Technische Universität München
Fakultät für Elektrotechnik und Informationstechnik

Hybrid System Identification

Yingwei Du

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender: Prof. Dr.-Ing. Eckehard Steinbach

Prüfende/-r der Dissertation:

1. Prof. Dr.-Ing./Univ. Tokio Martin Buss

2. Prof. Dr.-Ing. Matthias Althoff

Die Dissertation wurde am 23.09.2021 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 10.03.2022 angenommen.

Preamble

This dissertation summarizes my research conducted at the Chair of Automatic Control Engineering (LSR), at the Technical University of Munich (TUM), Germany. I am always grateful for the skills and experience I have obtained in this chair, where I have been allowed to freely study interesting topics and collaborate with talented colleagues for valuable research results. I would like to seize this opportunity to thank all the people who have offered me great support and assistance during my stay in this chair.

Foremost, I would like to sincerely thank my advisor and head of the chair Prof. Martin Buss for his strong support during my years in LSR. His continuous trust and support sustainably encourage me to overcome the challenges when seeking for novel methods for specific research problems. I benefit a lot from the freedom he granted in this lab, as well as his insightful comments and suggestions, so that I can concentrate on the topics I am interested in. I appreciate all his valuable contributions to my work. I would also like to thank Dr. Fangzhou Liu for her constructive suggestions on this my research of system identification dynamics and other related works. Furthermore, I want to thank Prof. Jianbin Qiu from the Harbin Institute of Technology for his valuable suggestions and feedback on my work.

I thank the whole team at LSR for both the productive and the fun times we shared. A special thanks goes to my friends and colleagues in LSR: Zengjie Zhang, Tim Bruedigung, Stefan Friedrich, Volker Gabler, Rameez Hayat, Gerold Huber, Tong Liu, Özgür Ögüz, Alexander Pekarovskiy, Cong Li, Yuhong Chen, Ni Dang and Zhehua Zhou. Without our extensive discussions, your constructive feedback, and our weekly meeting, this work would not have been possible. I would also like to thank Dr. Qingchen Liu in the Chair of Information-oriented Control (ITR) who helped me so much with my academic career.

Most importantly, no words can fully express my gratefulness to my family. I would like to thank my parents and my grandmother for their love, continuous support and valuable advice in all these years. I thank my grandfather in heaven for his spiritual encouragement. Last but not least, I am forever grateful to my wife Shanshan Huang for her love, support, and encouragement during my doctor career.

Munich, August 2021 Yingwei Du

Abstract

Hybrids systems arise when the continuous dynamic behaviors interact with the discrete ones, which provide an attractive framework for describing the complex systems or approximating the nonlinear systems. Prior to the analysis, verification, computation, and control of these systems, the identification of hybrid systems is essential. Naturally, this thesis is motivated by solving the identification problem of hybrid systems, including offline identification and online identification.

There are tremendous systems that can be categorized as hybrid systems. In this thesis, we concentrate on the identification of several representative hybrid systems, i.e., PieceWise Affine AutoRegressive eXogenous (PWARX) system, continuous-time PieceWise Affine (PWA) system, and continuous-time Switched Nonlinear System (SNS). Universally, the identification of hybrid systems includes estimating the number of subsystems, the switching signal/polyhedral partitions, and the subsystem models. The PWARX systems draw attention due to their simple formulations and universal approximation properties in recent years. In the thesis, we propose a novel method to address the identification problem of PWARX systems, which is motivated by reducing the computational complexity while estimating the polyhedral partitions. In particular, a cluster-based algorithm is designed to acquire the number of submodels, the initial labeled data set, and the initial parameters corresponding to each submodel. Additionally, we develop a modified self-training support vector machine algorithm to simultaneously identify the hyperplanes and parameters of each submodel with the outputs of the cluster-based algorithm. The proposed algorithm is computationally efficient for partition estimation and able to accomplish this task with only a small quantity of classified regression vectors.

The continuous-time PWA systems and switched nonlinear systems are more appropriate to describing the intrinsic mechanism of the complex system and designing the corresponding control strategies. However, the existing methods for identifying them are exiguous, especially for the online identification methods. Therefore, the motivation is to build a generalized framework for the continuous-time hybrid system in state-space form. The main challenge is the coupling between the subsystem dynamics and the switching behavior. In this dissertation, this framework is composed of three stages: a) switching detection and active mode estimation, b) subsystem identification, c) switching signal/polyhedral partition estimation. The corresponding algorithms and their proof is also presented. Specifically, the algorithms for switching detection and active mode estimation are designed based on the delay error dynamics (continuous-time PWA systems) and the subspace projection method (continuous-time SNSs). Following the estimated active mode, the modified integrated concurrent learning identifier is proposed to update the subsystem parameters with the assistance of the Luenberger-observer and state reset mechanism. Finally, the switching signal/polyhedral partitions are confirmed via the estimates. In addition, some refinement strategies are also provided to cope with the extreme situations (for continuous-time PWA systems) and the model structure selection (for continuous-time SNSs). The effectiveness of these proposed identification approaches for hybrid systems are illustrated via simulation results.

Zusammenfassung

Hybride Systeme entstehen, wenn die kontinuierlichen dynamischen Verhaltensweisen mit den diskreten interagieren, was einen attraktiven Rahmen für die Beschreibung der komplexen Systeme oder eine Annährung an nichtlineare Systeme bietet. Vor der Analyse, Überprüfung, Berechnung und Steuerung dieser Systeme ist die Identifizierung hybrider Systeme unabdingbar. Diese Arbeit ist durch die Lösung des Identifikationsproblems hybrider Systeme, einschließlich Offline-Identifikation und Online-Identifikation, motiviert. Es gibt eine Vielzahl an Systeme, die als Hybridsystem kategorisiert werden können. In dieser Arbeit konzentrieren wir uns auf die Identifizierung mehrerer repräsentativer Hybridsysteme, d. h. PWARX-System, zeitkontinuierliche PWA Systeme und geschaltete nichtlineare Systeme. Allgemein umfasst die Identifizierung von Hybridsystemen die Schätzung der Anzahl von Subsystemen, des Schaltsignals/der polyhedrischen Partition und der Subsystemmodelle. Die PWARX-Systeme machen in den letzten Jahren durch ihre einfache Formulierung und universellen Näherungseigenschaften auf sich aufmerksam. In der Dissertation schlagen wir eine neuartige Methode vor, um das Identifikationsproblem von PWARX-Systemen anzugehen, die durch die Reduzierung der Rechenkomplexität bei der Schätzung der polyedrischen Partitionen motiviert wird. Insbesondere ist ein clusterbasierter Algorithmus entworfen worden, um die Anzahl von Untermodellen, den gekenn-zeichneten und die Anfangsparameter des entsprechenden Untermodell zu erfassen. Darüber hinaus entwickeln wir einen modifizierten selbstlernenden Support-Vektor-Machine-Algorithmus, um gleichzeitig die Hyperebenen und Parameter jedes Teilmodells mit den Ausgaben des clusterbasierten Algorithmus zu identifizieren. Der vorgeschlagene Algorithmus ist recheneffizient für die Bereichsschätzung und in der Lage, diese Aufgabe mit nur einer kleinen Menge klassifizierter Regressionsvektoren zu erfüllen. Betrachtet man die zeitkontinuierlichen PWA-Systeme und geschaltete nichtlineare Systeme, sind sie besser geeignet, um den intrinsischen Mechanismus des komplexen Systems zu beschreiben und die entsprechenden Regelstrategien für sie zu entwerfen. Die Identifizierungsmethoden für sie sind jedoch aufgrund ihrer Formulierungen, insbesondere für die Online-Identifizierung, ausbaufähig. Daher besteht die Motivation darin, einen allgemeinen Rahmen für das zeitkontinuierliche Hybridsystem in Zustandsform zu erstellen. Die Hauptherausforderung ist die Kopplung zwischen der Subsystemdynamik und dem Schaltverhalten. In dieser Arbeit besteht dieser Rahmen aus drei Stufen: a) Schaltdetektion und aktive Modusschätzung, b) Subsystemidentifikation, c) Schaltsignalschätzung/polyedrische Partitionsschätzung. Die entsprechenden Algorithmen und deren Beweis werden in der Arbeit vorgestellt. Insbesondere werden die Algorithmen für die Schalterkennung und die Schätzung des aktiven Modus basierend auf der Verzögerungsfehlerdynamik (kontinuierliche PWA-Systeme) und dem Unterraumprojektionsverfahren (kontinuierliche SNSs) entworfen. Nach dem geschätzten aktiven Modus wird die modifizierte I-CL-Kennung vorgeschlagen, um die Subsystemparameter mit Hilfe des Luenberger-Beobachtungsverfahren und des Zustandsrücksetzmechanismus zu aktualisieren. Schließlich werden die Schaltsignal- /polyedrischen Partitionen über die geschätzten aktiven Modi und die Subsystemparameter bestätigt. Darüber hinaus werden auch einige Verfeinerungsstrategien bereitgestellt, um die Extremsituationen (für zeitkontinuierliche PWA-Systeme) und die Modellstrukturauswahl (für zeitkontinuierliche SNSs) zu bewältigen. Die Wirksamkeit dieser Identifikationsansätze für Hybridsysteme wird durch Simulationsergebnisse veranschaulicht.

Contents

Notation

Acronyms and Abbreviations

AHS	Automated Highway System
PWA	Piecewise affine (system)
PWL	Piecewise linear (system)
PWARX	Piecewise autoregressive exogenous (system)
HHARX	Hing hyperplane autoregressive exogenous (system)
PWNS	Piecewise nonlinear system
PWNARX	Piecewise nonlinear autoregressive exogenous (system)
SNARX	Switched nonlinear autoregressive exogenous (system)
SARX	Piecewise autoregressive exogenous (system)
SNS	Switched nonlinear system
SLS	Switched linear system
PE	Persistently exciting/persistent excitation
CL	Concurrent learning
LC	Local cluster
sub LC	Sub local cluster
I-CL	Integral concurrent learning
SVM	Support vector machine
M-SVM	Multicategory-Support vector machines
RLS	Recursive least squares
LS	Least squares
BFR	Best fit rate
WMR	Wheeled mobile robot
PBH	Popov-Belevitch-Hautus (Test)

Conventions

Scalars, Vectors, and Matrices

Scalars and *vectors* are denoted by lower case letters. *Matrices* are denoted by upper case letters.

Subscripts and Superscripts

$\dot{a}$	Time derivative of a
$\hat{a}$	Estimate of a
$\tilde{a}$	Difference between estimated and true value of a
a^{-1}	Inverse of a
a^{T}	Transpose of a
a_i	a_i refers to a of the i-th subsystem
a_c	Point a_c is the center of a local cluster
$(A)_{[i*]}$	i-th row vector of matrix A
$(A)_{[*i]}$	i-th column vector of matrix A
$(A)_{[\bar{i}*]}$	Matrix formed by removing the i-th row from A
$(A)_{[*\bar{i}]}$	Matrix formed by removing the i-th column from A
$(A)_{[\bar{i},\bar{j}]}$	Matrix formed by removing the i-th row and j-th column from A

Number Sets

$\mathbb{N}$	set of natural numbers
$\mathbb{N}^+$	set of positive natural numbers
$\mathbb{R}$	set of real numbers
$\mathbb{R}^+$	set of positive real numbers
$\mathbb{C}$	set of complex numbers

Operations

$\text{vec}(A)$	vec-operator which concatenates the columns of the matrix A
$\text{Cols}(A)$	space spanned by column vectors of matrix A
$\text{rank}(A)$	rank of matrix A
$\#\mathcal{A}$	cardinality of set $\mathcal{A}$
$\mathcal{B}/\mathcal{A}$	the set of elements in set B but not in set A.
$\preceq_{[i]}$	list of operators $\leq$ and $<$ for the i-th system
$\text{var}(\cdot)$	variance of a sequence of scalars.

For a vector $b \in \mathbb{R}^n$, the following operations exist.

$\|\|b\|\|_1$	ℓ_1 norm:$\|b\|_1 = \sum_{i=1}^{n} \|b_i\|$
$\|\|b\|\|_2$	ℓ_2 norm:$\|b\|_2 = \sqrt{\sum_{i=1}^{n} b_i^2}$
$\|\|b\|\|_\infty$	ℓ_∞ norm: $\|b\|_\infty = \max_i \|b_i\|$
$d(b_1, b_2)$	standardized Euclidean distance of vector b_1, b_2: $d(b_1, b_2) = \sqrt{\sum_{i=1}^{n} \frac{(b_i - b_i)^2}{s_i^2}}$.

Symbols

General

θ	system parameter vector
s	number of subsystems
$\mathcal{X}$	system polyhedral partition
u	input of a dynamic system
y	output of a dynamic system
V	Lyapunov function
δ, α, ρ	small constants
Δt	integration window
J	cost function

P	positive definite matrix in a quadratic Lyapunov function
σ	active mode
η	moving window
t	time
T	period of input signal
$\mathcal{N}$	normal distribution
$\mathcal{H}_i$	matrix defining the hyperplanes of region $\mathcal{X}_i$
μ_i	number of hyperplanes defining $\mathcal{X}_\rangle$
A_m	stable design matrix in parameter identifiers or reference system

PWARX system

x_k	regression vector
n_a, n_b	model orders
φ_k	extended regression vector
e_k	output-measurement noise
N	number of data samples for identification
$\mathcal{D}$	set of data points
$\mathcal{L}$	local cluster
(x_c, y_c)	center of a local cluster
$(\check{x}, \check{y})$	data points in a local cluster
ℓ	label of a data point
$\mathcal{D}_l$	initial labeled data set
$\mathcal{D}_T$	collection of the center of each sub LC
$\mathcal{G}$	classifier group
ξ	linear penalty
N_T	total amount of training data
r	the number of iterations
γ	geometric margins

Continuous-Time PWA system

x	state of a dynamic system
A_i	system matrix of the i-th subystem of a PWA system
B_i	input matrix of the i-th subystem of a PWA system
f_i	affine input vector of the i-th subystem of a PWA system
ε	delay error
t_e	delay shift
Ξ	sampling matrix
N	sampling size
t_s	sampling interval
Φ	state-transition matrix
χ	standard deviation
ε_0	initial error state vector
t_{sw}	switching time instant
$\mathbf{X}_i$,$\mathbf{U}_i$	integrated history stack of the i-th subsystem
$t_{i,q}^{in}$	Switching time: q-th activation of subsystem i
$t_{i,q}^{out}$	time: q-th deactivation of subsystem i
T_{sw}	sequence of switching instants
e_i	prediction error of the i-th subsystem

Continuous-Time SNS

x	state of a dynamic system
φ	basis function
κ	parameter of basis function
ψ	regression vector formed by basis functions
τ_Δ	time required to recognize the active mode correctly
τ_{dwell}	minimum dwell time
Υ_i,Φ_i	the sampling matrices of i-th subsystem

Π	projection matrix
ϵ	residual computed for switching detection
h_{NN}	last diagonal element of Π
ζ	threshold vector
N	sampling size
t_s	sampling interval
$\mathbf{H}_i$	history stack of i-th subsystem
Θ	parameter of the i-th subsystem before transformation
e_i	prediction error of the i-th subsystem
S	residual matrix for MSS

List of Figures

List of Tables

1. Introduction

Hybrid systems are dynamic systems that exhibit both continuous and discrete dynamic behaviors. They are especially profound in many technological systems, in which logic decision-making and embedded control actions are integrated with continuous physical processes. Hybrid dynamics provide a convenient framework for modeling systems in a wide range of engineering applications:

- **Mechanical Systems**: Continuous motions which are interrupted by collisions.
- **Electrical Circuits**: Continuous phenomena such as the charging of capacitors, etc. are interrupted by switches opening and closing, or diodes going on or off.
- **Chemical Process**: Continuous evolution of chemical reactions which is controlled by valves and pumps.
- **Embedded Computation Systems**: Digital computer interacts with a mostly analogue environment.

In addition, a majority of the complex dynamical systems around us may reasonably be depicted in hybrid terms, such as epidemiology [1], legged locomotion [2], cascading failures on the electrical grid [3], gear shift system [4], and automated highway system [5]. In particular, we describe the gear shift system and automated highway system in detail for intuitive expressions.

Gear Shift System

The gear shift system is indispensable in cars, which is used to move a vehicle forward, in reverse, or remain neutral. The core of the shift system is a gearbox, whose gears can be switched manically or automatically. Figure 1.1 shows a model of a car with a gearbox having four gears.

The longitudinal position of the car along the road is denoted by x_1 and its velocity by x_2 (lateral dynamics are ignored). The model has two control signals: the gear denoted $\mathbf{gear} = 1, \ldots, 4$ and the throttle position denoted $u = [u_{\mathbf{min}}, u_{\mathbf{max}}]$. Gear shifting is necessary because little power can be generated by the engine at very low or very high engine speed. The function f_i represents the efficiency of gear i. Typical shapes of the functions f_i are shown in Figure 1.2.

According to the description of the gear shift system, it can be modeled as a hybrid system. The dynamic of each subsystem (gear) is continuous and nonlinear. In addition, the gears are also switched manually or automatically during driving, which implies the discrete dynamics of the system.

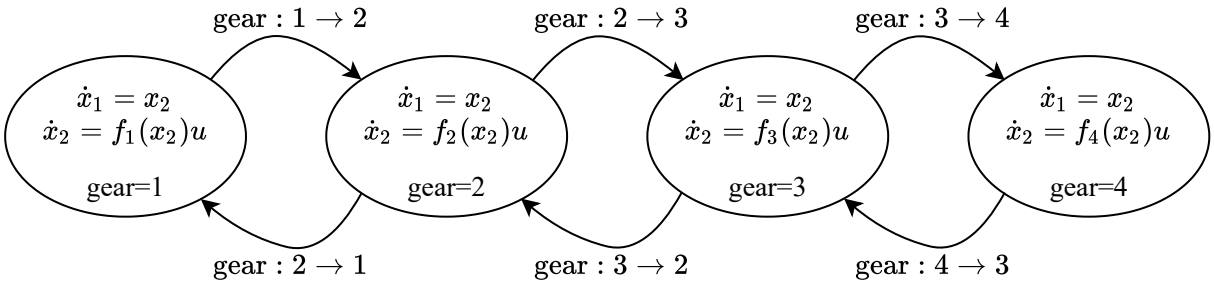


Figure 1.1.: A hybrid system modelling a car with four gears.

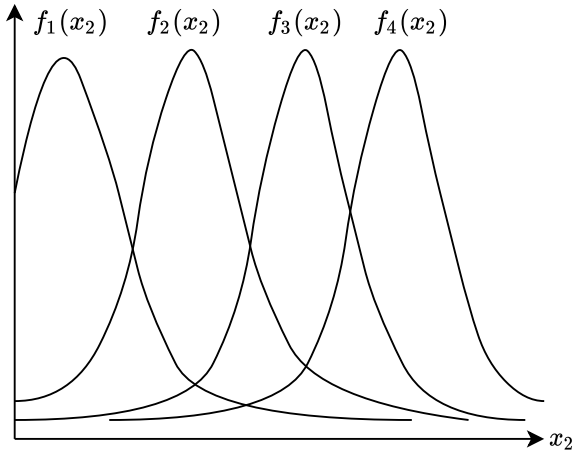


Figure 1.2.: The efficiency functions of the different gears.

Several interesting control problems can be posed for this gear shift system, such as designing an optimal control strategy to drive from A point to B point in minimum time. The problem is not trivial if we include the reasonable assumption that each gear shift takes a certain amount of time. It is worth noting that the design of the optimal controller is founded on the known efficiency functions of each gear f_i, the switching sequence, and the switching instants.

Automated Highway System

Highway congestion problem draws more and more attention these days, especially in and around urban areas. One of the feasible solutions considered for this problem is traffic automation, either partial or full. The use of an automated system that performs some or all of the tasks of the driver may reduce or eliminate human errors and hence improve safety. Moreover, as the automatic controller can react to disturbances faster than a human driver, automation may also decrease the average inter-vehicle spacing and hence increase throughput and reduce congestion and delays.

