}, b_j^{i_{\text{final}}})\}_{j=1}^{p_h}$, $\mathcal{D}_P' = \mathcal{D}_P^{i_{\text{final}}}$
  Find the optimal separating hyperplanes in the form of $\{(\underline{w}_j^*)^T \underline{\varphi} + b_j^* = 0\}_{j=1}^{p_h}$.
  Obtain the hyperplane matrices $\{H_i\}_{i=1}^s$ using (5).
  Determine the polyhedral partitions $\{\Phi_i\}_{i=1}^s$ using (4).
  Let $i = 0, j = 1$
  repeat
    Let $i = i + 1$
    Select randomly a data point from $\mathcal{D} - \mathcal{D}_P'$ and define it as $\underline{\psi}_i'$.
    Estimate the label of $\underline{\psi}_i'$ using the obtained classifiers, $\hat{q}(i) \in \{1,2,\cdots,s\}$.
    Let $z_j = \hat{q}(i)$, and repeat step 13.
    Let $\mathcal{D}_P' = \mathcal{D}_P' \cup \{(\underline{\psi}_i', \hat{q}_i)\}$
    Let $j = j + 1$
  until $\max_{z\in\{1,2,\cdots,s\}} (||\underline{\hat{\theta}}_z^i - \underline{\hat{\theta}}_z^{i-1}||) < \varepsilon_2$
  Let $\{\underline{\hat{\theta}}_z^*\}_{z=1}^s = \{\underline{\hat{\theta}}_z^i\}_{z=1}^s$
return $\{\underline{\hat{\theta}}_z^*\}_{z=1}^s$, $\{(\underline{w}_j^*, b_j^*)\}_{j=1}^{p_h}$, $\{\Phi_i\}_{i=1}^s$

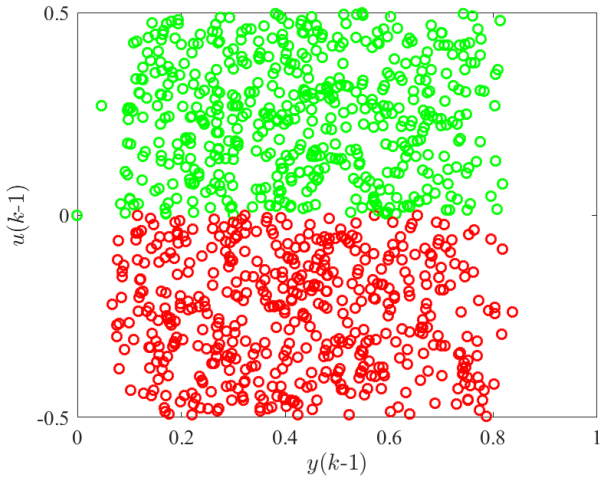

Fig. 3. A view of the collected data points for Example 1 in the regression space.

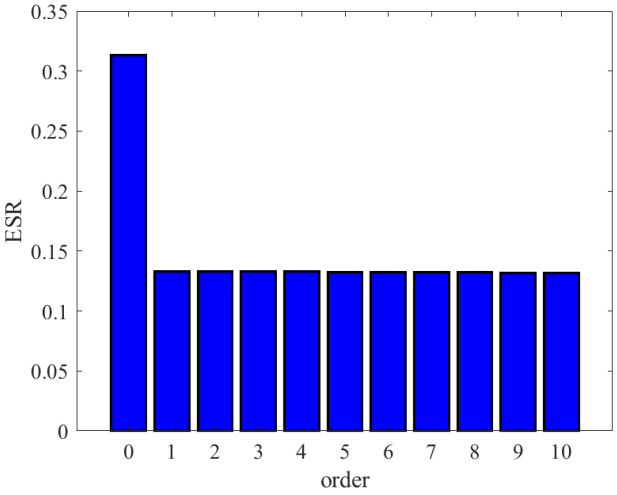

Fig. 4.The ESR values obtained for orders n=0,1,...,10 in Example 1.

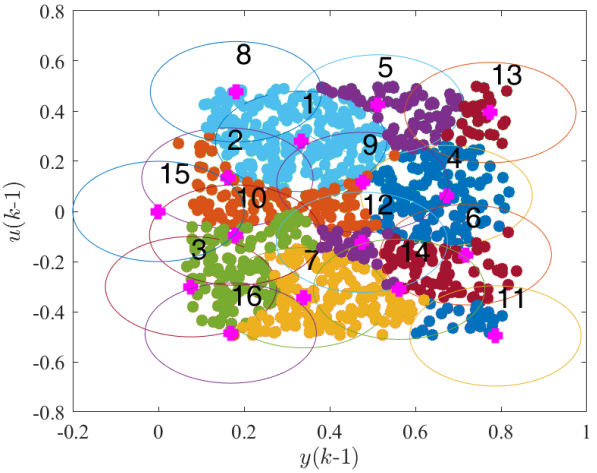

Fig. 5. Obtained hyperspherical clusters.

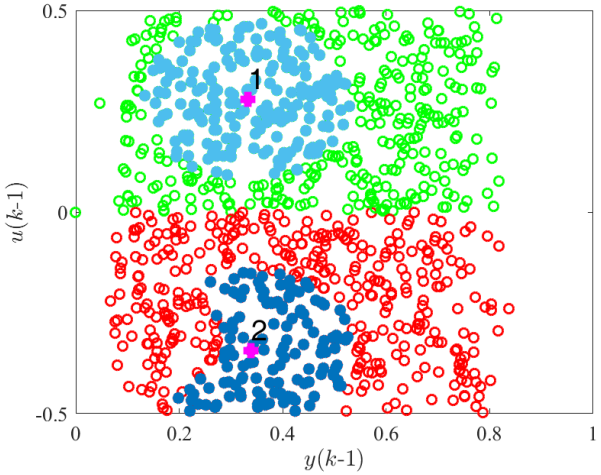

Fig. 6. A view of the obtained two key hyperspherical clusters for Example 1.

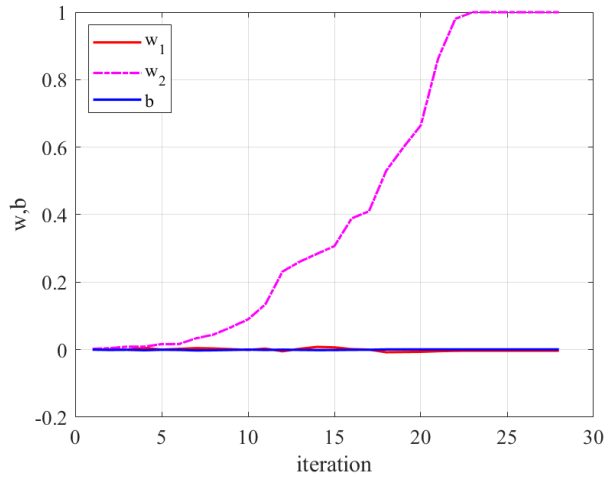

Fig. 7. The classifier's parameters in terms of the iteration.

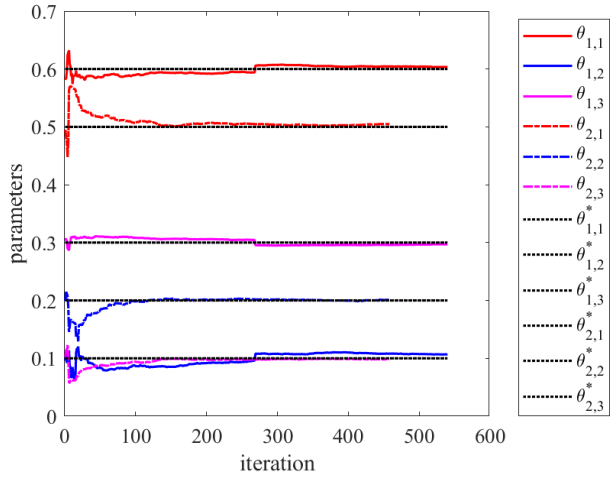

Fig. 8. The submodels' parameters in terms of the iteration.

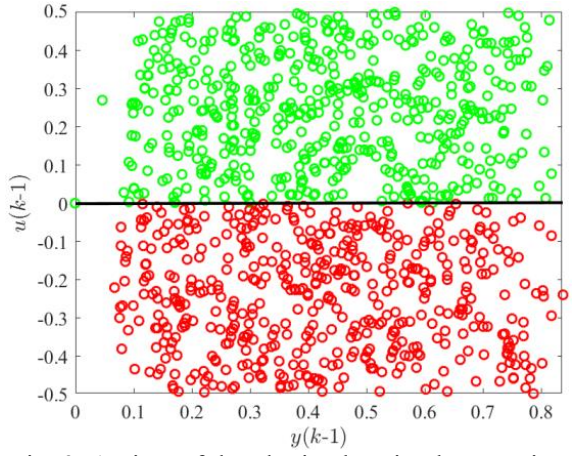

Fig. 9. A view of the obtained optimal separating hyperplane in the regression space.

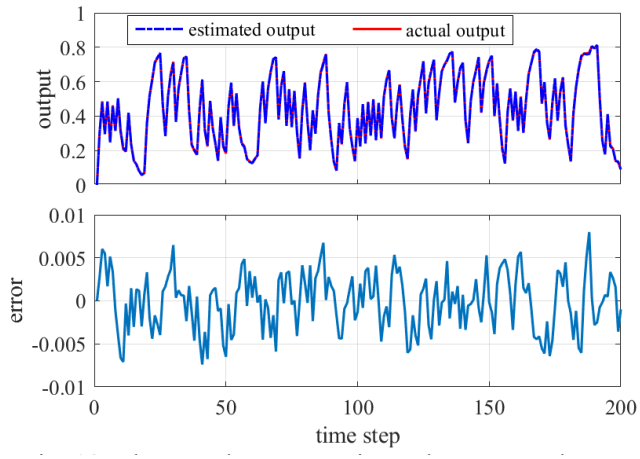

Fig. 10.The actual output, estimated output, and error for a validation data set in Example 1.

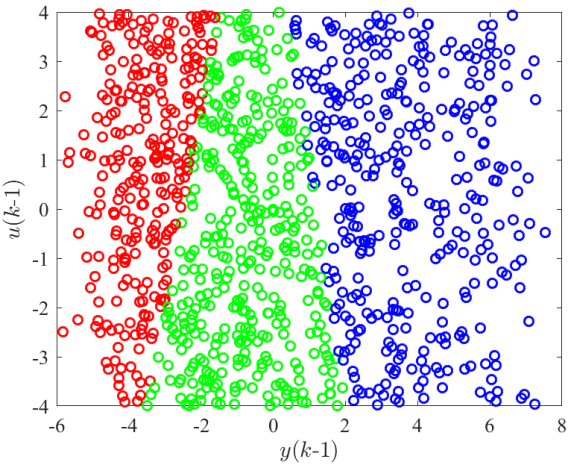

Fig. 11. A view of the collected data points for Example 2 in the regression space.

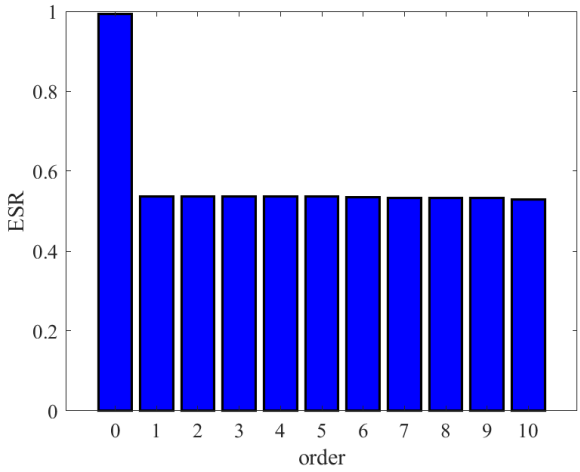

Fig. 12. The ESR value in terms of the order for Example 2.

where

$$\underline{\tilde{\varphi}}(k) = [y(k-1), u(k-1), 1]^T,$$

$$\underline{\theta}_1 = [-0.4, 1, 1.5]^T, \underline{\theta}_2 = [0.5, -1, -0.5]^T,$$

$$\underline{\theta}_3 = [-0.3, 0.5, -1.7]^T,$$

$$\Phi_1 = \{\underline{\varphi}(k) \in \mathbb{R}^2 | [4 \quad -1 \quad 10]\underline{\tilde{\varphi}}(k) < 0\}, \Phi_2 = \{\underline{\varphi}(k) \in \mathbb{R}^2 | \begin{bmatrix} -4 & 1 & -10 \\ 5 & 1 & -6 \end{bmatrix} \underline{\tilde{\varphi}}(k) \leq 0\},$$

$$\Phi_3 = \{\underline{\varphi}(k) \in \mathbb{R}^2 | [-5 \quad -1 \quad 6]\underline{\tilde{\varphi}}(k) < 0\}.$$

The PWARX system in (47) has been widely used in previous studies [17, 20, 23, 30, 31] as a benchmark for evaluating identification methods. Following [17], we generate an input signal with uniform distribution in $[-4,4]$ and collect $N = 1000$ data points. The noise signal $e(k)$ follows a uniform distribution in $[-0.2,0.2]$. Figure 11 shows the collected data points in the regression space.

Applying the first algorithm with $\rho = 10^{-2}$ yields the optimal submodel order $n_{opt} = 1$. Figure 12 displays the ESR values for orders $n = 0,1,\cdots,10$. The second algorithm with radius $r = 1.6$ generates 29 hyperspherical clusters, shown in Figure 13. Using criterion (31), the optimal number of submodels is estimated as three. The three key hyperspherical clusters are illustrated in Figure 14 along with the data points.