The design of an automated highway system (AHS) is an extremely challenging control yet a popular problem. Thus, a number of methods have been proposed for addressing it. One of the most forward-looking AHS designs involves a fully automated highway system that supports platooning of vehicles. The platooning concept [6] assumes that traffic on

the highway is organized in groups of tightly spaced vehicles (platoons). The platooning structure achieves a balance between safety and throughput. In recent years, computational and experimental studies have shown that an AHS that supports platooning is not only technologically feasible but, if designed properly, may lead to an improvement of both the safety and the throughput of the highway system, under normal operation.

Implementation of the platooning concept requires automatic vehicle control, since human drivers are not fast and reliable enough to produce the necessary inputs. A hierarchical controller is employed to manage the complexity of the design process. The controller is organized in four layers (Figure 1.3). The top two layers, called network and link, reside on the roadside and are primarily concerned with throughput maximization, while the bottom two, called coordination and regulation, are the major concerns in the safety and reside on the vehicles. The physical layer is not part of the controller. It contains the "plant", i.e. the vehicles and highway, with their sensors, actuators, and communication equipment. Subsequently, we focus on the description of the coordination and regulation layers.

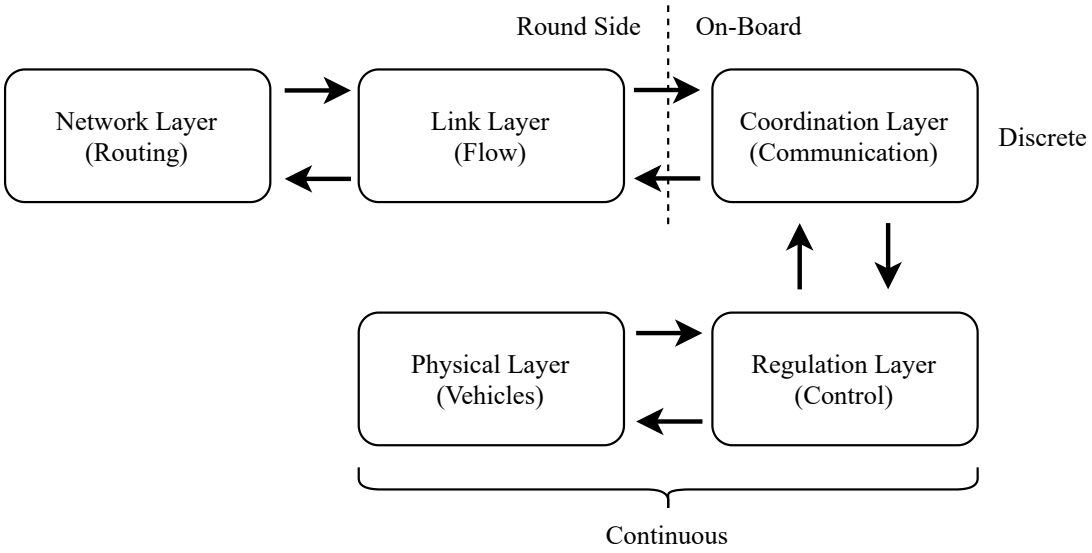


Figure 1.3.: The Automated Highway System control hierarchy.

The coordination layer coordinates the operation of neighbouring platoons by choosing manoeuvres that the platoons need to carry out. For normal operation, these manoeuvres are join to join two platoons into one, split to break up one platoon into two, lane change, entry and exit. The coordination layer is primarily a discrete controller. It uses communication protocols, in the form of finite state machines, to coordinate the execution of these manoeuvres between neighbouring vehicles.

The regulation layer receives the coordination layer commands and readings from the vehicle sensors and generates throttle, steering and braking commands for the vehicle actuators. For this purpose it utilises a number of continuous time feedback control laws that use the readings provided by the sensors to calculate the actuator inputs required for a particular manoeuvre. In addition to the control laws needed for the manoeuvres, the regulation layer makes use of two default controllers, one for leader and one for follower operation.

The interaction between the coordination layer (which is primarily discrete) and the regulation layer (which is primarily continuous) gives rise to interesting hybrid dynamics. To

ensure the safety of the AHS, one needs to verify that the closed loop hybrid system does not enter a bad partition of its state space (e.g. does not allow any two vehicles to collide at high relative velocity). It can be shown that information available through discrete coordination can be used together with appropriate continuous controllers to ensure the safety of the closed loop hybrid system.

Even though the hybrid models are more appropriate to describe the complex systems and approximate the nonlinear systems, most of the researchers still concentrate on studying dynamical systems which are either completely continuous or discrete due to their simple formulations. However, it is a platitude that completely continuous or discrete dynamical models are inadequate to describe most of the real systems. As a consequence, the field of hybrid system modeling has been greatly concerned in the last few decades.

Prior to the analysis, verification, computation, and control of these dynamic systems and phenomena, the corresponding hybrid models need to be identified. Therefore, the identification of hybrid systems has been greatly concerned in recent years [7–14]. Hybrid system identification aims at using statistical methods to construct the models of hybrid systems mainly from experimental data. Generally, two types of models are in the field of system identification: grey-box model and black-box model [15]. The focus of the grey-box model identification is the estimation of unknown free parameters while knowing the model structure. Alternatively, no prior model is available for the black-box model and the model structure is also required to be identified. From another perspective, the identification methods can be categorized as online and offline. The online identification methods estimate the parameters of a model when new data is available during the operation of the model. Considering offline identification, all the input/output data are collected before estimating the model parameters. The estimated parameters are updated with time in the online identification methods while invariant for the offline methods.

One of the main difficulties to discuss the identification of hybrid system is that the term "hybrid" is not restrictive: the interpretation of the term could be stretched to include virtually any dynamical system we can think of [16]. Obviously, it is unrealistic to discuss the identification for all hybrid systems. Therefore, we briefly categorized the hybrid systems into different types and concentrate on the identification of some representative types. The classified hybrid systems are shown in Table 1.1. It should be noted that hybrid systems can be classified according to various criteria. The switching mechanism and formulation are selected as the criteria since they are the major concerns of our thesis.

Various approaches have been presented to cope with the identification of hybrid systems [11–14, 17–19]. The offline identification for discrete-time hybrid systems in the input-output form is spotlighted in these approaches due to their easy implementation [20–23]. However, the traditional partition estimation for switching dependent hybrid systems is generally expensive and time-consuming, especially for large data sets [24], being one of the critical challenges in hybrid system identification. Compared with offline identification, online identification for continuous-time hybrid systems in the state-space form is more challenging but essential. It follows that the continuous-time hybrid systems in the state-space form are more effective while describing the intrinsic mechanism of dynamical systems. Also, direct identification of continuous-time models based on sampled data can outperform the discrete-time models in the case of rapidly or irregularly sampled data [25, 26]. In addition, the online or recursive identification methods are in favor for real-time applications due to their abilities to handle large amounts of data. Even with the aforementioned advantages,

Table 1.1.: Hybrid System Classification

Hybrid System	
Input-output form	SARX system, PWARX system, HHARX system, ...
State-space form	PWA system, PWL system, SLS, SNS, ...
Time dependent switching	SARX system, SLS, SNS, ...
State dependent switching	PWA system, PWL system, HHARX system,...
Linear subsystem	PWL system, SLS, SARX system, ...
Affine subsystem	PWA system,PWARX system, ...
Nonlinear subsystem	SNS, PWNS, PWNARX system, SNARX system, ...

only a few online identification methods are proposed for continuous-time hybrid systems in state-space form due to critical challenges. It is worth noting that these methods also have limitations in some aspects. Therefore, one of the critical challenges in hybrid system identification is to build a consummated framework for online identification of continuous-time hybrid systems in state-space form.

While the above introduction sets the general motivation for this work, it is obvious that researching the identification of all hybrid systems is unrealistic. Therefore, we focus on the identification of three kinds of representative hybrid systems in the thesis, which are the PieceWise AutoRegressive eXogenous (PWARX) system, continuous-time PieceWise Affine (PWA) system, and continuous-time switched nonlinear system (SNS). The following section describes the challenges addressed in this thesis in greater detail.

1.1. Challenges

In this thesis, we concern the following three challenges revolving the identification of hybrid systems.

i) **How to achieve the efficient offline identification of the PWARX system?**

ii) **How to accomplish the online identification of the continuous-time PWA system?**

iii) **How to carry out the online identification of the continuous-time switched nonlinear system?**

Among multitudes of hybrid systems, the representative ones are the PWARX system, continuous-time PWA system, and continuous-time SNS. These models have their own advantages and disadvantages while approximating the nonlinear systems and describing the complex systems. The strengths of the PWARX system are its simple formulation and universal approximation properties. On the other hand, the input-output form weakens its

ability to describe and control the state variables. Considering the continuous-time PWA systems, the intrinsic mechanism of the complex systems is easier to be described by them, which implies less effort while designing their control strategy. In addition, the continuous-time SNSs prevent the heavy computation burden by using tremendous subsystems to approximate the nonlinear systems. Also, the intrinsic mechanism of the complex systems, especially the nonlinear terms can be illustrated apparently through the SNSs. However, these advantages are built on its complex formulation which leading to hard identification of the system. To solve the first challenge, the polyhedral partition estimation is the major concern to improve the efficiency of offline identification for PWARX systems. For the second and third challenges, the framework based on integral concurrent learning is established for the online identification of continuous-time PWA systems and further enhanced for the continuous-time SNSs, respectively. A more detailed description of these challenges is provided in the following subsections. At the end of this chapter, we highlight the contribution and illustrate the structure of this dissertation.

1.1.1. Offline Identification of PWARX System

The first challenge is motivated by the fact that the offline identification methods for PWARX systems are generally expensive and time-consuming, especially for large data sets. It is beneficial to improve the efficiency of the PWARX system identification for some scenarios with limited computation power and memory.

PWARX systems are a subclass of hybrid systems representing the input-output relationship of PWA systems, with the specification that the switching mechanism is determined by a polyhedral partition of the regressor space. The research of PWA systems starts in the mid-twentieth. The modeling of saturated systems with a partitioning of the state space is proposed by Kalman in the 1950s as the first work related to the piecewise linear systems [27]. In the 1970s, PWA models experienced their first major growth in interest. The circuit community was in need for efficient simulation and analysis tools [28, 29] for large scale circuits with piecewise linear elements such as diodes. Later, Sontag provided pioneering insights into discrete-time PWA systems that are still appreciated today [30, 31]. Today, PWARX systems are particularly focused among PWA systems due to their simple formulation and universal approximation properties. Also, the identification of PWARX models has received great attention as the foundation of model-based control [32–36].

The offline identification of PWARX models incorporates the estimation of both parameters of each submodel and polyhedral partitions. Thus the identification approaches developed in previous works generally consist of two stages, i.e., i) identifying the parameters of each affine submodel and ii) estimating polyhedral partitions of the input-state space [37–41]. For the first stage, various methods have been presented to identify the parameters of each affine submodel. However, unlike the first stage, the SVM algorithm is dominantly employed to estimate the partitions of submodels [37–41]. It follows that the SVM algorithm is capable of estimating the hyperplanes precisely. However, effective and well-performed as SVM is regarding the identification tasks, it can be numerically inefficient. In SVM algorithm, all the data points are required to be labeled and classified before estimating the polyhedral partitions in these algorithms, which is generally expensive and time-consuming, especially for large data sets [24]. Besides, computing feasible partitions through the SVM algorithm may suffer high computational complexity [42] in this case. Therefore, the major concern

of the efficient offline PWARX identification in our thesis is to provide a novel algorithm to reduce the computational complexity of the polyhedral partition estimation.

1.1.2. Online Identification of Continuous-Time Hybrid Systems

Regarding the second and third challenges, it is essential to develop a complete framework for the online identification of the continuous-time PWA systems and continuous-time SNSs. They are two typical hybrid systems, whose formulations are continuous-time in state-space form. These systems are profitable while describing the intrinsic mechanism of the complex system and designing the corresponding control strategy. However, the costs of these profits are the difficulties while doing their identifications since their formulations are rather complicated.

Compared with offline identification, the online or recursive identification methods are in favor for real-time applications due to their abilities to handle large amounts of data. Moreover, the online identification of the parameters is the foundation of adaptive control. The continuous-time models are formulated in differential equations. They are probably more mainstream in science and engineering, and studied more extensively, than discrete-time models, because various natural phenomena (e.g., motion of objects, flow of electric current) take place smoothly over continuous time. Considering the aforementioned advantages, some papers have been presented for the continuous-time PWA systems and SNS identification [43, 44]. However, these works have their own limitations: the switching signal/polyhedral partitions need to be known in advance or only the bi-model/tri-model systems can be estimated. Due to the aforementioned advantages and limitations, a framework for the online identification of continuous-time PWA systems and SNS is essential to be developed.

1.2. Contributions and Thesis Outline

This dissertation provides solutions to the above-motivated challenges arising for the identification of hybrid systems. Before going into detail, the interested readers may refer to Appendix A for the preliminary information on the proposed semi-supervised learning-based algorithm and Appendix B for the helpful background on the stability analysis of the switched system in the proposed online identification framework. Chapter 2 begins with an introduction to PWARX systems and provides the formulation of their identification problem. Subsequently, we describe the novel offline identification approach for the PWARX systems which are based on the semi-supervised learning. The computational complexity of the proposed algorithm for polyhedral partition estimation is discussed and compared with the original algorithms to show its strength. Furthermore, numerical experiments on the PWARX system identification are conducted to demonstrate the effectiveness of the proposed approaches. In Chapter 3, we propose a complete framework for the online identification of continuous-time PWA systems, whose effectiveness is validated through numerical experiments. The rigorous mathematical proof for the algorithms proposed in the framework is also given. The enhanced version of the framework from Chapter 3 is presented to handle the online identification of continuous-time SNS in Chapter 4. In the enhanced framework, we adopt a new active mode recognition algorithm which is based on the subspace projection and integrate

a proposed MSS strategy. Final conclusions and possible directions for future work will be given in Chapter 5.

Offline Identification for PWARX systems (Chapter 2)

For the offline identification of PWARX systems, we propose an efficient approach by imposing the semi-supervised learning algorithms in the polyhedral partition estimation procedure. It is worthy to point out that the state-of-art methods generally employ the SVM algorithms for the polyhedral partition estimation, whose precision is built on the expensive and time-consuming implementation, especially for large data sets. In fact, this drawback is the motivation of presenting the novel identification method. The proposed approach is comprised of a cluster-based algorithm and a modified self-training SVM algorithm. The cluster-based algorithm is designed to obtain the initial labeled data set, initial parameters of each submodel, and the number of submodels simultaneously. The modified self-training SVM algorithm then computes the parameters of submodels and estimates the polyhedral partitions with the outputs of the cluster-based algorithm. By imposing a customized selection strategy, the modified self-training SVM algorithm is capable of estimating the polyhedral partitions precisely with partially-labeled data set. Moreover, the computational complexity of the modified self-training SVM algorithm is analyzed and compared with the traditional SVM algorithm to justify its higher efficiency.

The material presented in Chapter 2 has been published in IEEE Transactions on Circuits and Systems I: Regular Papers [129].

Online Identification for Continuous-Time PWA Systems (Chapter 3)

In this chapter, we aim at offering a complete framework for the online identification of continuous-time PWA systems. To the best of our knowledge, the existing approaches for solving this problem have their corresponding drawbacks, such as the limitation of only identifying bi-model/tri-model PWA systems or the assumption of knowing the polyhedral partitions in advance. Strictly, these methods can not be regarded as generalized identification methods for the continuous-time PWA systems. Therefore, we present the generalized framework for the online identification of continuous-time PWA systems in this chapter. The framework separates the online identification into three tasks, i.e., the online active mode recognition, the online estimation of each subsystem, and the identification of polyhedral partitions. The first task is achieved by analyzing the discrete-time dynamics of the proposed delay error. Furthermore, an online algorithm is designed to estimate the number of subsystems and recognize the active mode hinged on the dynamics. According to the recognized active mode, we generalize the integral concurrent learning identifier to estimate the parameters of each subsystem and provide the convergence proof of the identifier. The generalized identifier avoids the estimation of state derivatives while maintaining the convergence of parameters. Taking into consideration the estimation of polyhedral partitions, the process is accomplished through a cost function that is designed via the estimated subsystem parameters. The proposition of the approach sheds light on the online identification of continuous-time PWA systems in state-space form.

The material presented in Chapter 3 has been published in IEEE Transactions on Circuits and Systems I: Regular Papers [130].

Online Identification for Continuous-Time SNSs (Chapter 4)

In order to identify the continuous-time SNSs online, we enhance the framework in chapter 3 by replacing the active mode recognition algorithm and involving the model structure selection (MSS) strategy. Considering the complex formulation of the switched nonlinear system, it is not a surprise that the online identification of continuous-time SNSs is more challenging, which results in even fewer works published than the online identification of continuous-time PWA systems. Whereas, the switched nonlinear systems play a critical role in the maintenance of describing the intrinsic dynamics and physical interpretations of the complex systems [45]. Therefore, we present the enhanced framework for the online identification of continuous-time SNSs. Similarly, the task for online identification can be decomposed into: online estimation of the number of subsystems, the switching sequence, and each subsystem. Concretely speaking, a theorem based on the projection matrix in statistics is introduced to address the identification of the number of subsystems and the switching sequence. Derived from the theorem, an online algorithm is designed to estimate the number of subsystems and the active mode. Following the estimation of active mode, the modified I-CL identifier is also employed for the subsystem identification, whose convergence proof is provided. Additionally, a refinement strategy for MSS is introduced to improve the efficiency of the proposed method. The strategy can be integrated into the framework to determine the model structure of each subsystem before identifying their parameters. Clearly, the proposed approach fills a gap in the field of online identification of continuous-time SNSs.

The material presented in Chapter 4 has been published in the International Journal of Robust and Nonlinear Control [131].

Semi-supervised Learning-based Offline Identification of PWARX Systems

2.

In this chapter, the semi-supervised learning-based method for the offline identification of PWARX system is introduced. Section 2.1 begins with a formal definition of PWARX systems, which is the target system to be identified throughout this chapter. Based on the literature review, we also derive limitations in the state-of-art identification approaches. Subsequently, the proposed identification method is described in Section 2.2, including the cluster-based algorithm and the modified self-training SVM algorithm. Furthermore, the computational complexity of the proposed modified self-training SVM algorithm is evaluated and compared with the original SVM algorithm in this chapter. Naturally, the numerical experiments are provided to validate the effectiveness of the proposed method in Section 2.3. The summary and the open problems to be solved are the subjects of Section 2.4.

2.1. Introduction to PWARX Systems and Problem Formulation

The PWARX systems considered in this chapter are a special class of hybrid systems, i.e., systems whose dynamics are characterized by a regression vector and a discrete switch-mode. The regression vector is defined as the collection of past input and output to observations and the discrete mode depends on a partitioning of the state-input space. Owning to their simple structure and universal approximation properties, PWARX models are especially applicable for the approximation of nonlinear systems. In the following, the formulation of the PWARX system is introduced and the limitations of the existing identification methods for PWARX systems are stated based on the literature review.

2.1.1. PWARX Systems

The PWARX model belongs to the class of discrete-time models in input-output form, which map inputs $u \in \mathbb{R}^{n_u}$ to outputs $y \in \mathbb{R}$. They consist of a collection of ARX models that operate on the same *regressor vector*. The *regressor vector*, which constitutes the continuous state of the system, is defined as the collection of previous input and output observations, i.e.,

$$x_k = [y_{k-1} \quad \dots \quad y_{k-n_a} \quad u_{k-1}^{\mathsf{T}} \quad \dots \quad u_{k-n_b}^{\mathsf{T}}]^{\mathsf{T}}, \tag{2.1}$$

where u_k and y_k are the input vector and output vector at time k, respectively. n_a and n_b are the system orders, and therefore, $x \in \mathbb{R}^{n_x}$ with $n_x = n_a + n_u n_b$.