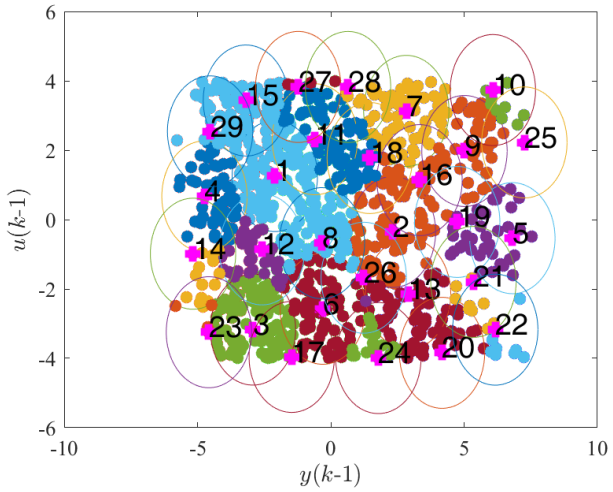

Fig. 13. Obtained hyperspherical clusters for Example 2.

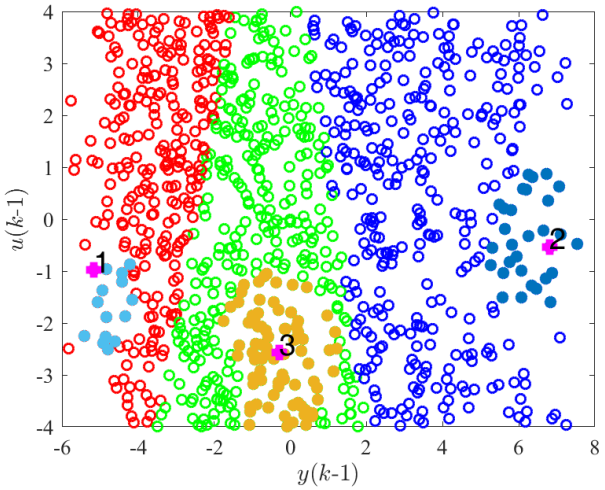

Fig. 14. A view of the obtained three key hyperspherical clusters.

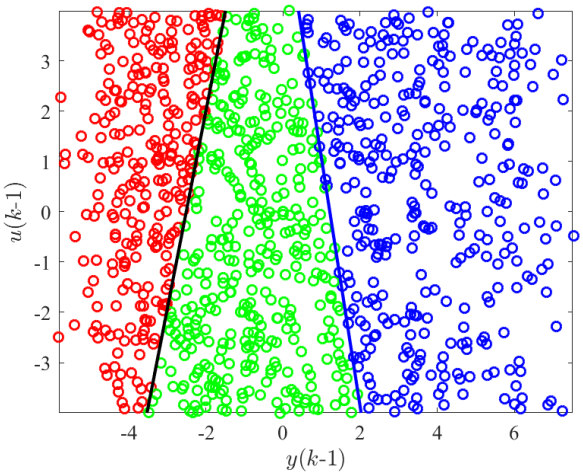

Fig. 15. A view of the obtained optimal separating hyperplanes for Example 2.

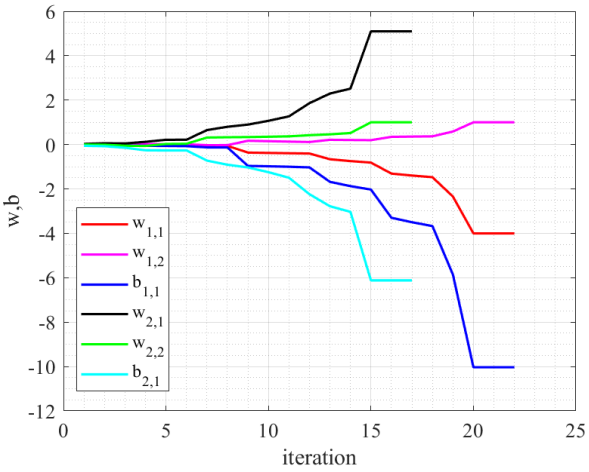

Fig. 16. Parameters of the classifiers versus the iteration.

The initial parameter vectors are obtained as:

$$\underline{\hat{\theta}}_1^{(0)} = [-0.3299, 1.0241, 1.6513]^T, \underline{\hat{\theta}}_2^{(0)} = [0.4806, -1.0061, -0.5045]^T,$$
$$\underline{\hat{\theta}}_3^{(0)} = [-0.3195, 0.4815, -1.5944]^T.$$

Executing the third algorithm with parameters $C = 1000$, $\varepsilon_1 = 10^{-5}$, and $\varepsilon_2 = 10^{-3}$ yields the first and second optimal separating hyperplanes after 22 and 17 iterations, respectively:

$$(\underline{w}_1^*)^T \underline{\varphi} + b_1^* = 0; \underline{w}_1^* = [\underline{w}_{1,1}^* \ \underline{w}_{1,2}^*]^T = [-4.0038, 1]^T, b_1^* = -10.0312$$
$$(\underline{w}_2^*)^T \underline{\varphi} + b_2^* = 0; \underline{w}_2^* = [\underline{w}_{2,1}^* \ \underline{w}_{2,2}^*]^T = [5.0997, 1]^T, b_2^* = -6.1206$$

Figure 15 shows the optimal separating hyperplanes along with the data points in the regression space. The final estimated parameter vectors are:

$$\underline{\hat{\theta}}_1^* = [-0.4020, 1.0075, 1.4910]^T,$$
$$\underline{\hat{\theta}}_2^* = [0.4915, -0.9928, -0.5040]^T,$$
$$\underline{\hat{\theta}}_3^* = [-0.3025, 0.5063, -1.6905]^T.$$

Figures 16 and 17 show the convergence of the classifier parameters and submodel parameters, respectively. For a validation dataset with $N = 200$ samples, the identified PWARX model achieves a BFR value of 97.025%. Figure 18 compares the actual output, estimated output, and error.

Table 2 compares the results of the proposed approach with prior studies for the system in (47). The results demonstrate

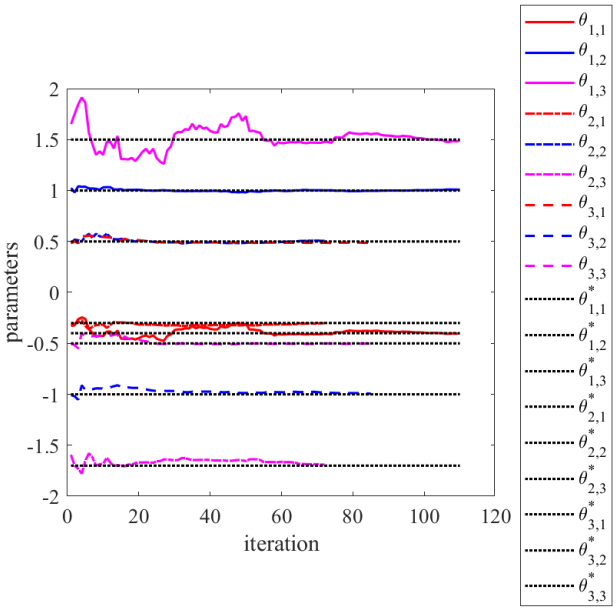

Fig. 17. Submodels' parameters versus the iteration.