Following the definition of *regression vector*, the PWARX model is defined as follows

$$y_k = \begin{cases} \theta_1^\mathsf{T} \begin{bmatrix} x_k \\ 1 \end{bmatrix} + e_k & \text{if } x_k \in \mathcal{X}_1, \\ \vdots & \vdots \\ \theta_s^\mathsf{T} \begin{bmatrix} x_k \\ 1 \end{bmatrix} + e_k & \text{if } x_k \in \mathcal{X}_s, \end{cases} \qquad k = 1, 2, \ldots, N. \tag{2.2}$$

where $e_k \in \mathbb{R}$ is the noise at time k and $\{\theta_i\}_{i=1}^s$ are the parameter vectors that define the constituent submodels of model (2.2). $s \in \mathbb{N}^+$ is the number of submodels. For the convenience, we denote $\varphi_k = [x_k^\mathsf{T} \quad 1]^\mathsf{T}$ as the *extended regression vector* at time k.

As aforementioned, the switching is state-depended, i.e., the current state vector stays in which polyhedron determines which subsystem is activated now. Therefore, the bounded polyhedron $\mathcal{X}_i$ with $i = 1, \ldots, s$ in equation (2.2) is the partition of the i-th submodel and is defined as

$$\mathcal{X}_i = \{x_k \in \mathbb{R}^{n_x} \mid \mathcal{H}_i \cdot \varphi_k \preceq_{[i]} 0\}, \tag{2.3}$$

where $\mathcal{H}_i \in \mathbb{R}^{\mu_i \times (n_x+1)}$ is the hyperplane matrix, i.e.,

$$\mathcal{H}_i = \begin{bmatrix} h_{1,i}^\top & \ldots & h_{\mu_i,i}^\top \end{bmatrix}, \tag{2.4}$$

and $\mu_i \in \mathbb{N}^+$ is the number of hyperplanes that bound the corresponding partition. In the equation (2.3), $\preceq_{[i]}$ indicates a list of μ_i operators $\leq$ and $<$ to decide which hyperplanes of $\mathcal{H}_i$ belong to $\mathcal{X}_i$. Note that the discrimination between $\leq$ and $<$ in the operator vector $\preceq_{[i]}$ is necessary to obtain a complete partition of the state space without overlapping partitions, i.e., there hold $\mathcal{X}_i \bigcap \mathcal{X}_j = \emptyset$, $\forall i \neq j$ and $\mathcal{X} = \bigcup_{i=1}^s \mathcal{X}_i$.

2.1.2. Problem Formulation and State of the Art Limitations

After introducing the PWARX system, we are now going to formulate the identification problem and present an overview of existing identification approaches. The most general problem formulation for the PWARX system identification reads as follows:

Problem 1 (**Identification of PWARX Systems**)**.** Given the system orders n_a and n_b of a PWARX model and a set of N data points $\mathcal{D} = \{(x_k, y_k)\}_{k=1}^N$ generated therefrom, estimate the number of submodels s, the parameter vectors $\{\theta_i\}_{i=1}^s$, and polyhedral partitions $\{\mathcal{X}_i\}_{i=1}^s$.

According to the definition of PWARX systems and Problem 1, the identification of PWARX model incorporates the estimation of both parameters of each submodel and their polyhedral partitions. To the best of our knowledge, the state-of-art identification approaches follow a general procedure with two stages, i.e., i) identifying the parameters of each affine submodel and ii) estimating polyhedral partitions of the input-state space [37–41]. For the first stage, tremendous methods have been presented in previous works to identify the parameters of each affine submodel. In [40], the parameters of each submodel are computed through a clustering technique based on k-means. Meanwhile, the confidence of each sample is introduced and exploited in order to improve the performance of the technique. A similar

algorithm proposed by [39] accomplishes the parameter estimation via employing a clustering method based on a Gaussian mixture model. By introducing kernel function-based weighted least squares (WLS), a recursive identification algorithm is developed in [38] for parameter identification. Conditions on the input signal and the PWARX system are imposed to guarantee the almost sure convergence of the WLS estimates. In [37], a convex framework based on ℓ_1-regularization is proposed to accomplish the identification of submodel parameters. However, unlike the first stage, the SVM algorithm is dominantly employed to estimate the partitions of submodels in the papers mentioned above [37–41]. The reason is that the SVM algorithm is capable of estimating the hyperplanes precisely. Even though the SVM algorithm is effective and well-performed regarding the identification tasks, it can be numerically inefficient. In SVM algorithm, all the data points are required to be labeled and classified before estimating the polyhedral partitions in these algorithms, which is generally expensive and time-consuming, especially for large data sets [24]. Besides, computing feasible partitions through the SVM algorithm may suffer high computational complexity [42] in this case. Taking into consideration these limitations of the SVM algorithm, recent literature propose various extensions [46–48], which, however, have not been applied in the PWA identification. Instead, the authors in [49] propose a linear multi-category discrimination method which is numerically efficient in the partitioning stage. This method, nonetheless, results in low accuracy while dealing with small training sets due to adopting the average stochastic gradient descent. Moreover, the large quantity of coefficients that required to be assigned and tuned in priority is also a drawback.

Another challenge in the PWARX identification is to specify a proper number of affine submodels to guarantee the overall accuracy. The trade-off between accuracy, generalization, and model complexity is necessary to be taken into account. As depicted in Problem 1, it is a challenge to identify all these parameters simultaneously with only the set of N data points. Therefore, the majority of the methods in the literature assume some quantities to be known in advance, particularly the number of subsystems [18,39,40]. However, it is obvious that the assumption of knowing the subsystem number is a critical limitation for the identification approaches. In the last decades, most works of PWARX system identification abandon this assumption and frontally tackle this problem. A fuzzy clustering validation-based algorithm was proposed in [50] to obtain the optimal submodel number. In [49], the optimal number is chosen by means of cross-validation, and the number of submodels corresponding to the largest best fit rate (BFR) is set to be the optimum. Specifically, the BFR is introduced as an index to assess model quality. In these methods, an algorithm to determine the optimal number of submodels needs to be specifically designed and implemented ahead of identification. As an alternative, the paper [51] tackles this issue by observing the distribution of the data points and fixing the number manually, which is subjective and inapplicable for general identification problems.

In summary, the assumption of knowing the number of subsystems in advance can be the major limitation of most existing approaches. Furthermore, the state-of-art identification approaches for PWARX systems lack the novelty in the polyhedral partition estimation process. The traditional linear discrimination methods are directly employed after gaining the parameters of the subsystem, especially the SVM algorithm. Whereas, the traditional SVM algorithm is expensive and time-consuming while dealing with a large data set. The motivation of proposing our identification approach is to overcome these limitations.

2.2. The Identification Approach

The identification approach consists of the cluster-based algorithm and the self-training SVM algorithm. The initial conditions are acquired by implementing the cluster-based algorithm. Then, the modified self-training SVM algorithm estimates the submodel parameters and polyhedral partitions via the initial conditions. In addition, the coefficients that are required to be tuned are discussed in the section as well. The overview of the approach is illustrated through the following block diagram.

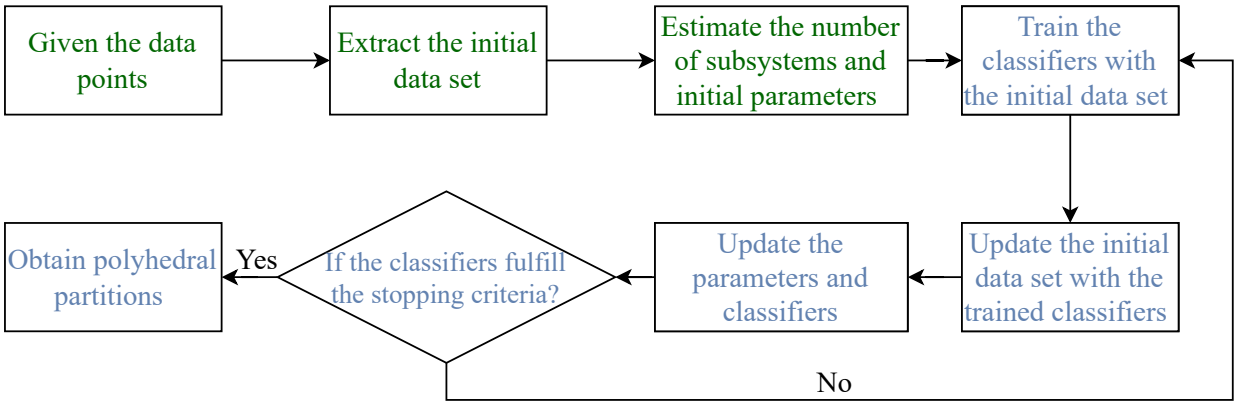


Figure 2.1.: The overview of the proposed approach. The green and blue blocks indicate the cluster-based algorithm and modified self-training SVM algorithm, respectively.

2.2.1. Cluster-based Algorithm

The cluster-based algorithm aims at obtaining the number of submodels, the initial labeled data set, and the initial parameters. It provides a method for initialization and is generally applicable in other identification approaches for PWA systems that require initialization as well, e.g., Bayesian-based methods, clustering-based methods [26]. In the algorithm, we introduce two original concepts: *local cluster* and *sub local cluster* for splitting the data set into clusters. Based on these concepts, two cost functions are designed to accomplish the task. Particularly, one of them is utilized to estimate the number of submodels.

Before presenting the cluster-based algorithm, it is necessary to introduce the concept of *local cluster*.

Definition 1 (Local Cluster)**.** A *local cluster* $\mathcal{L}$ of a center $(x_c, y_c) \in \mathcal{D}$ is built by collecting the data points (x_k, y_k) in the hypersphere of radius δ. i.e.,

$$\{(x_k, y_k) \mid d(x_c, x_k) < \delta, \forall (x_k, y_k) \in \mathcal{D}\} \tag{2.5}$$

Moreover, an LC is said to be pure, if it only possesses data points belonging to the same submodel. Otherwise, it is said to be a mixed LC.

Definition 1 is inspired by [40] where a similar concept, local data set, is introduced to select a fixed number of data points based on Euclidean norm. In this article, as an alternative, we adopt the standardized Euclidean distance in the definition of LC due to its advantages in obtaining pure LCs. This advantage is revealed through an illustrative example in Section

2.3. Meanwhile, Definition 1 indicates that the LCs are composed of the neighboring points. This characteristic is also beneficial to obtain pure LCs according to the formulation of the PWARX model.

Note that the LCs are extracted from the data set $\mathcal{D}$ iteratively and we denote the n-th LCs as $\mathcal{L}_n$. Additionally, we denote each LC as $\mathcal{L}_n = \{(\check{x}_i, \check{y}_i)\}_{i=1}^{c_n}$ where c_n is the number of data points in corresponding LC. At each iteration, an LC is extracted with respect to a randomly selected center and fixed radius δ. Hence, the data set $\mathcal{D}$ is updated as $\mathcal{D} \setminus \mathcal{L}_n$ from which we extract the next LC. This process terminates until $\mathcal{D} = \emptyset$. Clearly, the LCs satisfy $\bigcup \mathcal{L}_n = \mathcal{D}$ and $\mathcal{L}_n \bigcap \mathcal{L}_m = \emptyset, \forall n \neq m$. As the process is terminated, the data points from $\mathcal{D}$ are allocated into a group of LCs.

Clearly, the target of proposing the *local cluster* is to separate the data set into small subsets such that there are pure LCs for the next stage. Therefore, it is easy to find out that the LC can also be defined based on the number of data points in a cluster instead of the radius of a cluster, which is adopted in [40]. However, to obtain the LCs whose definition is based on the number of data points, the sort of the distance between the center and other points is necessary. Consider that the LCs are extracted from the data set in sequence until the data set is empty, we choose Definition 1 for easier implementation. Naturally, the easy implementation also pay the price, i.e., the choice of the radius δ. In Section 2.2.3, we discuss the choice of δ and illustrate the relationship between the performance of the proposed approach and the value of δ in the simulation (Section 2.3).

After the extraction process is terminated, a group of LCs is obtained. To acquire the initial labeled data set from the group, the concept of a set of *sub local clusters* (sub LCs) is introduced as follows.

Definition 2 (Sub LCs)**.** A set of sub LCs are s pure LCs with respect to each submodel in the PWARX model (2.2), i.e.,

$$\{\hat{\mathcal{L}}_i | \hat{\mathcal{L}}_i \subseteq \mathcal{X}_i, i = 1, 2, \ldots, s\}. \tag{2.6}$$

The set of sub LCs plays a fundamental role in the overall identification approach. It is assigned to be the initial labeled data set based on which the modified self-training SVM algorithm can be implemented. Thus the value of δ is required to be properly selected such that a set of sub LCs does exist. Detailed discussion on the choice of δ is presented in Section 2.2.3. Note that there could exist several sets of sub LCs and the pure LCs in one set of sub LCs may not be unique. In the cluster-based algorithm, we do not aim at selecting all sets of sub LCs but focus on extracting any one of them. To this end, we design the following cost function to extract a set of sub LCs.

$$J(\mathcal{L}_n) = R(\mathcal{L}_n) + \alpha S(\mathcal{L}_n). \tag{2.7}$$

In cost function (2.7), $R(\mathcal{L}_n)$ is the residual component designed to distinguish the pure LCs. Meanwhile, $S(\mathcal{L}_n)$ is presented as the similarity component for discriminating pure LCs from different submodels to comprise a set of sub LCs. α is the regularization coefficient to adjust the weights of the two components.

By the cost function (2.7), we can discriminate a set of sub LCs from $\mathcal{L}_n$ by solving the optimization problem $\min J(\mathcal{L}_n)$ iteratively. At each iteration, every $J(\mathcal{L}_n)$ is calculated and the minimum value corresponds to one element of a set of sub LC. Note that the results hold true only when the iteration time is less or equal to s. Now, we introduce the components of $J(\mathcal{L}_n)$ in detail.

Residual Component $R(\mathcal{L}_n)$

In cost function (2.7), the residual component $R(\mathcal{L}_n)$ for distinguishing pure LCs is given by

$$R(\mathcal{L}_n) = ||\mathcal{Y}_n - \Phi_n \check{\theta}_n||_\infty \tag{2.8}$$

where

$$\begin{aligned} &\check{\theta}_n = (\Phi_n^\mathsf{T}\Phi_n)^{-1}\Phi_n^\mathsf{T}\mathcal{Y}_n, \\ &\Phi_n = \begin{bmatrix} \check{x}_1 & \check{x}_2 & \dots & \check{x}_{c_n} \\ 1 & 1 & \dots & 1 \end{bmatrix}^\mathsf{T}, \quad \mathcal{Y}_n = [\check{y}_1 \quad \check{y}_2 \ \dots \ \check{y}_{c_n}]^\mathsf{T}. \end{aligned} \tag{2.9}$$

Here, the cost function stands for the infinity norm of residuals for each data point in $\mathcal{L}_n$. We elaborate the meaning of $R(\mathcal{L}_n)$ from the perspective of *outliers*. An outlier in a set of data is defined to be an observation that appears to be inconsistent with the remainder of that set of data [52]. Clearly, the data points in pure LC is generated by a linear model without outliers. Conversely, the mixed LC is a data set with some outliers, which are the data points from other submodels. Therefore, the residual used to detect outliers [52] can be imposed to distinguish pure LCs.

If a linear regression model is built with a data set without outliers, the residuals are assumed to have a normal distribution with a mean 0 [53]. Apparently, an LC is more likely to be pure if every residual has an absolute value closer to 0. Accordingly, we build a linear regression model for every LC and compute the corresponding value of $R(\mathcal{L}_n)$. If there is no existence of outliers in an LC, the corresponding $R(\mathcal{L}_n)$ is close to 0. Any outlier in an LC causes a greater value of $R(\mathcal{L}_n)$. Therefore, the cost function $R(\mathcal{L}_n)$ can be utilized as a criterion to distinguish pure LCs.

Similarity Component $S(\mathcal{L}_n)$

The similarity component $S(\mathcal{L}_n)$ is designed as follows to discriminate pure LCs from different submodels.

$$S(\mathcal{L}_n) = \begin{cases} 0 & \text{if } \hat{c} = 0, \\ 1 - \tanh(\min\limits_{i=1,\dots,\hat{c}}(||\check{\theta}_n - \hat{\theta}_i||_2)) & \text{if } \hat{c} > 0, \end{cases} \tag{2.10}$$

where $\hat{c}$ and $\hat{\theta}_i$ are the number of discriminated sub LCs and their corresponding parameter, respectively. Clearly, $S(\mathcal{L}_n)$ is updated with respect to $\hat{c}$. While discriminating the first sub LC ($\hat{c} = 0$), $S(\mathcal{L}_n)$ is dropped from the cost function J due to the fact that the first extracted pure LC must be a sub LC. Throughout the later procedure, $S(\mathcal{L}_n)$ is activated to obtain the rest members of a set of sub LCs. At each iteration, if one $\mathcal{L}_n$ and either of sub LC from $\{\hat{\mathcal{L}}_i\}_{i=1}^{\hat{c}}$ are from identical submodel, the corresponding $S(\mathcal{L}_n)$ value is close to 1. On the contrary, one $\mathcal{L}_n$ will correspond to a small $S(\mathcal{L}_n)$ if this LC is from a distinct submodel with any sub LCs in $\{\hat{\mathcal{L}}_i\}_{i=1}^{\hat{c}}$. Consequently, we can employ $S(\mathcal{L}_n)$ as a criterion to discriminate LCs from different submodels. Note that the scale of the parameters can be disparate in different local clusters. Thus tanh is adopted to normalize and re-scale the discrepancies. Moreover, the adoption of tanh ensures the cost function to be non-negative. The following figure shows the roles of the cost function terms in the selection of a set of sub LC intuitively.

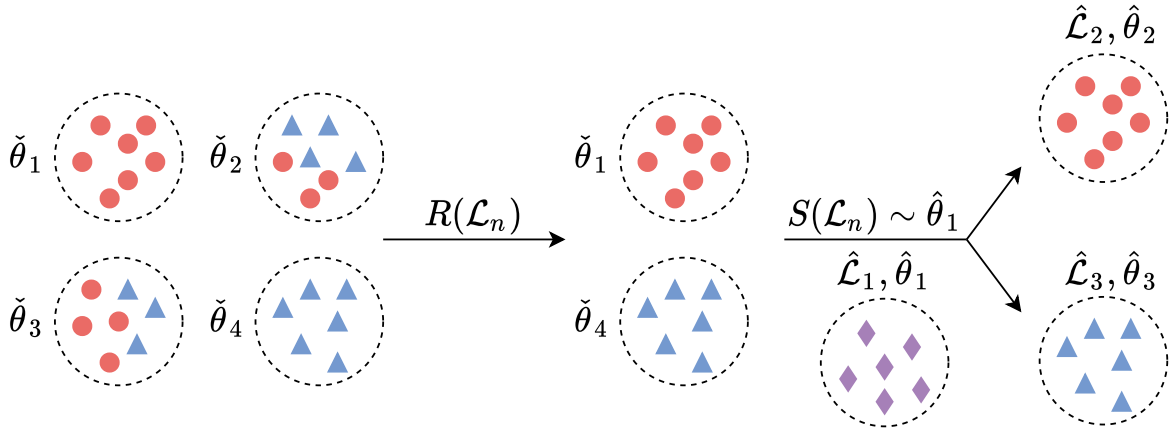


Figure 2.2.: The roles of the $R(\mathcal{L}_n)$ and $S(\mathcal{L}_n)$ in the extraction of a set of sub LCs. The first sub LC $\hat{\mathcal{L}}_1$ has been extracted. The term $R(\mathcal{L}_n)$ distinguishes the pure LCs and the term $S(\mathcal{L}_n)$ discriminates that these pure LCs are from different submodels.