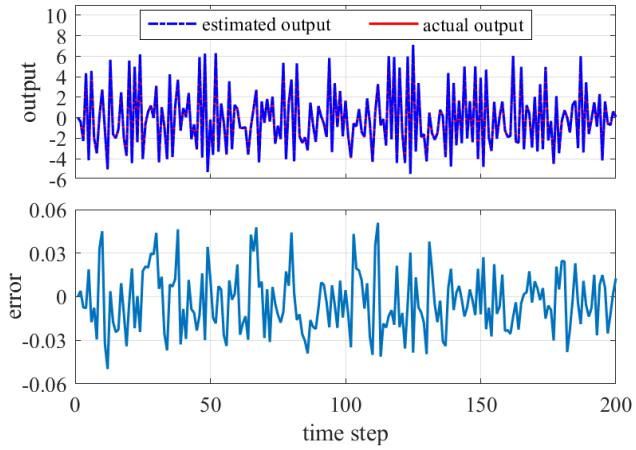

Fig. 18. The actual output, estimated output, and error for a validation data set.

Table 2. Comparison of proposed approach and previous results for Example 2.

Approaches	parameters of submodels	parameters of hyperplanes	BFR value
Actual values	$\underline{\theta}_1 = [-0.4, 1, 1.5]^T$ $\underline{\theta}_2 = [0.5, -1, -0.5]^T$ $\hat{\underline{\theta}}_3^* = [-0.3, 0.5, -1.7]^T$	$\underline{w}_1 = [4, -1]^T, b_1 = 10$ $\underline{w}_2 = [5, 1]^T, b_2 = -6$	
Proposed approach	$\hat{\underline{\theta}}_1^* = [-0.4020, 1.0075, 1.4910]^T$ $\hat{\underline{\theta}}_2^* = [0.4915, -0.9928, -0.5040]^T$ $\hat{\underline{\theta}}_3^* = [-0.3025, 0.5063, -1.6905]^T$	$\underline{w}_1^* = [4.004, -1]^T, b_1^* = 10.031$ $\underline{w}_2^* = [5.01, 1]^T, b_1^* = -6.121$	97.025
Approach in [17]	$\hat{\underline{\theta}}_1^* = [-0.3961, 0.9903, 1.5472]^T$ $\hat{\underline{\theta}}_2^* = [0.5018, -0.9980, -0.4994]^T$ $\hat{\underline{\theta}}_3^* = [-0.2989, 0.5045, -1.7072]^T$	$\underline{w}_1^* = [3.9591, -0.9665]^T, b_1^* = 10.01$ $\underline{w}_2^* = [5.0513, 1.1876]^T, b_2^* = -5.92$	96.17
Approach in [23]	$\hat{\underline{\theta}}_1^* = [-0.3994, 1.0010, 1.5147]^T$ $\hat{\underline{\theta}}_2^* = [0.5025, -0.9991, -0.4858]^T$ $\hat{\underline{\theta}}_3^* = [-0.2992, 0.4979, -1.7126]^T$	$\underline{w}_1^* = [4.279, -1.062]^T, b_1^* = 10$ $\underline{w}_2^* = [4.9782, 0.987]^T, b_2^* = -6$	95.23
Approach in [20]	$\hat{\underline{\theta}}_1^* = [-0.389, 0.9711, 1.5801]^T$ $\hat{\underline{\theta}}_2^* = [0.5036, -1.0102, -0.4766]^T$ $\hat{\underline{\theta}}_3^* = [-0.2816, 0.4945, -1.7084]^T$	$\underline{w}_1^* = [4.1124, -0.9925]^T, b_1^* = 10.47$ $\underline{w}_2^* = [4.915, 1.099]^T, b_2^* = -5.95$	95.29
Approach in [30]	$\hat{\underline{\theta}}_1^* = [-0.3921, 0.9978, 1.5426]^T$ $\hat{\underline{\theta}}_2^* = [0.4980, -0.9994, -0.4971]^T$ $\hat{\underline{\theta}}_3^* = [-0.30, 0.5005, -1.7011]^T$	$\underline{w}_1^* = [4.0036, -0.9854]^T, b_1^* = 9.5903$ $\underline{w}_2^* = [5.0002, 0.9990]^T, b_2^* = -6.2009$	95.67
Approach in [31]	$\hat{\underline{\theta}}_1^* = [-0.3902, 0.9586, 1.6511]^T$ $\hat{\underline{\theta}}_2^* = [0.3467, -0.8094, -0.2953]^T$ $\hat{\underline{\theta}}_3^* = [-0.2102, 0.4425, -2.0834]^T$	unknown	91.43

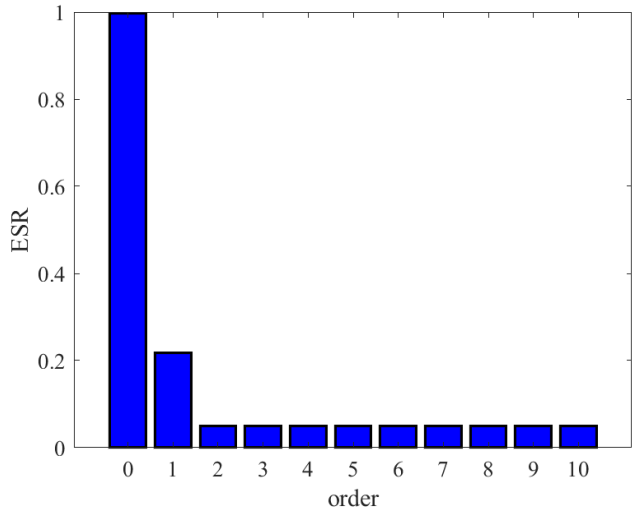

Fig. 19. ESR values in terms of the order for Example 3.

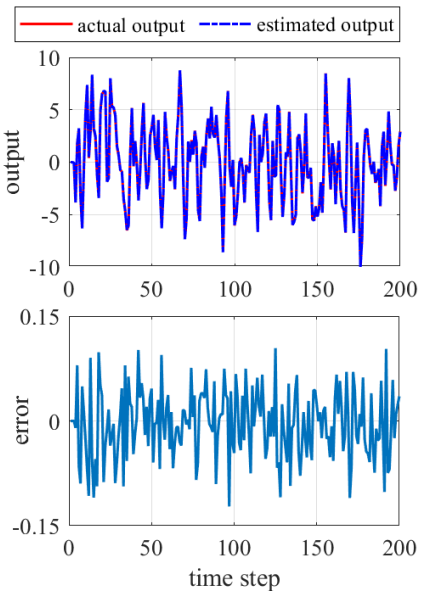

Fig. 20. The actual output, estimated output, and errors for a validation data set with N=200.

that our method achieves the highest BFR value among all compared approaches.