It is worth noticing that the residual component $R(\mathcal{L}_n)$ of each LC is invariant at each iteration during the selection of sub LCs. It follows that the value of $R(\mathcal{L}_n)$ is only related to the structure of each LC. Therefore, we can compute $R(\mathcal{L}_n)$ of each LC in advance and only update $J(\mathcal{L}_n)$ with new $S(\mathcal{L}_n)$ at each iteration.

Remark 1. In the cluster-based algorithm, the center of each LC, i.e., (x_n, y_n), is selected randomly from the data set $\mathcal{D}$. Although the choice of (x_n, y_n) affects the number of mixed and pure local clusters, the approach will still be well-performed as long as there exists at least one set of sub LCs in all local clusters. Even if all LCs are mixed due to the unfortunate choices of (x_n, y_n), we can avoid this situation effortlessly by adjusting ρ of LCs. Therefore, the random choices of (x_n, y_n) do not lead to poor performance of the proposed approach.

Apart from the sub LCs, the number of the submodels s can be obtained by recording the value of $\min J(\mathcal{L}_n)$ at each iteration. We denote the value of $\min J(\mathcal{L}_n)$ at the j-th iteration as J_i. Considering the case when a set of s sub LCs has been obtained after the s-th iteration. If we continue to solve the minimization problem, the value of J_{s+1} increases immediately. No matter the $(s+1)$-th LC is either a mixed one or from a repetitive submodel, leading to an abrupt increment of $R(\mathcal{L}_n)$ or $S(\mathcal{L}_n)$. Then, the cost function value of later LC fluctuates in a relatively high value. Thus, the submodel number can be estimated with the following cost function.

$$s = \arg \min_{i=2,\ldots,m-1} (\mathrm{var}(J_1, \ldots, J_{i-1}) + \mathrm{var}(J_{i+1}, \ldots J_m)) \tag{2.11}$$

where m is the iteration times and assigned as the upper bound of the number of submodels. The upper bound is given in advance and limits the number of submodels for the identification. A higher number of submodels s results in a more accurate description of the PWA model. However, this may cause poor generalization to the data not used in the identification phase and increase the computational burden of the proposed algorithm. Therefore, we impose an upper bound m and estimate the optimal submodel number inside the boundary.

Following the estimation of s, a set of sub LCs $\hat{\mathcal{L}}_i$ and the corresponding parameter $\hat{\theta}_i$ are picked up. Later, we assign the label ℓ_i to every data points in $\hat{\mathcal{L}}_i$ for $i = 1 \ldots s$. Consequently, the initial labeled data set is obtained by assembling the labelled sub LCs. It

is denoted as $\mathcal{D}_l = \{(\mathbf{x}_k, \ell_k)\}_{k=1}^{\hat{n}}$ where $\mathbf{x}_k = [x_k^\mathsf{T}\ y_k]^\mathsf{T}$ and $\hat{n}$ is the number of labeled data points.

To conclude, the number of submodels s, the initial parameters $\{\hat{\theta}_i\}_{i=1}^{s}$, and the initial labeled data set $\mathcal{D}_l$ are procured through the cluster-based algorithm. The algorithm is summarized in Algorithm 1.

Algorithm 1 Cluster-based algorithm

Input:

Data set $\mathcal{D} = \{(x_k, y_k)\}_{k=1}^{N}$, radius of LCs δ, upper bound m

Output:

Number of submodels s, initial labeled data set $\mathcal{D}_l$ and initial parameters $\{\hat{\theta}_i\}_{i=1}^{s}$

1: Set $n = 1$, $\hat{c} = 0$
2: **repeat**
3: Select a data point randomly from $\mathcal{D}$ and denoted as (x_n, y_n)
4: Build $\mathcal{L}_n$ by Definition 1.
5: $\mathcal{D} \leftarrow \mathcal{D} \setminus \mathcal{L}_n$
6: $n \leftarrow n + 1$
7: **until** $\mathcal{D} = \emptyset$
8: Calculate the parameter $\breve{\theta}_n$ of each $\mathcal{L}_n$ with the equation (2.9)
9: Calculate $R(\mathcal{L}_n)$ of each $\mathcal{L}_n$ with the equation (2.8)
10: **for** $i = 1$ to m **do**
11: Calculate $S(\mathcal{L}_n)$ of each $\mathcal{L}_n$ with the equation (2.10)
12: Calculate $J(\mathcal{L}_n)$ of each $\mathcal{L}_n$ with the equation (2.7)
13: $p \leftarrow \arg\ \min_n J(\mathcal{L}_n)$
14: $J_i \leftarrow J(\mathcal{L}_p)$, $\hat{\mathcal{L}}_i \leftarrow \mathcal{L}_p$, $\hat{\theta}_i \leftarrow \breve{\theta}_p$, $\hat{c} \leftarrow i$
15: Update $S(\mathcal{L}_n)$ with $\hat{\theta}_i$ and $\hat{c}$
16: **end for**
17: Calculate the number of submodels s with the equation (2.11)
18: Assign labels $\ell_1 \dots \ell_s$ to every data points in $\hat{\mathcal{L}}_1 \dots \hat{\mathcal{L}}_s$ and compose the labeled data set $\mathcal{D}_l$
19: **return** s, $\mathcal{D}_l$, $\{\hat{\theta}_i\}_{i=1}^{s}$

2.2.2. Modified Self-Training SVM Algorithm

Armed with initial conditions obtained by the cluster-based algorithm, the modified self-training SVM algorithm is introduced to estimate the parameters of submodels and corresponding the polyhedral partitions. By imposing a customized selection strategy, the modified self-training SVM algorithm is capable of estimating the polyhedral partitions precisely with partially-labeled data set. Moreover, the computational complexity of the modified self-training SVM algorithm is analyzed and compared with the traditional SVM algorithm to justify its lower computational complexity.

The modified self-training SVM algorithm is described in Algorithm 2. The initial training set $\mathcal{D}_T$ is a collection of the center corresponding to every sub LC in $\mathcal{D}_l$. At each iteration, the classifiers are trained with $\mathcal{D}_T$. Note that the kernel of the SVM for training is linear due to the definition of the polyhedral partition. Then $\mathcal{D}_T$ and $\mathcal{D}_l$ are updated and $\{\hat{\theta}_i\}_{i=1}^{s}$

are re-calculated with the updated $\mathcal{D}_l$. With the iterative procedures, the trained classifiers gradually converge to the optimal hyperplanes and the criterion for terminating the iteration is designed as follows. For a given tolerance ρ, if there holds

$$\max_{i=1,2,\ldots,q} \left(f(w_i^{(j)}, \xi_i^{(j)}) - f(w_i^{(j-1)}, \xi_i^{(j-1)}) \right) < \rho, \tag{2.12}$$

where

$$f(w_i, \xi_i) = \frac{1}{2}||w_i||^2 + C \sum_{k=1}^{\#\mathcal{D}_T} \xi_{k,i}, \tag{2.13}$$

then the iteration is terminated. In the equation (2.12), j, q and C indicates the index of iteration, the number of classifiers, and the penalty parameters, respectively. $\xi_{k,i}$ implies the linear penalty ξ of every sample in $\mathcal{D}_T$ w.r.t the classifier (w_i, b_i). Note that the termination condition is built on Theorem 1 that is proved to be true. Afterwards, the classifiers are trained for n_{re} times without discarding process (Step 19), where n_{re} is assigned to a small scalar by users.

In Algorithm 2, we design the strategy to update the training set based on the fact that only the SVs contribute to the optimal hyperplane. Consequently, data points that are not SVs could be eliminated without affecting the trained hyperplanes. To sum up, for a data set $\mathcal{D}$, the same optimal hyperplanes can be obtained with a small training set $\mathcal{D}_T$ as long as the SVs of $\mathcal{D}$ are involved in $\mathcal{D}_T$. In other words, Algorithm 2 converges to optimal hyperplanes in this circumstance. Therefore, our update strategy is designed to collect the samples which are more likely to be SVs into $\mathcal{D}_T$. The update strategy is composed of the union process and discard process.

As pointed out in Appendix A, the geometric margins between SVs and the classifiers of $\mathcal{D}$ are smaller than any other samples in $\mathcal{D}$. Thus, the data points whose geometric margin corresponding to either classifier is the minimum are collected as $\mathcal{M}$ in Algorithm 2. Note that these data points are selected from $\mathcal{D} \setminus \mathcal{D}_T$. Clearly, the data points collected based on the equation (2.14) is more likely to be the SVs of $\mathcal{D}$.

The design of the update strategy also improves the convergence speed of Algorithm 2. According to the proof of Theorem 1, the data points from $\mathcal{M}$ formulate the new constraints in the optimization problem of SVM algorithm. Note that the trained classifiers remain constant if the new constraints have been fulfilled. Clearly, this situation happens when all data points in $\mathcal{M}$ are correctly classified and their geometric margins w.r.t the current classifiers are greater than 1. Therefore, the update strategy based on minimum geometric margins ensures that the classifiers do not remain constant at each iteration in the early stage. In summary, by avoiding the classifiers from being constant, the update strategy improves the convergence rate of the proposed algorithm.

The update strategy of $\mathcal{D}_T$ also includes the procedure of discarding samples from D_T. The non-SVs of D_T are discarded after training the classifiers at each iteration. This procedure is designed to increase the computational efficiency of the algorithm. Moreover, the discarding procedure will not affect the convergence of the modified self-training algorithm, which is described in the appendix.

In the modified self-training SVM algorithm, the parameters of submodels are updated with the labeled data set $\mathcal{D}_l$ instead of using the training set $\mathcal{D}_T$ at each iteration. The reason is the parameters calculated with more samples are more plausible in the regression analysis. Compare to the training set, the labeled data set is updated without the procedure of discarding and contains more samples for computing the parameters.

Algorithm 2 Modified self-training SVM algorithm

Input: Initial labeled data set $\mathcal{D}_l$, initial parameters $\{\hat{\theta}_i\}_{i=1}^s$, tolerance ρ, Penalty coefficient C.

Output: Hyperplanes group $\mathcal{G}$ and estimated submodel parameters $\{\theta_i\}_{i=1}^s$

1: Let $j = 0$
2: Extract initial training set $\mathcal{D}_T$ from $\mathcal{D}_l$
3: **repeat**
4: Train a classifier group $\mathcal{G}^{(j)} = \{(w_i^{(j)}, b_i^{(j)})\}_{i=1}^q$ with training data set $\mathcal{D}_T$
5: Let $\mathcal{D}_T \leftarrow \mathcal{D}_T^{SV}$
6: Let

$$\mathcal{M} = \bigcup_{i=1}^{q} \arg\min_{\mathbf{x}\in\mathcal{D}\setminus\mathcal{D}_T} \gamma(w_i^{(j)}, b_i^{(j)}, \mathbf{x}) \tag{2.14}$$

7: Denote $\mathcal{M} = \{(x'_k, y'_k)\}_{k=1}^q$
8: **for** $k = 1$ to q **do**
9: Let

$$n = \arg\min_{i=1,\dots,s} ||y'_k - \hat{\theta}_i x'_k|| \tag{2.15}$$

10: $\ell'_k \leftarrow \ell_n$
11: **end for**
12: Denote $\mathcal{M}_\ell = \{(\mathbf{x}'_k, \ell'_k)\}_{k=1}^q$ where $\mathbf{x}'_k = [x_k'^{\mathsf{T}} \quad y'_k]^{\mathsf{T}}$
13: $\mathcal{D}_T \leftarrow \mathcal{D}_T^{SV} \cup \mathcal{M}_\ell$, $\mathcal{D}_l \leftarrow \mathcal{D}_l \cup \mathcal{M}_\ell$
14: Recomputed $\{\hat{\theta}_i\}_{i=1}^s$ with $\mathcal{D}_l$ and the equation (2.9)
15: $j \leftarrow j + 1$
16: **until** $\max_{i=1,2,\dots,q} \left(f(w_i^{(j)}, \xi_i^{(j)}) - f(w_i^{(j-1)}, \xi_i^{(j-1)})\right) < \rho$
17: **for** j to $j + n_{re}$ **do**
18: Step 4 and Step 6-18
19: **end for**
20: $\mathcal{G} = \{(w_i^{(j)}, b_i^{(j)})\}_{i=1}^q$, $\{\theta_i\}_{i=1}^s = \{\hat{\theta}_i\}_{i=1}^s$
21: **return** Hyperplane group $\mathcal{G}$ and estimated submodel parameters $\{\theta_i\}_{i=1}^s$

For the termination condition (2.12), the employment of $f(w_i, \xi_i)$ is based on the following theorem which is inspired by [54].

Theorem 1. For $f(w_i, \xi_i)$ defined in (2.13), we have

$$f(w_i^{(j)}, \xi_i^{(j)}) \geq f(w_i^{(j-1)}, \xi_i^{(j-1)}) \tag{2.16}$$

The proof of the theorem is as follows:

Proof. According to the modified self-training SVM Algorithm, we assume $(w^{(1)}, b^{(1)})$ is the classifier at the first iteration, which is the solution of the optimization problem with initial training data set $\mathcal{D}_T$.

$$\begin{aligned} &\min \ \frac{1}{2}||w^{(1)}||^2 + C\sum_{k=1}^{N_1} \xi_k \\ &s.t. \ \ell_k((w^{(1)})^{\mathsf{T}}\mathbf{x}_k + b^{(1)}) \geq 1 - \xi_k, k = 1, \cdots, N_1 \end{aligned} \tag{2.17}$$

where N_1 is the number of data points in $\mathcal{D}_T$, $\mathbf{x}_k$ is the data point, ℓ_k is the corresponding label. The label ℓ_k of each data point remains constant over the iterations.

Clearly, the solution of the optimization problem (2.17) is identical to the solution of the following problem based on the property of support vectors.

$$\begin{aligned} &\min \frac{1}{2}||w^{(1)}||^2 + C\sum_{k=1}^{N_{sv}} \xi_k \\ &\text{s.t. } \ell_k^{sv}((w^{(1)})^{\mathsf{T}}\mathbf{x}_k^{sv} + b^{(1)}) \geq 1 - \xi_k, k = 1, \cdots, N_{sv} \end{aligned} \tag{2.18}$$

where N_{sv} is the number of support vectors in $\mathcal{D}_T$, $\mathbf{x}_k^{sv}$ is one support vector of $\mathcal{D}_T$, ℓ_k^{sv} is the corresponding label.

For the second iteration of Algorithm 2, the updated training data set D_T is composed of support vectors $\mathbf{x}_k^{sv}$ and the data points from $\mathcal{M}$. We can find that $(w^{(2)}, b^{(2)})$ is the solution of the following optimization problem.

$$\begin{aligned} &\min \frac{1}{2}||w^{(2)}||^2 + C\sum_{k=1}^{N_{sv}} \xi_k + C\sum_{i=1}^{q} \xi_i \\ &\text{s.t. } \ell_k^{sv}((w^{(2)})^{\mathsf{T}}\mathbf{x}_k^{sv} + b^{(2)}) \geq 1 - \xi_k, k = 1, \cdots, N_{sv} \\ &\quad \ell_i'((w^{(2)})^{\mathsf{T}}\mathbf{x}_i' + b^{(2)}) \geq 1 - \xi_i, i = 1, \cdots, q \end{aligned} \tag{2.19}$$

where (x_k', ℓ_k') is the sample of $\mathcal{M}$. The classifier is trained with the updated D_T. From the equations (2.18) and (2.19), $(w^{(2)}, \xi^{(2)}, b^{(2)})$ is a feasible solution of the equation (2.17). Since $(w^{(1)}, \xi^{(1)}, b^{(1)})$ is an optimal solution of the equation (2.17), we have

$$f(w^{(1)}, \xi^{(1)}) \leq f(w^{(2)}, \xi^{(2)}) \tag{2.20}$$

where

$$f(w, \xi) = \frac{1}{2}||w||^2 + C\sum_{k=1}^{N} \xi_k. \tag{2.21}$$

It yields that

$$f(w^{(j-1)}, \xi^{(j-1)}) \leq f(w^{(j)}, \xi^{(j)}). \tag{2.22}$$

For a given labeled data set $\mathcal{D}$, the value of $f(w, \xi)$ trained with the modified self-training SVM is bounded. The maximum value of f is obtained while (w^*, ξ^*, b^*) is the solution of

$$\begin{aligned} &\min \frac{1}{2}||w^*||^2, \\ &\text{s.t. } \ell_k({w^*}^{\mathsf{T}}\mathbf{x}_k + b^*) \geq 1 - \xi_k^*, k = 1, \cdots, N, \end{aligned} \tag{2.23}$$

where $(\mathbf{x}_k, \ell_k)$ are the data points of $\mathcal{D}$. Thus $\frac{1}{2}||w^{(j)}||^2$ of the modified self-training SVM is convergent and monotonically non-decreasing over the iterations. □

Clearly, the proof implies the suitability of adopting the $f(w_i, \xi_i)$ as the stopping criterion and also analyzes the convergence of the modified self-training SVM algorithm. In addition, the stop ping criterion also indicates that the choice of ρ influences the efficiency and performance of our algorithm, which is presented in Section 2.2.3. It is worth noticing that the termination criterion $f(w_i, \xi_i)$ may converge to a local minimum by Theorem 1. The update

strategy only guarantees that the criterion will not converge to a local minimum in the early stage. Consequently, we design the retraining step (step 19) in the algorithm to avoid local minimum.

Compared with the traditional self-training SVM, the proposed modified self-training SVM designs a novel selection strategy for collecting the unlabeled data points. In our algorithm, the unlabeled data which are collected to augment the training set are corresponding to the minimum geometric margin instead of the highest confidence. Moreover, the predicted labels are determined by the estimated parameter of submodels in our algorithm rather than the classifiers trained at the last iteration in the original self-training. Clearly, the predicted labels in our algorithm are more plausible in the PWA system identification. Thus, the proposed modified self-training SVM is specifically designed for PWA identification and adopts the features of PWA systems sufficiently.

In summary, the estimated hyperplanes $\{(w_i^{(j)}, b_i^{(j)})\}_{i=1}^{q}$ and the estimated submodel parameters $\{\theta_i\}_{i=1}^{s}$ can be computed with the Algorithm 2 . Thus the identification of PWARX system described in Problem 1 is achieved by implementing the proposed cluster-based algorithm and modified self-training SVM algorithm.

Remark 2. The modified self-training SVM algorithm is designed for linear separable and inseparable cases by adjusting the penalty parameter C. The submodel of PWA systems is generally linear separable as stated in the identification task. Therefore, we generally choose a relatively large value of C in the algorithm corresponding to assigning a higher penalty to errors [55].

Remark 3. The modified self-training SVM algorithm includes calculating the geometric margins, computing submodel parameters with $\mathcal{D}_l$, and training the classifiers with $\mathcal{D}_T$ at each iteration. The computational complexity of the algorithm can be calculated as follows.

The complexity of calculating the geometric margins and submodel parameter is $O(r \cdot N) + O(r \cdot N_{D_l})$, where r is the number of iteration, N is the number of samples in $\mathcal{D}$, $N_{\mathcal{D}_l}$ is the number of samples in $\mathcal{D}_l$. The complexity of the SVM training is $O\left([r(n_{sv}+s)]^3\right)$, where n_{sv} is the number of support vectors at each iteration, s is the number of submodels. Therefore, the total computational complexity of modified self-training SVM is

$$O(r \cdot N) + O(r \cdot N_{D_l}) + O\left([r(n_{sv}+s)]^3\right) \tag{2.24}$$

Above complexity grows linearly with respect to the number of iterations r. The number of iteration $r \ll N$ in the algorithm. The scale of r is presented in Section 2.3. Moreover, the other coefficients in the (2.24) also fulfill $n_{sv}, s, N_{D_l} \ll N$ as well. Therefore, it is obvious that (2.24) is much smaller than the computational complexity of a normal SVM $O(N^3)$.