Example 3. This example investigates the capability of the proposed identification approach for offline identification of nonlinear systems via the PWARX model. Consider the following nonlinear system [32-34]:

$$y(k)=\frac{1.5y(k-1)y(k-2)}{1+y^2(k-1)+y^2(k-2)}+\sin(y(k-1)+y(k-2))+u(k-1)+0.8u(k-2)+e(k) \quad (48)$$

To generate the identification dataset, a random input signal with uniform distribution in $[-5,5]$ is applied to the system. The noise signal $e(k)$ follows a uniform distribution in $[-0.1,0.1]$.

Using the collected dataset with $N=1000$ samples, the first algorithm determines the optimal submodel order as $n_{opt}=2$. Figure 19 shows the ESR values versus the candidate model orders.

Implementing the second and third algorithms with $n_{opt}=2$ yields an optimal number of submodels $s=16$. The resulting PWARX model achieves a BFR value of 97.6954%, demonstrating its high accuracy. Figure 20 compares the model output with the actual output for a validation dataset of $N=200$ samples.

4- 1- Discussion on Practical Considerations and Limitations

The proposed framework is designed for black-box identification where no prior structural knowledge is available. To assess its practical applicability, key considerations are discussed below.

Robustness to Noise: The algorithm's performance under noise is validated in Examples 1 and 2, where significant Gaussian and uniformly distributed noise was added. The high Best Fit Rates (BFR > 97%) obtained demonstrate effective noise rejection. This robustness stems from two core mechanisms: (1) the ESR test for order selection, which is designed to find a plateau in the presence of noise rather than an exact zero, and (2) the clustering stage, which fits local linear models to groups of points, inherently averaging out local noise effects. Performance may degrade only under extreme noise-to-signal ratios that corrupt the initial clustering purity, a common challenge for data-driven methods.

Computational Load and Scalability: As analyzed in Appendix A, the novel SL-SVM algorithm has significantly lower complexity than standard SVM. The primary computational cost arises from the initial hyperspherical clustering (Algorithm 2). This step scales with the number of data points N and the chosen radius r. For very large datasets (e.g., $N>10^5$), this step could become a bottleneck. Future work may investigate efficient clustering variants (e.g., using spatial indexing) for such scenarios. The algorithm's complexity with respect to the regression space dimension n_φ is linear in the clustering and parameter estimation steps, making it suitable for moderate-dimensional systems common in practical applications like those cited in Section 1 [3-6].

Parameter Sensitivity and Guidelines: The method has two main parameters: the clustering radius r and the SVM penalty parameter C. As detailed in Section 3-2-4, selecting an appropriate r is crucial. The practical guideline of normalizing the data to [−1, 1] and then choosing r (e.g., between 0.1 and 0.3 for normalized data) provides a robust

starting point. The parameter C, common to all SVM-based methods, was kept at a constant high value ($C = 1000$) across all examples with consistent success, indicating low sensitivity for the typical separation tasks encountered in PWARX identification.

General Applicability and Limitations: The framework's strength is in complete structure discovery for systems that admit a PWARX representation. The benchmark systems used for validation are standard in the hybrid systems literature [17, 20, 23, 31], allowing for direct comparison with the state-of-the-art. The method is most suitable for offline identification, as employed here. However, the recursive parameter update (Eq. 42) provides a clear pathway for online adaptation or fine-tuning, which is a promising direction for future work, particularly in control applications for uncertain nonlinear systems [35]. A primary limitation is its dependence on the existence of sufficiently pure local clusters in the regression space; highly overlapping subsystems or excessively large noise may challenge the initial clustering stage.

5- Conclusion

This paper has introduced a fully automated framework for complete structure discovery and parameter estimation of PWARX systems. The proposed methodology fundamentally advances the state-of-the-art by simultaneously determining the number of submodels, their orders, parameter vectors, and polyhedral partitions without requiring any prior structural knowledge. The core contribution lies in the integrated three-stage approach that systematically addresses the complete identification problem. The order selection algorithm combining orthogonal least squares with error-to-signal ratio testing provides a principled method for determining submodel complexity. The clustering-based structure identification algorithm effectively estimates the number of submodels while generating robust initial parameter estimates. Most significantly, the novel Self-Labeling SVM algorithm enables efficient partition estimation with guaranteed convergence properties, overcoming computational limitations of standard SVM approaches. Theoretical analyses establish rigorous convergence guarantees for both parameter vectors and hyperplane parameters, while computational complexity analysis demonstrates the framework's efficiency. Extensive simulations validate the approach's superiority, showing higher accuracy compared to existing methods across various system configurations. Furthermore, the framework's effectiveness in nonlinear system identification through PWARX approximation underscores its practical versatility. The proposed method not only provides a comprehensive solution to the complete PWARX identification problem but also establishes a foundation for future research in adaptive control and real-time system identification, where the recursive estimation capabilities can be leveraged for online applications.

References

[1] Y. Huo, Z. Chen, Q. Li, Q. Li, and M. Yin, "Machine Learning Based Model Predictive Control with Piecewise-Affine Approximation Structure for Maximizing Wind Energy Capture," Journal of Modern Power Systems and Clean Energy, 2025.

[2] J. Tian, S. Wang, X. Zhao, Y. Li, Z. Zheng, and H. Zhang, "Integrated Control Strategy of Active Front-Wheel Steering and Active Suspension Based on 3D Piecewise Affine Tire Model," IEEE Transactions on Transportation Electrification, 2025.

[3] J. Guo and Z. Ren, "On Prediction of Air Pollution Using Piecewise Affine Models," Polish Journal of Environmental Studies, vol. 34, no. 1, 2025.

[4] Z. Ren, J. Zhang, Y. Zhou, and X. Ji, "Prediction of PM2. 5 with a piecewise affine model considering spatial-temporal correlation," Journal of Intelligent & Fuzzy Systems, vol. 46, no. 4, pp. 9525–9542, 2024.

[5] L. Zeineb and A. Kamel, "Extension of a clustering identification approach to multivariable piecewise affine systems: Application to an industrial dryer," Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, vol. 238, no. 6, pp. 1037–1049, 2024.

[6] X. Li, T. Zhao, and Y. Weng, "Research of PWA Modeling and Predictive Control for the MCR-BWPT Systems," IEEE Transactions on Transportation Electrification, 2025.

[7] M. Lotfi, M. B. Menhaj, S. A. Hosseini, and A. S. Shirani, "A design of switching supervisory control based on fuzzy-PID controllers for VVER-1000 pressurizer system with RELAP5 and MATLAB coupling," Annals of Nuclear Energy, vol. 147, p. 107625, 2020.

[8] S. A. Hosseini, A. S. Shirani, M. Lotfi, and M. B. Menhaj, "Design and application of supervisory control based on neural network PID controllers for pressurizer system," Progress in Nuclear Energy, vol. 130, p. 103570, 2020.

[9] N. Moustakis, B. Zhou, T. Le Quang, and S. Baldi, "Fault detection and identification for a class of continuous piecewise affine systems with unknown subsystems and partitions," International Journal of Adaptive Control and Signal Processing, vol. 32, no. 7, pp. 980–993, 2018.

[10] X. Sun, P. Wu, Y. Cai, S. Wang, and L. Chen, "Piecewise affine modeling and hybrid optimal control of intelligent vehicle longitudinal dynamics for velocity regulation," Mechanical Systems and Signal Processing, vol. 162, p. 108089, 2022.

[11] P. Bacher and H. Madsen, "Identifying suitable models for the heat dynamics of buildings," Energy and buildings,

vol. 43, no. 7, pp. 1511–1522, 2011.

[12] M. Mejari, V. V. Naik, D. Piga, and A. Bemporad, "Identification of hybrid and linear parameter-varying models via piecewise affine regression using mixed integer programming," International Journal of Robust and Nonlinear Control, vol. 30, no. 15, pp. 5802–5819, 2020.

[13] X. Wang, G. Chen, Z. Wang, and J. Xia, "Stability Analysis of Piecewise Affine Networked Control Systems under Aperiodic Sampling," IEEE Control Systems Letters, 2025.

[14] A. Ma, D. Li, and Y. Xi, "Event-triggered distributed model predictive control for PWA systems," International Journal of Robust and Nonlinear Control, vol. 35, no. 2, pp. 435–451, 2025.

[15] C. Zhang, Q. Gao, Y. Deng, and J. Qiu, "An integral sliding-mode parallel control approach for general nonlinear systems via piecewise affine linear models," International Journal of Robust and Nonlinear Control, vol. 33, no. 8, pp. 4438–4458, 2023.

[16] S. Xiao, G. Yan, X. Huang, and R. Li, "Adaptive Fault-Tolerant Tracking Control of Piecewise Affine Systems With Performance Constraint," International Journal of Robust and Nonlinear Control, 2025.

[17] A. Bemporad, A. Garulli, S. Paoletti, and A. Vicino, "A bounded-error approach to piecewise affine system identification," IEEE Transactions on Automatic Control, vol. 50, no. 10, pp. 1567–1580, 2005.

[18] W. Bai, F. Guo, L. Chen, K. Hao, and B. Huang, "Variational Bayesian inference for robust identification of PWARX systems with time-varying time-delays," IEEE Transactions on Cybernetics, vol. 53, no. 6, pp. 3613–3623, 2021.

[19] R. Vidal, S. Soatto, Y. Ma, and S. Sastry, "An algebraic geometric approach to the identification of a class of linear hybrid systems," in 42nd IEEE International Conference on Decision and Control (IEEE Cat. No. 03CH37475), 2003, vol. 1: IEEE, pp. 167–172.

[20] Y. Du, F. Liu, J. Qiu, and M. Buss, "A semi-supervised learning approach for identification of piecewise affine systems," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 67, no. 10, pp. 3521–3532, 2020.

[21] M. Lotfi and M. B. Menhaj, "An overview of intelligent systems (neural networks) from the perspective of classical theory and their application in modeling and control of complex systems," Journal of Control, vol. 17, no. 2, pp. 47–79, 2023.

[22] G. Ferrari-Trecate, M. Muselli, D. Liberati, and M. Morari, "A clustering technique for the identification of piecewise affine systems," Automatica, vol. 39, no. 2, pp. 205–217, 2003.

[23] H. Nakada, K. Takaba, and T. Katayama, "Identification of piecewise affine systems based on statistical clustering technique," Automatica, vol. 41, no. 5, pp. 905–913, 2005.

[24] M. Lotfi, M. Habibi, B. Vahidi, S. H. Hosseinian, and M. B. Menhaj, "Physics-aware neural networks for integrated energy systems management," in Physics-Aware Machine Learning for Integrated Energy Systems Management: Elsevier, 2025, pp. 381–420.

[25] L. Zeineb and A. Kamel, "A Modified DBSCAN clustering algorithm for the identification of PWA systems," in 2023 20th International Multi-Conference on Systems, Signals & Devices (SSD), 2023: IEEE, pp. 306–311.

[26] V. Breschi and M. Mejari, "Shrinkage strategies for structure selection and identification of piecewise affine models," in 2020 59th IEEE Conference on Decision and Control (CDC), 2020: IEEE, pp. 1626–1631.

[27] S. Billings, S. Chen, and M. Korenberg, "Identification of MIMO non-linear systems using a forward-regression orthogonal estimator," International journal of control, vol. 49, no. 6, pp. 2157–2189, 1989.

[28] L. Bottou and C.-J. Lin, "Support vector machine solvers," 2007.

[29] L. Wang and H. Garnier, System identification, environmental modelling, and control system design. Springer, 2011.

[30] A. Bemporad, A. Garulli, S. Paoletti, and A. Vicino, "A greedy approach to identification of piecewise affine models," in Hybrid Systems: Computation and Control: 6th International Workshop, HSCC 2003 Prague, Czech Republic, April 3–5, 2003 Proceedings 6, 2003: Springer, pp. 97–112.

[31] A. K. Shah and D. M. Adhyaru, "Parameter identification of PWARX models using fuzzy distance weighted least squares method," Applied Soft Computing, vol. 25, pp. 174–183, 2014.

[32] L. C. YIN, "Identification and control of nonlinear systems using multiple models," 2011.

[33] F.-C. Chen and H. K. Khalil, "Adaptive control of a class of nonlinear discrete-time systems using neural networks," IEEE Transactions on Automatic Control, vol. 40, no. 5, pp. 791–801, 1995.

[34] K. S. Narendra and J. Balakrishnan, "Adaptive control using multiple models," IEEE transactions on automatic control, vol. 42, no. 2, pp. 171–187, 1997.

[35] H. Ahmadian, M. Lotfi, M. B. Menhaj, H. A. Talebi, and I. Sharifi, "A novel L1 adaptive-hybrid control with guaranteed stability for a class of uncertain nonlinear systems: A case study on SA330 Puma," Journal of the Franklin Institute, vol. 359, no. 17, pp. 9860–9885, 2022.

Appendix A. Computational Complexity of the Proposed SL-SVM Algorithm

The computational complexity of the proposed SL-SVM algorithm consists of two main components. The complexity for training the classifier with the labeled dataset $\mathcal{D}_P^{(i-1)}$ at each iteration i is of order $O\left[i_{\text{final}}.\left(|\mathcal{D}_{SV}^{(i-1)}|+|\mathcal{D}_{HPSV}^{(i-1)}|\right)^3\right]=O\left[i_{\text{final}}.\left(|\mathcal{D}_{SV}^{(i-1)}|+1\right)^3\right]$, while the dataset update complexity is $O\left[i_{\text{final}}.(n_\varphi+1)\left(|\mathcal{D}|-|\mathcal{D}_P^{(i-1)}|\right)\right]=O\left[i_{\text{final}}.(n_\varphi+1)\left(N-|\mathcal{D}_P^{(i-1)}|\right)\right]$. In comparison, standard SVM algorithms exhibit $O(N^3)$ complexity [28]. Given that $i_{\text{final}}, \max_i |\mathcal{D}_{SV}^{i-1}| \ll N$ in practical applications, the total complexity of our SL-SVM algorithm satisfies:

$$O\left[i_{\text{final}}.\left(|\mathcal{D}_{SV}^{(i-1)}|+1\right)^3\right]+O\left[i_{\text{final}}.(n_\varphi+1)\left(N-\left|\mathcal{D}_P^{(i-1)}\right|\right)\right]\ll O(N^3). \tag{A.1}$$

This demonstrates the substantial computational advantage of our proposed approach, particularly for large-scale identification problems.