2.2.3. Tuning the Coefficients δ, ρ, and α

In Definition 1, δ is given in advance to define LC. Consider the LC extraction procedure in cluster-based algorithm, a lower δ implies more LCs and further, more pure LCs. However, when the noise level is not negligible, a lower δ implies fewer data points in the regression leading to poor initial parameter estimation. This causes inaccuracy in the identification procedure. On the contrary, a higher δ provides more data for regression in every LC and improves the accuracy of parameters estimations. Whereas, a higher δ denotes fewer pure LCs and fewer number of iterations for extracting sub LCs. Therefore, the choice of the

parameter δ is a trade-off between the number of pure LCs and the accuracy of regression. Before extracting LCs, the data points are normalized to adjust them to a notionally common scale. This strategy ensures the selection of δ is not determined by the PWA models. As a result, it is reasonable to choose a relatively large δ since it increases the credibility of estimated parameters and reduces the number of iterations as well. The reduction of pure LCs caused by the choice will not influence the algorithm performance as long as at least one set of sub LCs still exist. Nevertheless, too large δ that vanishes all the pure LCs is apparently unacceptable.

The parameter ρ is the terminal condition of our algorithm. Obviously, a smaller ρ leads to more iterations before converge and brings a higher BFR. However, the rise of $f(w, \xi)$ is decided by the newly joined samples and not continuous. This characteristics illustrates there is a lower bound for ρ, The accuracy of the estimation will not getting better after ρ exceeds the boundary.

Consider the parameter α, which is a regularization coefficient to adjust the weights of the residual component and the similarity component in the cost function (2.7). Too small α leads to the situation that the LCs from the same submodels are involved in the set of sub LCs. Conversely, mixed LCs can be collected into the set of sub LCs while choosing a too large α. Therefore, When a validation data set is available, a sensible idea would be to tune α with cross-validation techniques.

2.3. Simulation Studies

In the section, the effectiveness of the proposed approach is validated through three simulations. The first simulation is designed to show the superiority of adopting the standardized Euclidean distance to define the *local cluster*. The utility of the proposed approach is demonstrated through the second simulation. Finally, the performances of the existing approaches and the proposed approach are contrasted in the third simulation.

2.3.1. Comparison of Local Cluster Definitions

The main challenge of the cluster-based algorithm is to extract sub LCs from the data set. In the definition, we adopt the standardized Euclidean distance to define LCs instead of the Euclidean distance mentioned in [40]. Both of the algorithms aim at obtaining LCs but our algorithm concentrates more on the pure LCs. Utilizing the Euclidean distance may cause no existence of pure LCs in some circumstances. To illustrate this drawback, a PWARX system is given as an instance.

$$y_k = \begin{cases} [50 \;\; 0.2 \;\; 0.1]\varphi_k \text{ if } x_k \in \mathcal{X}_1, \\ [60 \;\; 0.1 \;\; 0.3]\varphi_k \text{ if } x_k \in \mathcal{X}_2, \end{cases} \tag{2.25}$$

where $\varphi_k = [x_k^\mathsf{T} \; 1]^\mathsf{T}$, $x_k = [y_{k-1} \; u_{k-1}]^\mathsf{T}$ and

$$\mathcal{X}_1 = \{x \in \mathbb{R}^2 : [0 \quad 1 \;\; 0]\varphi_k < 0\},$$
$$\mathcal{X}_2 = \{x \in \mathbb{R}^2 : [0 \;\; -1 \;\; 0]\varphi_k \leq 0\}.$$

In this simulation, the input u_k is distributed uniformly in the interval $[-0.5, 0.5]$ and the initial regression vector is $x_0 = [y_0, u_0]^\mathsf{T} = [0, 0]^\mathsf{T}$. The regression vectors generated by the

system (2.25) is shown in Figures 2.3 and 2.4, where the data points with the same color belong to an LC. As is presented in Figure 2.3, all LCs are mixed LCs if the LC is defined with Euclidean distance which is sensitive to the scaling of vectors. Obviously, this circumstance makes the cluster-based algorithm infeasible. In contrast, using the standardized Euclidean distance avoids this drawback by introducing normalization. The LCs defined with standard Euclidean distance are illustrated in Figure 2.4 and six of them are pure LCs. Clearly, our definition of LC is more suitable for our algorithm.

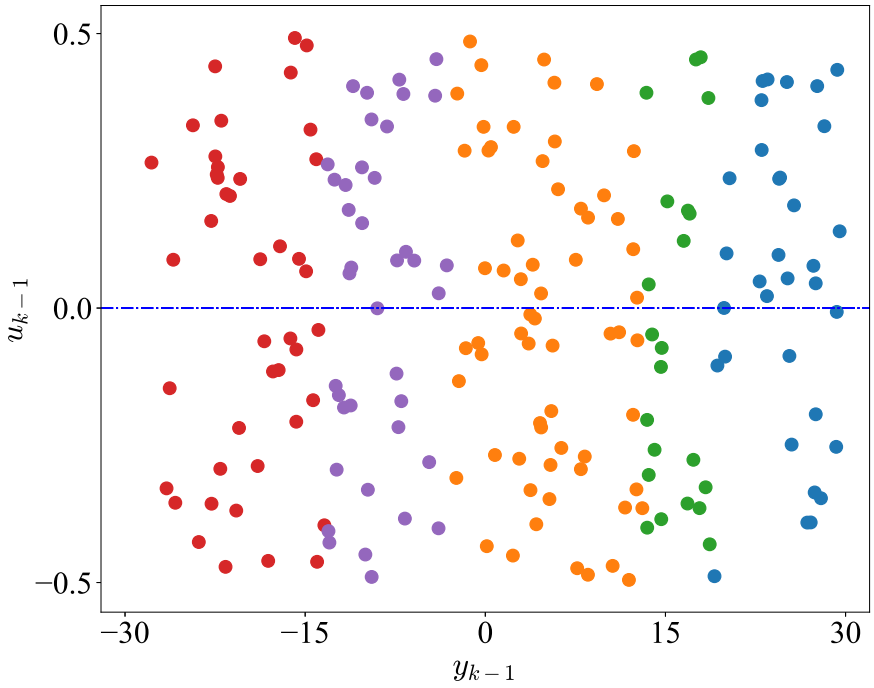


Figure 2.3.: The generated regression vectors (data points) and the hyperplane (blue line) of the system (2.25), where the LCs (data points with same color) are extracted by adopting Euclidean distance in the definition.

2.3.2. Identification of PWARX Model

In this section, a series of simulations are conducted to illustrate the effectiveness of the proposed identification approach. We consider the following single-input-single-output (SISO) PWARX model composed of three submodels, i.e, $s = 3$, with the orders $n_a = n_b = 1$.

$$y_k = \begin{cases} [-0.4 \quad 1 \quad\quad 1.5]\,\varphi_k + e_k & \text{if } x_k \in \mathcal{X}_1, \\ [0.5 \quad -1 \quad -0.5]\,\varphi_k + e_k & \text{if } x_k \in \mathcal{X}_2, \\ [-0.3 \quad 0.5 \quad -1.7]\,\varphi_k + e_k & \text{if } x_k \in \mathcal{X}_3 \end{cases} \tag{2.26}$$

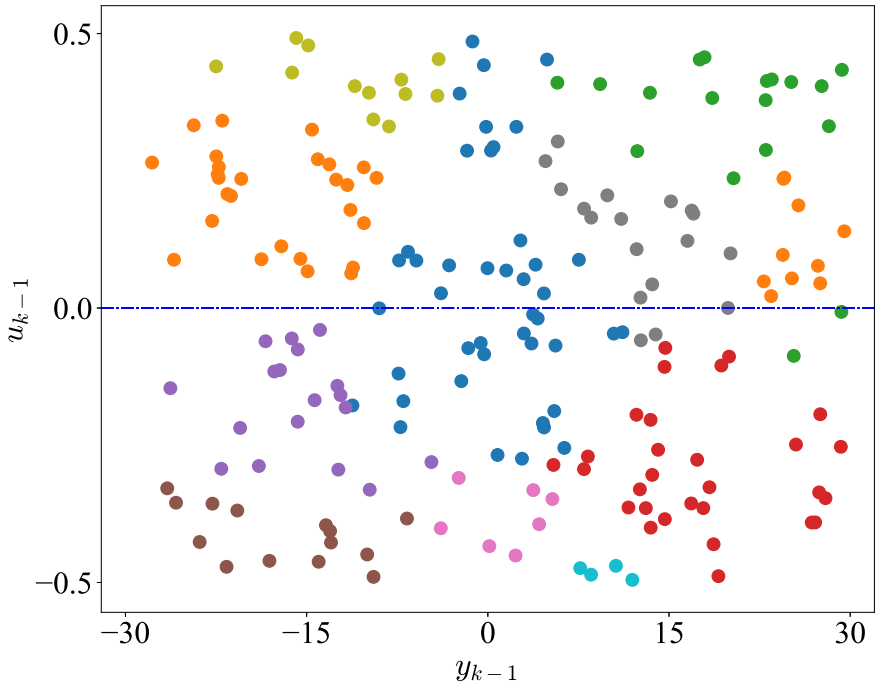


Figure 2.4.: The generated regression vectors (data points) and the hyperplane (blue line) of the system (2.25). Extracted LCs (data points with same color) by adopting standard Euclidean distance in the definition.

where $\varphi_k = [x_k^\mathsf{T}\ 1]^\mathsf{T}$, $x_k = [y_{k-1}\ u_{k-1}]^\mathsf{T}$ and

$$\begin{aligned}
\mathcal{X}_1 &= \{x \in \mathbb{R}^2 : \begin{bmatrix} 4 & -1 & 10 \end{bmatrix} \varphi_k < 0\},\\
\mathcal{X}_2 &= \{x \in \mathbb{R}^2 : \begin{bmatrix} -4 & 1 & -10 \\ 5 & 1 & -6 \end{bmatrix} \varphi_k \leq 0\},\\
\mathcal{X}_3 &= \{x \in \mathbb{R}^2 : \begin{bmatrix} -5 & -1 & 6 \end{bmatrix} \varphi_k < 0\}.
\end{aligned} \tag{2.27}$$

The PWARX model is typical and widely utilized for simulation in the existing papers [39, 50, 56, 57]. The input u_k is distributed randomly on the interval $[-2, 2]$ with $N = 1000$. The output y_k of the system is corrupted by an additive normal noise $e_k \sim \mathcal{N}(0, 0.12)$. To qualify the effect of noise, we impose the Signal-to-Noise Ratio (SNR) as follows

$$\text{SNR} = 10 \log \frac{\sum_{k=1}^{N} (y_k - e_k)^2}{\sum_{k=1}^{N} e_k^2}. \tag{2.28}$$

By implementing (2.28), the SNR of the noise e_k is 25dB in the simulation. Under aforementioned configurations, 1000 data points $\{(x_k, y_k)\}_{k=1}^{1000}$ conforming to the model (2.26) are generated and shown in Figure 2.5 in the regression vector space. Then the proposed identification approach is implemented with the coefficients $\delta = 0.1$, $C = 1000$, $\alpha = 2$, $n_{re} = 6$, and $\rho = 10^{-3}$. By running the cluster-based algorithm, the number of submodels is estimated as $s = 3$. In addition, a set of sub LCs is extracted, which is shown as stars

in Figure 2.5. The set D_T for classifier training is also shown as pentagons in the same figure. Apparently, all the sub LCs in this set satisfy Definition 2. Subsequently, the modified self-training SVM converges after 24 iterations and only 24.2% data points are classified and labeled in total. Compared with previous works, our algorithm is capable of estimating the hyperplanes precisely with a small fraction of labeled data points.

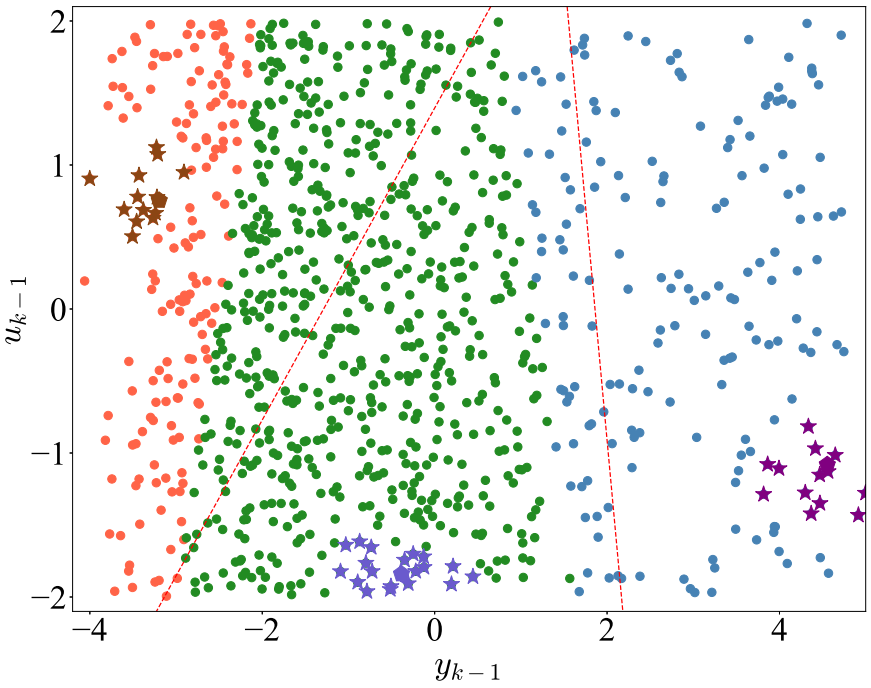


Figure 2.5.: Regression vectors (red, blue, green) generated by system (2.26), whose color represents the corresponding submodel. The sub LCs $\hat{\mathcal{L}}_i$ (pentagon points with different colors) extracted with cluster-based algorithm and hyperplanes (dashed red line) trained with initial labeled data set $\mathcal{D}_T$.

Based on the results obtained by the cluster-based algorithm, we then show the identification performance of the modified self-training SVM algorithm. As is presented in Figure 2.6, the blue and red dashed lines are the estimated and the true hyperplanes, respectively. In detail, we plot the estimated hyperplanes with respect to the iteration times in Figure 2.7. Note that w_i of the hyperplane is a vector which is donated by $w_i = \begin{bmatrix} w_i^1 & w_i^2 \end{bmatrix}^\mathsf{T}$ in the simulation. In Figure 2.7, the entries of each hyperplane are plotted with different colors and styles. As is illustrated in Figure 2.7, the estimated hyperplanes converge to the true hyperplanes $\begin{bmatrix} -4 & 1 & -10 \end{bmatrix}^\mathsf{T}$ and $\begin{bmatrix} 5 & 1 & -6 \end{bmatrix}^\mathsf{T}$ after 24 iterations. Apart from the hyperplanes, as is presented in Table 2.1, the estimated parameters of each submodel are close to the true ones. Thus the simulation results indicates the submodel parameters and hyperplanes are well-estimated. In addition, the proposed algorithm is more efficient than traditional SVM since less data points are employed.

We evaluate the performance of the proposed approach by the best fit rate (BFR) [58]

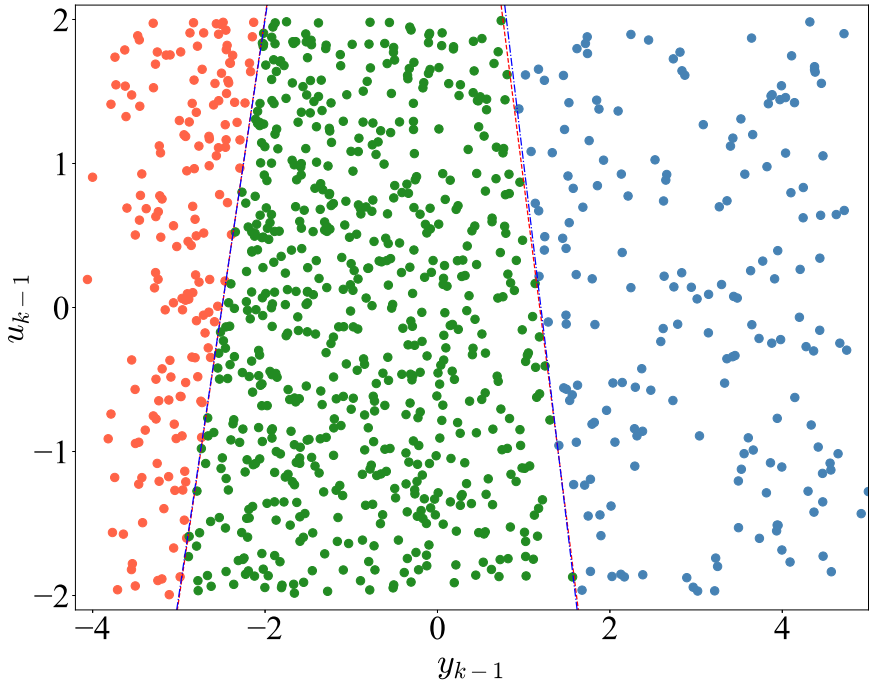


Figure 2.6.: Regression vectors (red, blue, green) generated by system (2.26), whose color represents the corresponding submodel. The true hyperplanes (blue dashed line) and the estimated hyperplanes (red dashed line) of system (2.26).

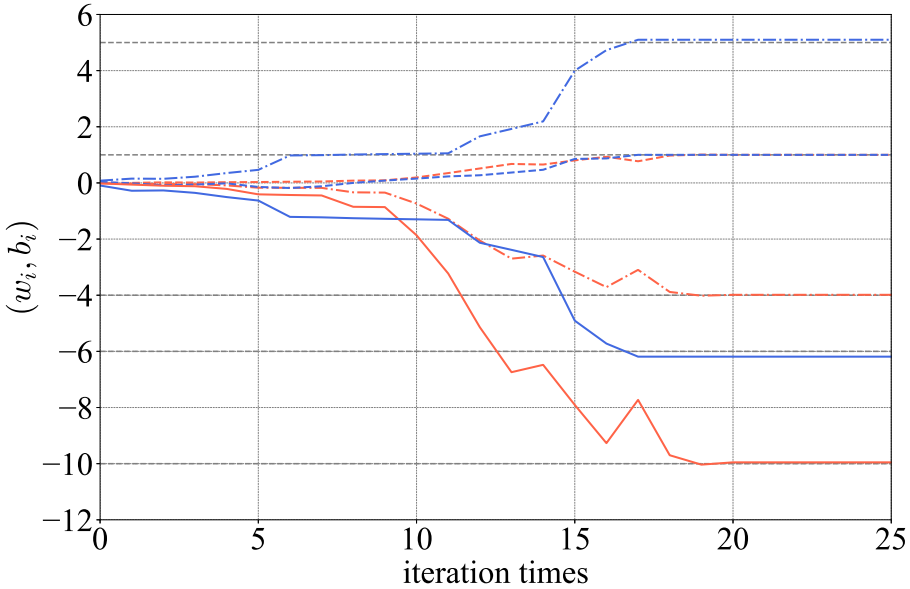


Figure 2.7.: The hyperplanes (w_1, b_1) and (w_2, b_2) with respect to the iteration times while implementing Algorithm 2. The red lines are the values of w_1^1, w_1^2, b_1 (dash-dotted line, dashed line, and solid line), respectively. The blue lines are the value of w_2^1, w_2^2, and b_2 (dash-dotted line, dashed line, and solid line), respectively.

Table 2.1.: True (θ_i) and estimated ($\hat{\theta}_i$) parameter vectors in the simulation

θ_1	$\hat{\theta}_1$	θ_2	$\hat{\theta}_2$	θ_3	$\hat{\theta}_3$
-0.4	-0.3890	0.5	0.5036	-0.3	-0.2816
1	0.9711	-1	-1.0102	0.5	0.4945
1.5	1.5801	-0.5	-0.4766	-1.7	-1.7084

which is defined as follows

$$\text{BFR} = \max\left\{1 - \frac{||y - \hat{y}||}{||y - \bar{y}\mathbf{1})||}, 0\right\} \times 100\%, \tag{2.29}$$

where $y \in \mathbb{R}^N$ is the true output sequence, $\hat{y} \in \mathbb{R}^N$ is the estimated output sequence. Additionally, $\bar{y} \in \mathbb{R}$ is the mean of the true output sequence and $\mathbf{1}$ is an all-ones vector with suitable dimension.

A validation data set containing 200 data points is generated in the same conditions as (2.26). Figure 2.8 depicts the true output y and the estimated output $\hat{y}$, as well as their error $y - \hat{y}$. For the sake of vision, 100 data points are plotted. The BFR value of the estimated PWARX system is 92.289%. The value implies the accuracy of the proposed approach is quite satisfying.

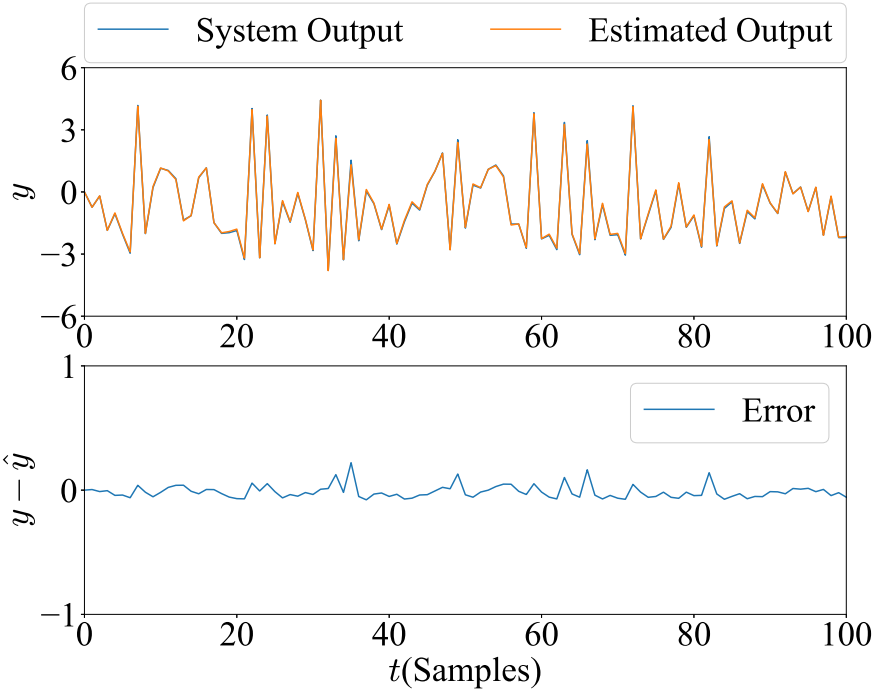


Figure 2.8.: System output y (blue line) and estimated output $\hat{y}$ (red line) of the validation data set, as well as their error $y - \hat{y}$.

The total amount of training data N_T and the number of iterations r are crucial factors to evaluate the computational complexity of our algorithm.The total amount of training

data is calculated by summing the cardinality of $\mathcal{D}_T$ at each iteration. Therefore, we record them with different ρ, N and PWA model (from [56] and [40]) in Table 2.2. The percentage of N_T/N is around 20% while $N = 1000$. The percentage is getting much smaller with respect to the increment of N. While $N = 50000$, the percentage is around 4%. The result shows that fewer points are used in the training process compared with traditional SVM. Therefore, the modified self-training SVM is more efficient than traditional SVM, especially for large data set. The scale of iteration times of two different PWA models are similar to each other, showing the efficiency of modified self-training is not highly related to the PWA model type. According to our simulations, the convergent speed is related to the choice of ρ. Generally, the algorithm convergence very fast (in 40 iterations). In addition, a smaller tolerance ρ requires more iterations to converge. The values of BFR are around 90% in various conditions, showing a satisfying performance of our algorithm even for a small training set.

Table 2.2.: Number of iterations, total amount of training samples and BFR of modified self-training SVM in various conditions

N	PWA model I [56]		PWA model II [40]	
	$\rho = 10^{-2}$	$\rho = 10^{-3}$	$\rho = 10^{-2}$	$\rho = 10^{-3}$
1000	$r = 16$	$r = 26$	$r = 18$	$r = 27$
	$N_T = 193$	$N_T = 242$	$N_T = 215$	$N_T = 287$
10000	$r = 41$	$r = 45$	$r = 38$	$r = 47$
	$N_T = 607$	$N_T = 679$	$N_T = 581$	$N_T = 712$
50000	$r = 58$	$r = 64$	$r = 52$	$r = 61$
	$N_T = 1372$	$N_T = 1907$	$N_T = 1057$	$N_T = 1812$
1000	BFR = 88.29%	BFR = 92.29%	BFR = 89.82%	BFR = 89.59%
10000	BFR = 89.04%	BFR = 93.86%	BFR = 92.02%	BFR = 91.33%
50000	BFR = 90.95%	BFR = 93.41%	BFR = 93.34%	BFR = 91.30%

The coefficients δ, ρ and α influence the performance of our algorithm greatly. Therefore, we analysis the sensitivity of δ, ρ and α with respect to BFR for the PWA system (2.26) through the same validation set used in Figure 2.8. Figs. 2.9, 2.10, and 2.11 illustrate the values of BFR with respect to different values of δ, ρ, and α, respectively. Note that, the BFR rates are relatively high with the coefficient $0.10 \leq \delta \leq 0.15$, indicating a satisfying performance of our algorithm. For $\rho \leq 10^{-3}$, the final estimate is fairly insensitive to the tolerance. Moreover, a wild range of $1.0 \leq \alpha \leq 10.0$ is allowed under the condition that the performance of our algorithm is satisfying.

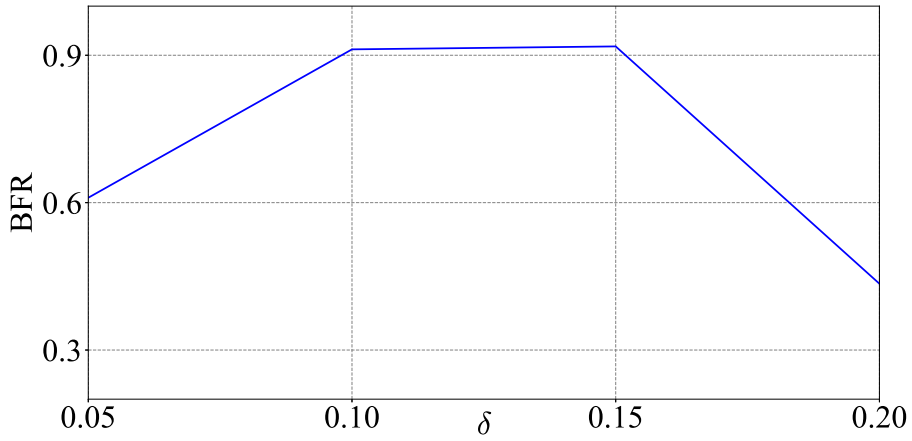


Figure 2.9.: BFR with respect to the tuning parameter δ in the PWA system (2.26)

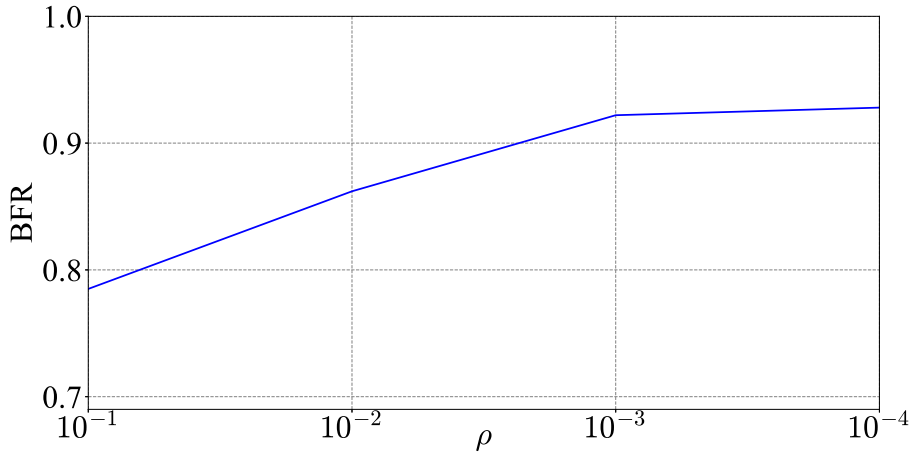


Figure 2.10.: BFR with respect to tuning parameter ρ in the PWA system (2.26)

2.3.3. Comparison with Previous Works

For comparison, the models simulated in the [49,50] are also identified with our approach by using the data points generated in the same conditions. In [50], a 3-submodel PWA system is identified with its algorithm and BFR value is 74.43%. The same regression problem is solved by our identification algorithm and the BFR equals 90.04%. Moreover, to evaluate the ability of the proposed approach, we approximate the hybrid dynamic model mentioned in [49] with the data points generated in same conditions. Likewise, the BFR value computed with our algorithm is 88.19% whereas 85.19% in [49] and 53.40% in [40] as is listed in Table 2.3. In [49], the hybrid dynamic model is approximated with a PWA system owing 12 local submodels to obtain the best BFR (85.19%). By using our approach, a 5-submodels PWA system is established with 88.19% BFR. Therefore, the proposed approach provides a better quality PWA model with fewer submodels in the approximation of hybrid dynamic model.

The simulation results indicate that the proposed approach provides a good performance for the PWA system identification, even with a quite low signal/noise ratio. Moreover,

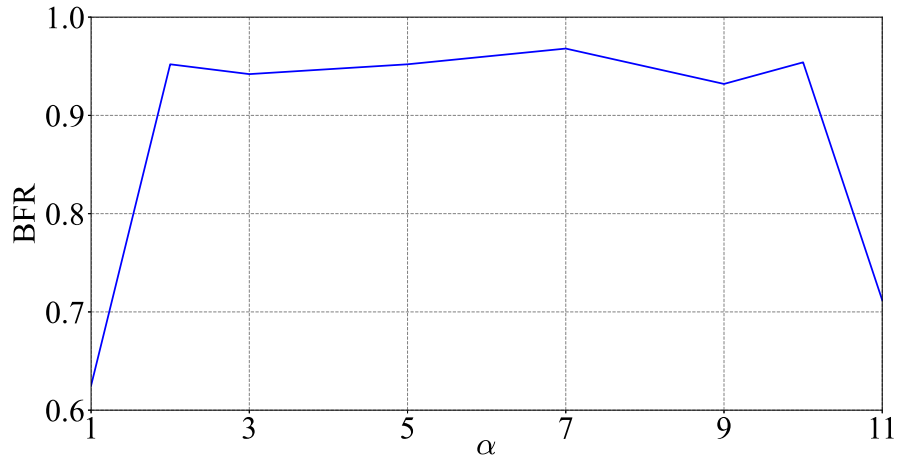


Figure 2.11.: BFR with respect to tuning parameter α in the PWA system (2.26)

Table 2.3.: Comparison of BFR values between the proposed approach and other recent works

Approaches	hybrid dynamic model [49]
Proposed approach	88.37%
Breschi [49]	85.19%
Ferrari-Trecate [40]	53.40%

fewer data points are required to be labeled and used for training by using the modified self-training algorithm. The total numbers of training samples in various conditions are listed in the simulation. This advantage increases the efficiency of modified self-training algorithm compared to the traditional SVM algorithm. Moreover, compared with the very recent works [49], the simulation results imply that the proposed approach achieves satisfying performance for the approximation of hybrid dynamic systems. As a matter of fact, our algorithm approximates the model even preciser with fewer number of submodels.

2.4. Summary

In this chapter, we focus on the identification of a representative hybrid system, PWARX system. The novel efficient approach designed for the identification of PWARX systems is presented. The approach consists of the cluster-based algorithm and modified self-training SVM algorithm. The number of submodels, the parameter of each submodel, and the polyhedral partition of each submodel are estimated via the approach.

The cluster-based algorithm is constructed on the *local cluster* and *sub local cluster*, which are defined with the standardized Euclidean distance. The essence of the algorithm is to search a set of sub LCs which contains the information of every submodel. The procedure

is achieved via the designed cost function. The initial labeled data set, initial parameters, and the number of submodels, are then determined from the extracted set of sub LCs. Subsequently, the modified self-training SVM algorithm is designed to identify the polyhedral partitions by regarding the initial labeled data set, initial parameters, and the number of submodels as prerequisites. In the modified self-training SVM algorithm, the hyperplanes are updated with the labeled training set, which is initially extracted from the initial labeled data set and enlarged in every iteration. The support vectors and the unlabeled data points are two key elements of the proposed strategy to enlarge the labeled training set. The parameter of each subsystem is also updated with the enlarged labeled data set. Through the iterations, the hyperplanes and the parameters converge to the true values, whose proof is demonstrated in the chapter. We also analyze the computational complexity of the modified self-training SVM algorithm and compare it with the traditional SVM to justify the profit of our algorithm.

Following the description of the proposed method, we do a series of simulations to provide an intuitive expression of our method. The first simulation shows the superiority of adopting the standardized Euclidean distance to define the local cluster. The effectiveness of the proposed approach is illustrated through the second simulation. Finally, the performances of the existing approaches and the proposed approach are contrasted in the third simulation.

Through the analysis of the computational complexity and the simulation, it is obvious that the proposed approach is computationally efficient for the estimation of the polyhedral partition. In addition, our method is capable of accomplishing the estimation of the polyhedral partition with partially-labeled data set. However, the proposed approach also suffers a minor drawback: Several parameters, e.g., δ, ρ, and α, are required to be tuned. The choice of these parameters may affect the performance of the algorithms. This limitation can be compensated by incorporating some known knowledge about the system, i.e. fixing the number of subsystems. Additional benefits of incorporating known knowledge are more precise estimates and faster convergence.

Future research will be devoted to searching for a more efficient cluster-based algorithm for initialization. In addition, proposing a novel identification method with fewer or even no coefficients to be tuned can also be considered as future works.

Framework for Online Identification of Continuous-Time Piecewise Affine Systems

3.

In this chapter, we propose an online identification framework for continuous-time PWA systems in state-space form, with an arbitrary number of subsystems and unknown partitions or subsystems. The number of subsystems, subsystem parameters, and the corresponding polyhedral partitions are estimated via the framework. In the framework, the identification problem of PWA systems is solved in three stages: a) online active mode recognition, b) online estimation of the subsystem parameters, c) estimation of polyhedral partitions. In the first stage, we analyze the discrete-time dynamics of the proposed delay error and design an online algorithm to recognize the active mode and estimate the number of subsystems hinged on the dynamics. According to the recognized active mode, we generalize the integral concurrent learning identifier to estimate the parameters of each subsystem and provide its convergence proof. Consider the estimation of polyhedral partitions, the process is accomplished by minimizing the proposed cost function based on the estimated subsystem parameters.

3.1. Introduction to PWA Systems and Problem Formulation

The dynamics of the PWA systems are characterized by a continuous state and a discrete mode. Specifically, the dynamics of the PWA systems vary from the switching behavior and the switching dependents on the polyhedral partition that the state vector stays. The continuous-time PWA system is more superior while describing the intrinsic mechanism of complex systems. In addition, continuous-time PWA systems provide a better depiction of the systems in the case of rapidly or irregularly sampled data.

3.1.1. Continuous-Time PWA Systems

The continuous-time PWA system in state-space form is defined as follows

$$\dot{x}(t) = \begin{cases} A_1 x(t) + B_1 u(t) + f_1, \text{ if } \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} \in \mathcal{X}_1, \\ \quad \vdots \qquad\qquad\qquad \vdots \\ A_s x(t) + B_s u(t) + f_s, \text{ if } \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} \in \mathcal{X}_s, \end{cases} \tag{3.1}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^p$ are measurable state and input vectors. The PWA system (3.1) consists of $s \in \mathbb{N}^+$ subsystems and each of them is characterized by the unknown parameters $A_i \in \mathbb{R}^{n\times n}$, $B_i \in \mathbb{R}^{n\times p}$ and $f_i \in \mathbb{R}^n$, respectively. Correspondingly, the state-input space $\mathcal{X}$ is partitioned into s polyhedral partitions. The i-th subsystem is only activated when the state-input vector stays inside the polyhedral partitions $\mathcal{X}_i$. Each polyhedral partitions $\mathcal{X}_i$ can be represented by a number of $\mu_i \in \mathbb{N}^+$ hyperplanes with a combined matrix

$$\mathcal{H}_i = \begin{bmatrix} h_{1,i} \\ \vdots \\ h_{\mu_i,i} \end{bmatrix}, \tag{3.2}$$

where $\mathcal{H}_i \in \mathbb{R}^{\mu_i\times(n+p+1)}$. Since the partition $\mathcal{X}_i$ is the intersection of μ_i half spaces, the i-th polyhedral partition is given by

$$\mathcal{X}_i = \left\{ \begin{bmatrix} x \\ u \end{bmatrix} \in \mathbb{R}^{n+p} \middle| \mathcal{H}_i \begin{bmatrix} x \\ u \\ 1 \end{bmatrix} \preceq_{[\mu_i]} 0 \right\}, \tag{3.3}$$

where $\preceq_{[\mu_i]}$ stands for a list of operators ($<$ or $\leq$). Here, the operators need to be selected appropriately to avoid overlapping partitions and ensure that only one subsystem being activated once. Therefore, the polyhedron partitions fulfill the conditions $\mathcal{X} = \bigcup_{i=1}^{s} \mathcal{X}_i$ and $\mathcal{X}_i \bigcap \mathcal{X}_j = \emptyset$, $\forall i \neq j$.

It is worth noticing that the transformation between the PWA system and the PWARX system is feasible. The interplay of continuous dynamics and state-based mode switches in PWA systems and PWARX systems complicates the realization theory of this class of systems [59, 60]. For every PWARX system, there exists an equivalent PWA state-space system. There exist, however, some PWA state-space systems that cannot be converted to the class of PWARX systems. It follows that the class of systems realizable by PWARX models is a subset of the class of systems realizable by PWA state-space models. Even if the PWA state-space systems can be transformed into the class of PWARX systems, one known problem is that the conversion of PWA state-space systems to PWARX systems can result in a tremendous increase of modes and parameters.

3.1.2. Problem Formulation and State of the Art Limitations

The most general problem formulation for identification of continuous-time PWA systems can be described as follows:

Problem 2. Given the current and previous measured state vectors x and input vectors u generated by the PWA system (3.1), derive a recursive (online) algorithm such that the estimation of the number of subsystems s, the parameters of each subsystem A_i, B_i, f_i, and the polyhedral partition of each subsystem $\mathcal{X}_i$ are guaranteed to converge to the true system parameters.

The identification of PWA systems includes estimating the number of subsystems, parameters of each affine subsystem together with the corresponding partitions. The main

challenge that arises here is the coupling between subsystem identification and partition estimation. Even provided the measurements of the state and input, it is still difficult to reveal which subsystem and partition they belong to. Therefore, multifarious methods to tackle the identification problem of PWA systems have been presented. Overviews of identification approaches for PWA systems can be referred to in [26, 61].

The majority of the identification approaches of PWA systems are based on discrete-time models in input-output form. For instance, in [37–41], various identification approaches are presented to deal with the piecewise autoregressive exogenous (PWARX) models. Nevertheless, by comparing with the input-output form, the PWA models in the state-space form are more suitable to describe the intrinsic mechanism of systems so as to design the corresponding control strategies. It is straightforward to identify the PWA models in the state-space form by converting them into the PWARX models. However, the conversion can lead to a tremendous increase of modes and parameters [26]. Therefore, some approaches for identifying the discrete-time PWA models in the state-space form directly have been reported in [62–64]. Note that the proposed identification approaches in the aforementioned papers only focus on discrete-time PWA models. However, direct identification of continuous-time models based on sampled data can outperform the discrete-time models in the case of rapidly or irregularly sampled data [25]. Therefore, the identification of continuous-time PWA systems has drawn attention and a series of papers to solve the problem have been published in the past decade [65, 66].