Appendix B. Convergence Analysis of hyperplane Parameters in the Third Algorithm

B-1. Intuitive Overview

The convergence of the SL-SVM classifier parameters is guaranteed because, in each iteration, the algorithm solves an SVM problem on a refined dataset. This dataset retains all support vectors from the previous solution—the only data points that define the optimal hyperplane—and augments them with new high-potential points (HPSVs) closest to the current decision boundary. This ensures the cost function J_{SLSVM} cannot increase. Since the cost is bounded below by the optimal SVM solution on the entire dataset, the sequence of solutions must converge to a stable set of hyperplane parameters $(\underline{w}^*, b^*)$.

B-2. Formal Proof

Without loss of generality, we consider a single classifier case, though the proof extends directly to multiple classifiers. Let $(\underline{w}^{*(1)}, b^{*(1)}, \underline{\eta}^{*(1)})$ denote the solution to the constrained optimization problem (34) at the first iteration $(i=1)$:

$$\begin{cases} \min\limits_{\underline{w}^{(1)},b^{(1)},\underline{\eta}^{(1)}} 0.5||\underline{w}^{(1)}||^2 + C\sum_{k=1}^{|\mathcal{D}_P^{(0)}|}\eta_k^{(1)} \\ s.t.: \hat{q}(k)((\underline{w}^{(1)})^T\underline{\varphi}(k)+b^{(1)}) \geq 1-\eta_k^{(1)}, k=1,2,\cdots,|\mathcal{D}_P^{(0)}|. \\ \eta_k^{(1)} \geq 0, k=1,2,\cdots,|\mathcal{D}_P^{(0)}| \end{cases} \tag{B.1}$$

Since only support vectors influence the optimal separating hyperplane, the solution remains unchanged when non-support vectors are removed. Thus, the solution is equivalent to that of the reduced problem:

$$\begin{cases} \min\limits_{\underline{w}^{(1)},b^{(1)},\underline{\eta}^{(1)}} 0.5||\underline{w}^{(1)}||^2 + C\sum_{k=1}^{|\mathcal{D}_{SV}^{(1)}|}\eta_k^{(1)} \\ s.t.: \\ \hat{q}^{SV}(k)((\underline{w}^{(1)})^T\underline{\varphi}^{SV}(k)+b^{(1)}) \geq 1-\eta_k^{(1)}, k=1,2,\cdots,|\mathcal{D}_{SV}^{(1)}| \\ \eta_k^{(1)} \geq 0, \quad k=1,2,\cdots,|\mathcal{D}_{SV}^{(1)}| \end{cases}, \tag{B.2}$$

where $\mathcal{D}_{SV}^{(1)} = \{\underline{\psi}^{SV}(k), \hat{q}^{SV}(k)\}_{k=1}^{|\mathcal{D}_{SV}^{(1)}|}$ represents the support vector set.

After obtaining the solution, the labeled dataset updates to $\mathcal{D}_P^{(1)} = \mathcal{D}_{SV}^{(1)} \cup \mathcal{D}_{HPSV}^{(1)}$, where $\mathcal{D}_{HPSV}^{(1)} = \{(\underline{\psi}_1^{HPSV}, \hat{q}_1^{HPSV})\}$. In the second iteration, the solution $(\underline{w}^{*(2)}, b^{*(2)}, \underline{\eta}^{*(2)})$ is determined by solving:

$$\begin{cases} \min\limits_{\underline{w}^{(2)}, b^{(2)}, \underline{\eta}^{(2)}} 0.5||\underline{w}^{(2)}||^2 + C \sum\limits_{k=1}^{|\mathcal{D}_P^{(1)}|} \eta_k^{(2)} \\ s.t.: \hat{q}(k)((\underline{w}^{(2)})^T \underline{\varphi}(k) + b^{(2)}) \geq 1 - \eta_k^{(2)}, k = 1,2,\cdots,|\mathcal{D}_P^{(1)}| \\ \eta_k^{(2)} \geq 0, k = 1,2,\cdots,|\mathcal{D}_P^{(1)}| \end{cases}. \tag{B.3}$$

This expands to:

$$\begin{cases} \min\limits_{\underline{w}^{(2)}, b^{(2)}, \underline{\eta}^{(2)}} 0.5||\underline{w}^{(2)}||^2 + C\eta_1^{(2)} + C \sum\limits_{k=1}^{|\mathcal{D}_{SV}^{(1)}|} \eta_{k+1}^{(2)} \\ s.t.: \\ \hat{q}^{HPSV}((\underline{w}^{(2)})^T \underline{\varphi}^{HPSV} + b^{(2)}) \geq 1 - \eta_1^{(2)} \\ \eta_1^2 \geq 0 \\ \hat{q}^{SV}(k)((\underline{w}^{(2)})^T \underline{\varphi}^{SV}(k) + b^{(2)}) \geq 1 - \eta_{k+1}^{(2)}, k = 1,2,\cdots,|\mathcal{D}_{SV}^{(1)}| \\ \eta_{k+1}^{(2)} \geq 0, k = 1,2,\cdots,|\mathcal{D}_{SV}^{(1)}| \end{cases}. \tag{B.4}$$

From the optimization structures in (B.2) and (B.4), we observe that $(\underline{w}^{*(2)}, b^{*(2)}, \underline{\eta}^{*(2)})$ constitutes a feasible solution for problem (B.1). Since $(\underline{w}^{*(1)}, b^{*(1)}, \underline{\eta}^{*(1)})$ is the optimal solution for (B.1), and considering the cost function $J_{\text{SLSVM}}(\underline{w}^{(i)}, b^{(i)}, \underline{\eta}^{(i)}) = 0.5\|\underline{w}^{(i)}\|^2 + C\sum_{k=1}^{|\mathcal{D}_P^{(i-1)}|} \eta_k^{(i)}$, it follows that:

$$J_{\text{SLSVM}}(\underline{w}^{*(1)}, b^{*(1)}, \underline{\eta}^{*(1)}) \leq J_{\text{SLSVM}}(\underline{w}^{*(2)}, b^{*(2)}, \underline{\eta}^{*(2)}). \tag{B.5}$$

This reasoning extends to subsequent iterations, yielding the monotonic relationship:

$$J_{\text{SLSVM}}(\underline{w}^{*(i-1)}, b^{*(i-1)}, \underline{\eta}^{*(i-1)}) \leq J_{\text{SLSVM}}(\underline{w}^{*(i)}, b^{*(i)}, \underline{\eta}^{*(i)}). \tag{B.6}$$

For any finite dataset $\mathcal{D}_P = \{(\underline{\psi}(k), q(k))\}_{k=1}^N$, the cost function $J_{\text{SLSVM}}(\underline{w}, b, \underline{\eta})$ is bounded above by $J_{\text{SLSVM}}^{\max}$, achieved when $(\underline{w}'^*, b'^*, \underline{\eta}'^*)$ solves:

$$\begin{cases} \min\limits_{\underline{w}', b', \underline{\eta}'} 0.5\left\|\underline{w}'\right\|^2 \\ s.t.: \hat{q}(k)((\underline{w}')^T \underline{\varphi}(k) + b') \geq 1 - \eta_k', k = 1,2,\cdots,N \\ \eta_k' \geq 0, k = 1,2,\cdots,N \end{cases} \tag{B.7}$$

Therefore, the sequence $\left\{\underline{w}^{*(i)}\right\}$ generated by the SL-SVM algorithm converges, demonstrating the monotonic non-decreasing behavior of the classifier parameters through iterations.