From another perspective, the identification methods for PWA systems can also be categorized as online and offline. Most of the previous works investigate offline methods which suffer from high computational costs with an increasing number of measurements and higher system dimensions. Alternatively, the online or recursive methods are in favor for real-time applications due to their abilities to handle large amounts of data. Thus, some researchers also propose the online or recursive methods to solve the identification problem of discrete-time PWA systems [67–69]. In this regard, we focus on the online identification of continuous-time PWA systems in this paper.

Thanks to the aforementioned advantages, the online identification of continuous-time PWA models in the state-space form has gradually received considerable attention in recent years. Very recently, Kersting and Buss proposed a series of online identification methods for PWA systems [44]. In [44], the identification problem is tackled by generalized adaptive parameter identifiers and enhanced concurrent learning identifiers independently. Note that persistence excitation (PE) condition is the preliminary of the convergence of the former identifiers. The PE condition plays a critical role in system identification and can be described as: If the exogenous reference input contains as many spectral lines as the number of unknown parameters, then the plant states are PE, and the parameter error converges exponentially to zero [70]. However, the condition on persistent excitation of the input is restrictive and often infeasible to monitor online [71]. Therefore, the enhanced concurrent learning identifiers based on the relaxed PE condition are also proposed in [44]. The shortcoming of the enhanced identifiers, nonetheless, is the requirement of the state derivatives. Generally, the precision of the estimated state derivatives may not be guaranteed. It is also worth noticing that the premise of the above identification approaches is that the switching hyperplanes are known a priori. This assumption severely limits the application of the identifiers since the switching hyperplanes are generally required to be estimated. Meanwhile, it also transforms the identification problem of PWA systems into several affine systems to

avoid the coupling challenge. In addition, the paper [43] proposed an online identification method for PWA systems with unknown subsystems and partitions. The continuous PWA is recast into max-form representation and estimated online via a designed cost function. The approach is limited to solely identify the bimodal and trimodal PWA systems.

The concurrent learning technique has been applied widely for the system control and identification [44, 72–74]. The technique can be summarized as updating the estimated parameters recursively based on designed updating law until the estimated parameters converge to the true values. It is firstly introduced in [71] to overcome the limitations in adaptive systems, which is the assumption on the persistence of excitation. The concurrent learning technique improves the convergence rates of the estimated parameters and relaxes the assumption by using the current and past measurements concurrently. Furthermore, concurrent learning is especially promising in the field of switched systems due to providing a continuous adaptation regardless of which subsystem is activated [44, 73, 74]. Later, a novel integral concurrent learning method is presented in [75] that circumvents the need for state derivatives while maintaining parameter convergence properties. Note that the method in [75] is developed for linear systems instead of switching systems.

3.2. Framework for Online Identification of Continuous-Time PWA System

The online identification task can be decomposed into three sub-tasks: the switching detection and active mode estimation, the subsystem identification, and the polyhedral partition estimation. The algorithms for accomplishing these tasks compose the framework for online identification of PWA system. The overview of the approach is illustrated through the following block diagram.

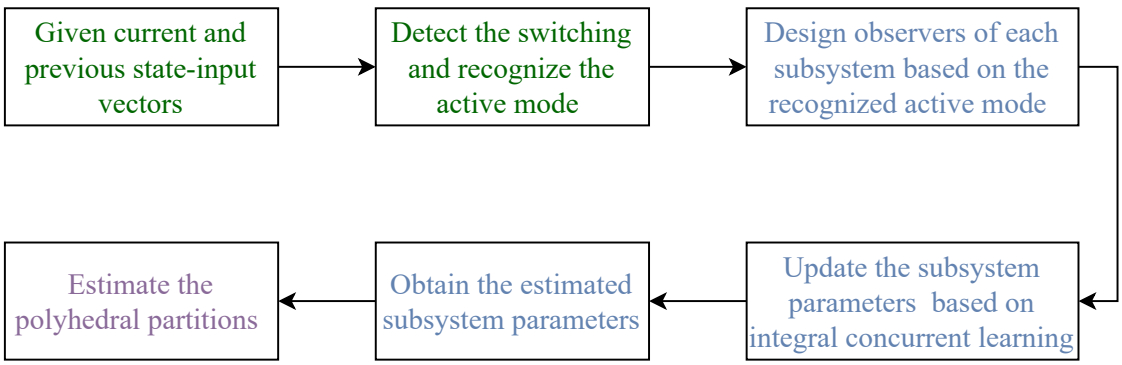


Figure 3.1.: The overview of the proposed online identification approach. The different color blocks indicate different stages in the proposed approach.

3.2.1. Switching Detection and Active Mode Recognition

In this section, we propose the algorithm for recognizing which subsystem is activated currently, i.e., the active mode, in the PWA system. To achieve the objective, it is intuitive to consider the parameters of each subsystem as the criteria for estimating the active mode.

However, it's impractical to acknowledge the parameters of each subsystem directly. Therefore, the proposed method recognizes the active mode indirectly by imposing delay errors and analyzing the dynamics of the delay errors while exciting the PWA system with a proper input signal.

Before presenting the algorithm, we introduce the following assumption and theorem.

Assumption 1. The state matrices of the PWA system (3.1) satisfy

$$e^{A_i} \neq e^{A_j}, \forall i \neq j \text{ and } i, j \in \{1, 2, \dots s\}. \tag{3.4}$$

The recognition of active mode is quite a challenge under the condition of solely measuring the previous and current inputs and states. Thus, we present the technical assumption (Assumption 1) such that different characteristics of each subsystem are provided for active mode recognition. Moreover, the assumption is feasible while considering the real systems that can be modeled as PWA, such as wheeled mobile robot (WMR) or hybrid two-tank system [43, 44].

According to the measured previous and current state vectors of the PWA system, we define the delay error $\varepsilon(t) \in \mathbb{R}^n$ as follows

$$\varepsilon(t) = x(t) - x(t - t_e), \quad t \geq t_e, \tag{3.5}$$

where $t_e \in \mathbb{R}^+$ is the user-defined delay shift. By the definition of error vector, we sample a group of $\varepsilon(t)$ at current and previous time instants to comprise the sampling matrix $\Xi(t) \in \mathbb{R}^{n \times N}$

$$\Xi(t) = [\varepsilon(t), \varepsilon(t - t_s), \dots, \varepsilon(t - (N-1)t_s)], \tag{3.6}$$

under the condition that $t \geq t_e + Nt_s$. In the equation, $N \in \mathbb{N}^+$ is the sampling size and $t_s \in \mathbb{R}^+$ is the sampling interval. Note that the choice of N, t_e, and t_s are discussed later. Following the above concepts, the following theorem is asserted to carry out the switching detection.

Theorem 2. Given a delay shift t_e, suppose each subsystem of the PWA system is excited by a periodic input signals whose period T satisfies $iT = t_e$, where $i \in \mathbb{N}^+$. Consider a time interval $[t_1, t_2]$, $t_2 > t_1 + Nt_s + t_e$. If there is no switching occurs within $[t_1, t_2]$, then there exists a fixed state-transition matrix Φ satisfying

$$\Xi(t) = \Phi \Xi(t - t_s), \tag{3.7}$$

while $t \in [t_1 + Nt_s + t_e, t_2]$.

Proof. To prove the theorem, we consider the case when one subsystem is activated. For the sake of convenience, the subsystem index i is neglected throughout the proof (i.e. $A_i \triangleq A$, $B_i \triangleq B$ or $f_i \triangleq f$). For the current time t and $t - t_e$, the system dynamics are given by

$$\dot{x}(t) = Ax(t) + Bu(t) + f, \tag{3.8}$$

$$\dot{x}(t - t_e) = Ax(t - t_e) + Bu(t - t_e) + f. \tag{3.9}$$

By the subtraction of equations (3.8) and (3.9), the dynamics of $\varepsilon(t)$ is governed by

$$\dot{\varepsilon}(t) = A\varepsilon(t) + B\left(u(t) - u(t - t_e)\right). \tag{3.10}$$

Additionally, in light of $u(t) - u(t - t_e) = 0$, the equation (3.10) can be rewritten as

$$\dot{\varepsilon}(t) = A\varepsilon(t). \tag{3.11}$$

Clearly, the equation (3.11) is state-space representation of a linear invariant system with zero input. The solution of (3.11) can be calculated explicitly as follows:

$$\varepsilon(t) = e^{At}\varepsilon_0, \tag{3.12}$$

where $\varepsilon_0 \in \mathbb{R}^n$ is the first error state vector after the switching occurs, i.e., initial error state vector. After acquiring the solution of $\varepsilon(t)$, the state error $\varepsilon(t)$ can be described as follows by imposing a time interval t_s

$$\varepsilon(t) = e^{At}\varepsilon_0 = e^{At_s}e^{A(t-t_s)}\varepsilon_0 = e^{At_s}\varepsilon(t - t_s). \tag{3.13}$$

Thus, we obtain a series of equations

$$\begin{aligned} \varepsilon(t) &= e^{At_s}\varepsilon(t - t_s), \\ \varepsilon(t - t_s) &= e^{At_s}\varepsilon(t - 2t_s), \\ &\vdots \\ \varepsilon(t - (N-1)t_s) &= e^{At_s}\varepsilon(t - Nt_s). \end{aligned} \tag{3.14}$$

By the definition of $\Xi(t)$, the series of equations can be written in matrix form

$$\Xi(t) = e^{At_s}\Xi(t - t_s) = \Phi\Xi(t - t_s), \tag{3.15}$$

where Φ is the state transition matrix. This leads to the result of Theorem 2. □

The main challenge of implementing Theorem 2 for switching detection is how to design the input signals. Fortunately, the papers [44, 66] have already provided the exemplary input signals for online identification. In these papers, the input signals are designed as the sum of different frequencies sinusoidal signals and piecewise constant/ramp signal. The piecewise constant or ramp signal is designed for switching between different subsystems. In addition, the sum of different frequencies sinusoidal signals is adopted to excite each subsystem and fulfill the PE-like condition as well. The PE-like condition guarantees the converge of estimated parameters. These signals perfectly conform that each subsystem is excited by a periodic input signal. Therefore, it is easy to design a similar input signal which meets the requirements of Theorem 2 by adjusting the delay shift t_e and the frequencies of sinusoidal signals. Specifically, a typical instance of the input signal for online identification is presented in Section 3.3.

It is worth noticing that Theorem 2 only aims at switching detection. It is still essential to recognize the mode of the active subsystem in online identification. Theorem 2 describes the existence of a fixed state-transition matrix (STM) Φ in the non-switching conditions. The matrix Φ is also required to be unique such that the estimated Φ can be adopted for the active mode recognition. Obviously, the uniqueness of Φ is equivalent to the existence of a full row rank sampling matrix. It follows that all the sampling matrices in a non-switching time interval are full row rank as long as there exists one full row rank sampling matrix in the interval according to the equations (3.5) and (3.7). Generally, there always exist a full

row rank $\Xi(t)$ due to the sampling size $N > n$ and the measurement noises. Nonetheless, there are still some extreme situations that lead to the row rank-deficient $\Xi(t)$ (noiseless) or ill-posed $\Xi(t)$ (with noises). In the section 3.2.1, we discuss the extreme situation and propose a refinement strategy to acquire the unique Φ under the situation. Following the uniqueness prerequisite, the constant value of estimated Φ only depends on the state matrix A_i of each subsystem while t_s is fixed. In the light of Assumption 1, the constant value of Φ with respect to different active modes are distinct. Therefore, the steady STM Φ can be estimated as a criterion to recognize the active mode.

The online recognition of the active mode is accomplished by implementing Algorithm 3. In the algorithm, the STM $\Phi(t)$ is estimated via the iterative Tikhonov method [76,77]. Note that the estimated STM is denoted by $\hat{\Phi}(t)$. In fact, the traditional least square (LS) method seems also applicable to calculate $\hat{\Phi}(t)$ by solely referring to the equation (3.7). However, the estimation problem is ill-posed with the passage of time while the STM is convergent. In other words, noises will dominate the sampling matrix $\Xi(t)$ and a small perturbation of the data may result in large perturbations of the solution [78]. The ill-posed problem can not be solved by the traditional LS method [76]. Therefore, we employ the iterative Tikhonov method, which is proposed to tackle the ill-posed problem, in our algorithm. The iterative Tikhonov method in standard form is summarized as solving the following penalized minimization problem

$$\min_{\hat{\Phi}\in\mathbb{R}^{n\times n}} ||\Xi(t) - \hat{\Phi}(t)\Xi(t-t_s)|| + \alpha||\hat{\Phi}(t) - \hat{\Phi}(t-t_s)||, \tag{3.16}$$

where α is a predefined regularized parameter. The value of α determines how sensitive $\hat{\Phi}(t)$ is to the $\Xi(t)$ and how close the solution is to the previous estimated STM $\hat{\Phi}(t - t_s)$. The choice of α is important in the iterated Tikhonov method and many strategies have been proposed in the literature [79]. To simplify, we adopt a stationary α during the iteration. Consider our estimation problem, the regularized parameters α should be assigned a relatively large value such that the estimated $\hat{\Phi}(t)$ is not too sensitive to the sampling matrix $\Xi(t)$ while it is dominated by noises. On the other hand, a too large value of α inhibits the abrupt change of $\hat{\Phi}$ while switching occurs. This situation is also unacceptable for active mode recognition. In fact, the choice of α is quite loose in our algorithm and the effects on performance are presented in the simulation section.

The iterated Tikhonov method [79] can then be expressed compactly in the following form as the solution of the penalized minimization problem 3.16

$$\begin{aligned}\hat{\Phi}(t) = \hat{\Phi}(t-t_s) + \left(\Xi(t) - \hat{\Phi}(t-t_s)\Xi(t-t_s)\right)\Xi^{\mathsf{T}}(t-t_s) \\ \left(\Xi(t-t_s)\Xi^{\mathsf{T}}(t-t_s) + \alpha I\right)^{-1}.\end{aligned} \tag{3.17}$$

Obviously, the iterative Tikhonov method considers the previous estimated STM in the estimation. This strategy ensures that the estimated STM is not determined by the noises in the ill-posed conditions.

Subsequently, the online active mode recognition can be treated as the online steady-state detection of the estimated STM $\hat{\Phi}$ by Theorem 2. Some methods for online steady-state detection have been documented in [80–82]. In the algorithm, we adopt the method from [80] and analyze the standard deviation of $\hat{\Phi}$ over a predefined moving window η and a predefined threshold γ.

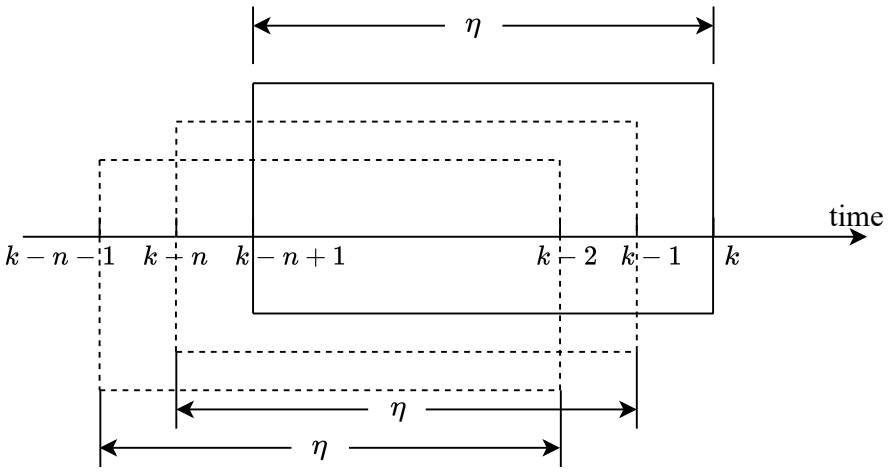


Figure 3.2.: Moving windows of η data points at near k-th time

The online steady-state detection algorithm in [80] is introduced for a vapor compression system based on a moving window and the standard deviations of seven measurements. The concept of the steady-state detector originates from noise filter theory. The most common and simple steady-state detectors analyze the data over a predefined moving window, as illustrated in Figure 3.2.

The variance or standard deviation of important parameters is typically utilized to indicate the statistical spread within the data distribution and can be used to characterize random variation of the measured signals. The steady-state detector in [80] calculates the standard deviation of the parameters in a recursive fashion. Suppose that at any instant k, the average of the latest η samples of a data sequence, ϕ_i, is given by

$$\bar{\phi}_k = \frac{1}{\eta} \sum_{i=k-\eta+1}^{k} \phi_i. \tag{3.18}$$

The difference between two averages of the latest η samples at the current time, k, and at the previous time instant, $k-1$, is

$$\bar{\phi}_k - \bar{\phi}_{k-1} = \frac{1}{\eta} \left(\sum_{i=k-\eta+1}^{k} \phi_i - \sum_{i=k-\eta}^{k-1} \phi_i \right) = \frac{1}{\eta} \left(\phi_k - \phi_{k-\eta} \right). \tag{3.19}$$

Naturally, the following equation yields

$$\bar{\phi}_k = \bar{\phi}_{k-1} + \frac{1}{\eta} \left(\phi_k - \phi_{k-\eta} \right). \tag{3.20}$$

This approach is known as a moving window average because the average at each k-th instant is based on the most recent set of η values. Similarity, a moving window variance v_k can be define as

$$v_k = \frac{1}{\eta} \sum_{i=k-\eta+1}^{k} \left(\phi_i - \bar{\phi}_k \right)^2 = \frac{1}{\eta} \sum_{i=k-\eta+1}^{k} \phi_i^2 - \bar{\phi}_k^2, \tag{3.21}$$

$$v_k = v_{k-1} + \frac{1}{\eta} \left(\phi_k^2 - \phi_{k-\eta}^2 \right) - \left(\bar{\phi}_k^2 - \bar{\phi}_{k-1}^2 \right). \tag{3.22}$$

The moving window standard deviation is then given as

$$\chi_k = \sqrt{v_k}. \tag{3.23}$$

The steady-state detector identifies steady operation if the standard deviations for the selected features representing the status of the system fall below the defined threshold γ.

As reported in [80], the threshold γ is generally assigned as 3χ, where χ is the standard deviation of the steady-state. If $\hat{\Phi}$ remains a steady-state, the switching does not occur and the steady-state is recorded as the criterion for the active subsystem. Additionally, we impose a user-defined threshold ρ to discriminate whether the currently active mode is a new one or repetitive one and determine the active mode $\hat{\sigma}(t)$ accordingly. Otherwise, the switching occurs in the moving window, the estimated active mode $\hat{\sigma}(t)$ is set as zero until $\hat{\Phi}$ keeps steady again. Note that the state and input vectors $x(t)$, $u(t)$ corresponding to $\hat{\sigma}(t) = 0$ are skipped in the next stage of identification. Eventually, the online switching detection and active mode recognition are realized via Algorithm 3.

Algorithm 3 Recognition of Active Mode

Input: delay shift t_e, sampling interval t_s, sampling size N, moving windows size η, thresholds ρ, γ

Output: The estimated number of subsystems $\hat{s}$, and the recognized active mode $\hat{\sigma}(t)$.

Let $\hat{s} = 1$, $t_0 = t_e + Nt_s$.
Calculate the initial STM $\hat{\Phi}(t_0)$ with LS
while $t \geq t_0$ **do**
  Use the iterative Tikhonov method to update $\hat{\Phi}(t)$
  Discriminate the steady-state of $\hat{\Phi}(t)$ with γ and η.
  if $\hat{\Phi}(t)$ is a steady state **then**
    if $\Xi(t)$ is row rank-deficient **and** Flag==True **then**
      Update the $\hat{\Phi}(t)$ for iterative Tikhonov method with the refinement strategy in Section 3.2.1
      Flag← False
    end if
    if $\min_{i=1,\ldots,\hat{s}}(||\hat{\Phi}(t) - \hat{\Phi}_i||) > \rho$ **then**
      $\hat{s} \leftarrow \hat{s} + 1$, $\hat{\sigma}(t) \leftarrow \hat{s}$, $\hat{\Phi}_i \leftarrow \hat{\Phi}(t)$
    else
      $\hat{\sigma}(t) \leftarrow \text{argmin}_{i=1,\ldots,\hat{s}}(||\hat{\Phi}(t) - \hat{\Phi}_i||)$
    end if
  else
    $\hat{\sigma}(t) \leftarrow 0$, Flag← True
  end if
end while
return $\hat{s}$, and $\hat{\sigma}(t)$.