Appendix C. Convergence Analysis of Parameter Vectors in the Third Algorithm

C-1. Intuitive Overview

The convergence of the submodel parameter vectors follows from the properties of the RLS estimator. With each update using a new data point, the error covariance matrix $P_{z_j}^{(i)}$ is guaranteed to decrease (Eq. C.2). This shrinking covariance reflects the algorithm's growing confidence in its estimates. Consequently, the magnitude of parameter updates $\left\|\hat{\underline{\theta}}_{z_j}^{(i)} - \hat{\underline{\theta}}_{z_j}^{(i-1)}\right\|$ diminishes over iterations. When updates become negligibly small (below ε_2), the parameters have effectively converged to their final values.

C-2. Formal Proof

To analyze the convergence of the submodels' parameter vectors, for any $j \in \{1,2,\cdots,p_h\}$ and corresponding $z_j \in \{1,2,\cdots,s\}$, consider the covariance matrix update:

$$P_{z_j}^{(i)} = P_{z_j}^{(i-1)} - \frac{P_{z_j}^{(i-1)}\underline{\varphi}_{j,i}^{HPSV}(\underline{\varphi}_{j,i}^{HPSV})^T P_{z_j}^{(i-1)}}{1+(\underline{\varphi}_{j,i}^{HPSV})^T P_{z_j}^{(i-1)}\underline{\varphi}_{j,i}^{HPSV}} \quad , P_{z_j}^{(0)} = \left(\frac{1}{N_{c_{z_j}}}\sum_{k=1}^{N_{c_{z_j}}}\tilde{\underline{\varphi}}(k)\tilde{\underline{\varphi}}^T(k)\right)^{-1} > 0. \tag{C.1}$$

Since $P_{z_j}^{(0)}$ is positive definite and symmetric, both $P_{z_j}^{(i-1)}$ and the subtracted term remain positive definite and symmetric for all iterations $i = 2,3,\cdots$. To demonstrate that $P_{z_j}^{(i)}$ maintains positive definiteness, we prove:

$$P_{z_j}^{(i-1)} > \frac{P_{z_j}^{(i-1)}\underline{\varphi}_{j,i}^{HPSV}(\underline{\varphi}_{j,i}^{HPSV})^T P_{z_j}^{(i-1)}}{1+(\underline{\varphi}_{j,i}^{HPSV})^T P_{z_j}^{(i-1)}\underline{\varphi}_{j,i}^{HPSV}}. \tag{C.2}$$

Right-multiplying by $(P_{z_j}^{(i-1)})^{-1}$ yields:

$$I > \frac{P_{z_j}^{(i-1)}\underline{\varphi}_{j,i}^{HPSV}(\underline{\varphi}_{j,i}^{HPSV})^T}{1+(\underline{\varphi}_{j,i}^{HPSV})^T P_{z_j}^{(i-1)}\underline{\varphi}_{j,i}^{HPSV}}. \tag{C.3}$$

Equivalently

$$I(1 + (\underline{\varphi}_{j,i}^{HPSV})^T P_{z_j}^{(i-1)}\underline{\varphi}_{j,i}^{HPSV}) > P_{z_j}^{(i-1)}\underline{\varphi}_{j,i}^{HPSV}(\underline{\varphi}_{j,i}^{HPSV})^T. \tag{C.4}$$

Noting that $\mathrm{trace}(P_{z_j}^{(i-1)}\underline{\varphi}_{j,i}^{HPSV}(\underline{\varphi}_{j,i}^{HPSV})^T) = (\underline{\varphi}_{j,i}^{HPSV})^T P_{z_j}^{(i-1)}\underline{\varphi}_{j,i}^{HPSV}$, we obtain:

$$I(1 + \mathrm{trace}(P_{z_j}^{(i-1)}\underline{\varphi}_{j,i}^{HPSV}(\underline{\varphi}_{j,i}^{HPSV})^T)) > P_{z_j}^{(i-1)}\underline{\varphi}_{j,i}^{HPSV}(\underline{\varphi}_{j,i}^{HPSV})^T. \tag{C.5}$$

This inequality always holds, confirming that $P_{z_j}^{(i)}$ remains positive definite in each iteration. Consequently, the diagonal elements of $P_{z_j}^{(i)}$ decrease monotonically, approaching zero asymptotically. The symmetry of $P_{z_j}^{(i)}$ follows directly from the symmetry of its constituent matrices.

Now consider the parameter update equation:

$$\hat{\underline{\theta}}_{z_j}^{(i)} = \hat{\underline{\theta}}_{z_j}^{(i-1)} + P_{z_j}^{(i)}\underline{\varphi}_{j,i}^{HPSV}(y_{j,i}^{HPSV} - (\underline{\varphi}_{j,i}^{HPSV})^T\hat{\underline{\theta}}_{z_j}^{(i-1)}), \tag{C.6}$$

As $P_{z_j}^{(i)}$ approaches zero, the correction term $P_{z_j}^{(i)}\underline{\varphi}_{j,i}^{HPSV}(y_{j,i}^{HPSV} - (\underline{\varphi}_{j,i}^{HPSV})^T\hat{\underline{\theta}}_{z_j}^{(i-1)})$ vanishes, leading to $\hat{\underline{\theta}}_{z_j}^{(i)} = \hat{\underline{\theta}}_{z_j}^{(i-1)} =$

$\hat{\underline{\theta}}_{z_j}$. Thus, the algorithm converges to fixed parameter vectors $\hat{\underline{\theta}}_{z_j}$ for all $z_j \in \{1,2,\cdots,s\}$. Although convergence rates may vary across parameters, the stopping criterion $\max_{z_j \in \{1,2,\cdots,s\}} (\left\| \hat{\underline{\theta}}_{z_j}^{(i)} - \hat{\underline{\theta}}_{z_j}^{(i-1)} \right\|) < \varepsilon_2$ ensures comprehensive convergence of all parameter vectors.

HOW TO CITE THIS ARTICLE
M. Lotfi, M. Bagher Menhaj,M. Karrari, M. Haeri , Complete Automated Structure Discovery and Parameter Estimation for Piecewise Affine Models with Guaranteed Convergence, AUT J. Model. Simul., 57(2) (2025) 191-214.

DOI: 10.22060/miscj.2026.25021.5447