According to Algorithm 3, there exists a delay between active mode recognition and the switching occurrence, i.e., $t_e + Nt_s + \eta$. For instance, if the i-th subsystem is activated at time instant $\hat{t}$, the active mode is estimated as i at the time instant $\hat{t} + t_e + Nt_s + \eta$ via Algorithm 3. Therefore, the dwell time, being the time between consecutive switchings, should be long

enough for the recognition of the active mode. Then, the identification of each subsystem is carried out based on the recognized active mode. Obviously, the requirement can be easily fulfilled by adjusting the piecewise constant or ramp component in the input signal as well as t_e, N, t_s, and η.

The assignment of t_e, N, t_s, and η are still a major problem to implement the algorithm. The sampling matrix $\Xi(t)$ and the delay in the mode recognition are affected by the choice of t_e, N, and t_s, Generally, the sampling size is chosen such that $N \gg n$ to ensure the full row rank of $\Xi(t)$. Meanwhile, the delay shift t_e and sampling interval t_s are assigned small values for a shorter delay in the mode recognition. However, too small t_e and t_s cause that the entries of all sampling matrices are close to zero. Hence, the estimated $\hat{\Phi}$ is not applicable for the mode recognition. Clearly, this circumstance only occurs while choosing extremely small t_e and t_s.

Remark 4. From Theorem 2 and Algorithm 3, the switching detection and mode recognition are achieved via the estimated STM. The formula of the STM is

$$\Phi_i = e^{A_i t_s}. \tag{3.24}$$

In the above equation, t_s is the pre-defined sampling interval and A_i is the state matrix of the i-th subsystem. It is quite natural to wonder if it is possible to calculate the state matrix A_i of the i-th subsystem via the estimated state-transition matrix Φ_i. However, the state matrix A_i cannot be calculated with the identified STM Φ_i. The reason is that some matrices may have more than one logarithm. The logarithm of a matrix is another matrix such that the matrix exponential of the latter matrix equals the original matrix. It is the inverse operation of the matrix exponential. For instance, if the estimated STM $\Phi_i = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$, then the state matrix A_i could be either $\frac{1}{t_s} \cdot \begin{bmatrix} 0 & \pi \\ -\pi & 0 \end{bmatrix}$ or $\frac{1}{t_s} \cdot \begin{bmatrix} 0 & 3\pi \\ -3\pi & 0 \end{bmatrix}$. Thus, the estimated STM can not be used to directly calculate the state matrix A_i. Actually, this feature is also the inspiration of the Assumption 1. The state-transition matrix is employed as the criteria for mode recognition in the proposed method and should be different in each subsystem. However, one state-transition matrix Φ_i may correspond to more than one state matrix A_i as stated above. Therefore, the technical assumption focuses on the distinct state-transition matrices of each subsystem for mode recognition.

Discussion

Full row rank of the sampling matrix $\Xi(t)$ plays a fundamental role in the proposed identification approach. However, in very extreme situations, row rank-deficiency may occur. Thus, in this subsection, we look into the row rank of the sampling matrix $\Xi(t)$ and provide a refinement strategy to deal with the row rank-deficient $\Xi(t)$.

It is worth noticing that the STM $\hat{\Phi}(t)$ can also be estimated even if $\Xi(t)$ is row rank-deficient due to the adoption of the iterative Tikhonov method. Nonetheless, the estimated $\hat{\Phi}(t)$ is not only determined by the sampling data in $\Xi(t)$ but also the last estimation $\hat{\Phi}(t-t_s)$. This feature leads to the consequence that different fixed estimated STMs may be obtained

for the same subsystem by applying the proposed method. It follows that the initial STM which is calculated by the first sampling matrices Ξ_0 might be different each time the same subsystem is activated. Therefore, the outline of the refinement strategy is to reformulate a full row rank matrix by concatenating the sampling matrices excited by the adjusted inputs. Note that the adjusted input guarantees that no switching occurs. Ultimately, we can obtain the unique initial STM for the iterative Tikhonov method by using the reformulated matrix.

According to the equation (3.7), all sampling matrices in the interval are row rank-deficient if there exists a row rank-deficient sampling matrix in a non-switching interval. Thus, we only need to analyze the row rank of one sampling matrix in this section.

The first sampling matrices Ξ_0 after the switching instant t_{sw} is represented as

$$\Xi_0 = \begin{bmatrix} e^{A(N-1)t_s}\varepsilon_0 & e^{A(N-2)t_s}\varepsilon_0 & \dots & e^{At_s}\varepsilon_0 & \varepsilon_0 \end{bmatrix}, \tag{3.25}$$

where

$$\varepsilon_0 = x(t_{sw} + t_e) - x(t_{sw}). \tag{3.26}$$

According to these equations and Popov-Belovich-Hautus (PBH) test in control theory, the sampling matrix Ξ_0 is row rank-deficient if and only if the rank of the augmented matrix $\text{rank}[e^{At_s} - \lambda I \quad \varepsilon_0] < n$ for every complex eigenvalue λ of A [83]. This condition indicates that the row rank of the sampling matrix is resolved by ε_0. The specific relationship between ε_0 and e^{At_s} is not the focus of this dissertation. Instead, we propose the refinement strategy and handle the problem based on the role of ε_0 in the determination of row rank. In the refinement strategy, we reformulate a matrix by uniting the sampling matrices with distinct ε_0, which is achieved via the adjusted input signal. For instance, two sampling matrices Ξ_0 and Ξ_0' with distinct initial vector ε_0 and ε_0' for the same subsystem are acquired. Then the reformulated matrix is given by

$$[\Xi_0 \ \Xi_0'] = \begin{bmatrix} e^{A(N-1)t_s}[\varepsilon_0 \ \varepsilon_0'] & e^{A(N-2)t_s}[\varepsilon_0 \ \varepsilon_0'] & \dots & e^{At_s}[\varepsilon_0 \ \varepsilon_0'] & [\varepsilon_0 \ \varepsilon_0']. \end{bmatrix} \tag{3.27}$$

Now, the row rank of $[\Xi_0 \ \Xi_0']$ is determined by $\text{rank}[e^{At_s} - \lambda I \quad \varepsilon_0 \ \varepsilon_0']$ for every complex eigenvalue λ of A according to the PBH test [83]. The reformulated matrix can be easily full row rank by concatenating the sampling matrices with distinct initial vectors, i.e., ε_0. These sampling matrices with distinct ε_0 are obtained by adjusting the input signal in the refinement strategy.

The refinement strategy is described as follows. Firstly, after the switching occurs, if the first sampling matrix Ξ_0 is row rank-deficient, it is recorded as part of the reformulated matrix. Then, the phase of the input signal is adjusted to acquire other sampling matrices. During the excitation of the adjusted input signals, a sampling matrix is recorded again and appended into the reformulated matrix. Note that the adjustment of the input signal could be the aptitude, phase, or frequency as long as the period t_e remains and the current subsystem is still activated. The above steps are repeated until the reformulated matrix is of full row rank. Then, the initial STM is determined via the reformulated matrix and the iterated Tikhonov method carries on estimating the STM based on the initial STM. Consider the characteristics of the iterative Tikhonov method, the estimated STM remains steady until the switching occurs. Meanwhile, the unique matrix can be treated as the criterion for mode recognition.

The main limitation of the refinement strategy is the time cost for establishing the reformulated matrix. The active mode is unacknowledged and set as zero before the reformulated matrix is obtained, which requires more time for identification. Therefore, the delay shift t_e and t_s shall be assigned a small value under this circumstance. In Section 3.3, a simulation is provided to validate the effectiveness of the proposed refinement strategy while the sampling matrix is singular.

3.2.2. Parameter Identifiers based on Integral Concurrent Learning

In this section, we concentrate on the online identification of each subsystem and propose a generalized integral concurrent learning adaptive identifier. The identifier was firstly proposed to identify linear systems and documented in [75]. We generalize the identifier to realize the identification of the PWA system. Before introducing the identifier, we define the integrated history stacks as follows.

Definition 3. Let t_{i_k} denote the k-th selected time instant while the i-th subsystem is active. Suppose there holds $t_{i_k} \in [\Delta t, t]$, where Δt is the integration window and t is the current time. The q-element ($q \in \mathbb{N}^+$) integrated history stacks of states and inputs of subsystem i are defined as

$$\begin{aligned} \mathbf{X}_i &= [X(t_{i_1}), X(t_{i_2}), \dots, X(t_{i_q})], \\ \mathbf{U}_i &= [U(t_{i_1}), U(t_{i_2}), \dots, U(t_{i_q})], \end{aligned} \tag{3.28}$$

where

$$X(t) = \int_{t-\Delta t}^{t} x(\tau) d\tau, \quad U(t) = \int_{t-\Delta t}^{t} u(\tau) d\tau, \tag{3.29}$$

and

$$\begin{bmatrix} x(\tau)^\mathsf{T} & u(\tau)^\mathsf{T} \end{bmatrix}^\mathsf{T} \in \mathcal{X}_i \text{ for } \tau \in [t_{i_k} - \Delta t, t_{i_k}]. \tag{3.30}$$

For the generalized integral concurrent learning identifier, the following assumption on the integrated history stacks arises to ensure the convergence of the estimated parameters.

Assumption 2. For each subsystem i, the integrated history stacks $\mathbf{X}_i$ and $\mathbf{U}_i$ contain q elements, such that there exists $n+p+1$ linearly independent vectors $[X(t_{i_k})^\mathsf{T}, U(t_{i_k})^\mathsf{T}, 1]^\mathsf{T}$.

This assumption 2 indicates that the sufficient excitation only lasts for a finite period of time, which is weaker than the typical PE condition. However, this assumption is adequate for the integral concurrent learning techniques to accomplish the identification. The proof of parameters convergence based on the assumption will be provided later. Note that the number of recorded elements is generally $q = n + p + 1$ in order to confine computational complexity. The methods for recording new integrated history stacks and replacing existing ones are presented in Section 3.2.3 in detail.

We denote the estimated parameters by $\hat{A}_i \in \mathbb{R}^{n\times n}$, $\hat{B}_i \in \mathbb{R}^{n\times p}$, and $\hat{f}_i \in \mathbb{R}^n$. The objective of the identifier is to design an update law so that these estimates converge to the true parameters. To achieve the target, we firstly design the state observer to predict the state of the i-th subsystem

$$\dot{\hat{x}}_i = A_m \hat{x}_i + (\hat{A}_i - A_m)x + \hat{B}_i u + \hat{f}_i, \quad \text{if } \hat{\sigma}(t) = i, \tag{3.31}$$

where $A_m \in \mathbb{R}^{n\times n}$ is a stable (Hurwitz) matrix. For a Hurwitz matrix, there always exists a symmetric, positive definite $P \in \mathbb{R}^{n\times n}$ satisfying

$$A_m^\mathsf{T} P + P A_m = -Q_e, \tag{3.32}$$

where $Q_e \in \mathbb{R}^{n\times n}$ is a positive definite matrix.

The updating law of the proposed identifier is based on the prediction error, which is defined as $e_i = \hat{x}_i - x$. However, the state observer (3.31) is only designed to predict the state while the subsystem is active. Thus, we impose the discrete state reset from [66] to overcome this limitation. The predicted state $\hat{x}_i$ is reset and set equal to the true state x while the corresponding subsystem is inactive, i.e.,

$$\hat{x}_i = x, \quad \text{if } \hat{\sigma}(t) \neq i. \tag{3.33}$$

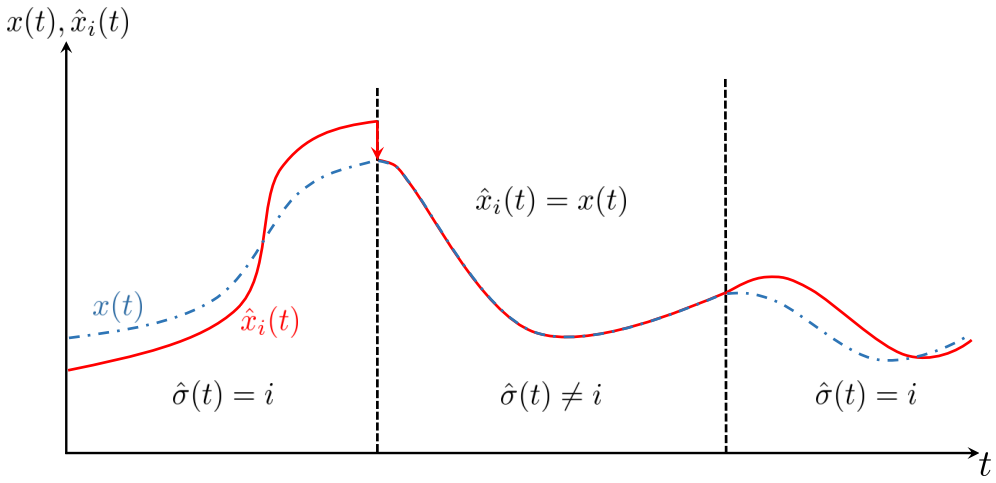


Figure 3.3.: State reset of the state prediction for the i-th subsystem upon deactivation of the i-th subsystem dynamics, i.e., $\hat{x}_i(t) = x(t)$ as $\hat{\sigma}(t) \neq i$.

Figure 3.3 shows that the state is predicted while the i-th subsystem is active and reset to the true state while it is inactive. The state reset guarantees that the states from other subsystems are abandoned during the i-th subsystem identification.

Before proposing the update law, we transform the estimated and true parameters into vectors for the sake of convenience. We denote

$$\theta_i = \begin{bmatrix} \operatorname{vec}(A_i)^\mathsf{T} & \operatorname{vec}(B_i)^\mathsf{T} & \operatorname{vec}(f_i)^\mathsf{T} \end{bmatrix}^\mathsf{T} \in \mathbb{R}^{n(n+p+1)}, \tag{3.34}$$

$$\hat{\theta}_i = \begin{bmatrix} \operatorname{vec}(\hat{A}_i)^\mathsf{T} & \operatorname{vec}(\hat{B}_i)^\mathsf{T} & \operatorname{vec}(\hat{f}_i)^\mathsf{T} \end{bmatrix}^\mathsf{T} \in \mathbb{R}^{n(n+p+1)}. \tag{3.35}$$

In addition, we make use of the Kronecker product $\otimes$ and introduce the matrices

$$\Gamma = \begin{bmatrix} \gamma_1 I_{nn} & 0 & 0 \\ 0 & \gamma_2 I_{np} & 0 \\ 0 & 0 & \gamma_3 I_n \end{bmatrix}, \quad \Psi = \begin{bmatrix} x \\ u \\ 1 \end{bmatrix} \otimes I_n, \tag{3.36}$$

where $\gamma_1, \gamma_2, \gamma_3 \in \mathbb{R}^+$ and $\Psi \in \mathbb{R}^{n(n+p+1)\times n}$. Furthermore, I_{nn}, I_{np}, and I_n are identity matrices of dimension nn, np, and n, respectively. Afterwards, the identification for PWA systems can be achieved by designing the following update law.

Theorem 3. Consider the PWA system (3.1) with the estimated active mode $\hat{\sigma}(t)$. Let the state of each subsystem be estimated by the corresponding observer. Then, the parameter update law of the i-th subsystem is designed as

$$\dot{\hat{\theta}}_i = -\Gamma\Psi P e_i + \frac{k_{\mathrm{CL}}}{2}\Gamma \sum_{k=1}^{q} \mathcal{Y}(t_{i_k})^{\mathsf{T}}\Big(x(t_{i_k}) - x(t_{i_k} - \Delta t) - \mathcal{Y}(t_{i_k})\hat{\theta}_i\Big), \tag{3.37}$$

where for any $k = 1, \ldots, q$,

$$\mathcal{Y}(t_{i_k}) = \int_{t_{i_k}-\Delta t}^{t_{i_k}} \Psi^{\mathsf{T}}(\tau) d\tau. \tag{3.38}$$

In the equation (3.37), $k_{\mathrm{CL}} \in \mathbb{R}^+$ is a control gain, $\Delta t \in \mathbb{R}^+$ is the size of the integration window. The update law (3.37) ensures that e_i converge to zero and the estimated parameters $\hat{\theta}_i$ converge to θ_i exponentially.

The proof of Theorem 3 is presented as follows.

Proof. In the section, Theorem 3 is proved through the designed Lyapunov function for each subsystem. The derivative of all Lyapunov functions is justified to be negative regardless of the active mode. Specifically, the estimated parameter $\hat{\theta}_i$ of subsystem i converges to the true parameter θ_i no matter the i-th subsystem is active ($\hat{\sigma} = i$) or not ($\hat{\sigma} \neq i$). Therefore, we analyze the stability of all subsystems separately at first and naturally neglect the index i throughout the proof(i.e. $A \triangleq A_i$ or $\mathcal{Y}(t) \triangleq \mathcal{Y}_i(t)$).

Based on the formulations of θ and Ψ in the equation (3.34) and (3.36), one subsystem of the PWA system is written as

$$\dot{x} = \Psi^{\mathsf{T}}\theta. \tag{3.39}$$

Naturally, the integration of the equation (3.39) yields

$$\int_{t-\Delta t}^{t} \dot{x}(\tau) d\tau = \int_{t-\Delta t}^{t} \Psi^{\mathsf{T}}(\tau)\theta d\tau, \ \forall t > \Delta t. \tag{3.40}$$

Then, substituting (3.38) into (3.40) and using the fundamental theorem of calculus, we get

$$x(t) - x(t - \Delta t) = \mathcal{Y}(t)\theta, \forall t > \Delta t. \tag{3.41}$$

Combining the equations (3.37) and (3.41), The update law of the subsystem is given by

$$\dot{\tilde{\theta}} = \dot{\hat{\theta}} = -\Gamma\Psi P e - \frac{k_{\mathrm{CL}}}{2}\Gamma \sum_{k=1}^{q} \mathcal{Y}(t_k)^{\mathsf{T}}\mathcal{Y}(t_k)\tilde{\theta}, \forall t_k > \Delta t, \tag{3.42}$$

where

$$\tilde{\theta} = \hat{\theta} - \theta. \tag{3.43}$$

In addition, the derivative of the prediction error for the active subsystem along the equations (3.36) and (3.43) takes the form

$$\dot{e} = \dot{\hat{x}} - \dot{x} = A_m e + \Psi^{\mathsf{T}}\tilde{\theta}. \tag{3.44}$$

In the proof, the quadratic candidate Lyapunov function is designed as

$$V = \begin{bmatrix} e^\mathsf{T} & \tilde{\theta}^\mathsf{T} \end{bmatrix} \begin{bmatrix} P & 0 \\ 0 & \Gamma^{-1} \end{bmatrix} \begin{bmatrix} e \\ \tilde{\theta} \end{bmatrix}. \tag{3.45}$$

Following the equations (3.42) and (3.44), the time derivative of V is given by

$$\begin{aligned} \dot{V} &= e^\mathsf{T} A_m^\mathsf{T} P e + e^\mathsf{T} P A_m e + \tilde{\theta}^\mathsf{T} \Psi P e + e^\mathsf{T} P \Psi^\mathsf{T} \tilde{\theta} \\ &\quad - e^\mathsf{T} P \Psi^\mathsf{T} \tilde{\theta} - \tilde{\theta}^\mathsf{T} \Psi P e - k_{\mathrm{CL}} \tilde{\theta}^\mathsf{T} \sum_{k=1}^{q} \mathcal{Y}(t_k)^\mathsf{T} \mathcal{Y}(t_k) \tilde{\theta} \\ &= e^\mathsf{T} (A_m^\mathsf{T} P + P A_m) e - \tilde{\theta}^\mathsf{T} k_{\mathrm{CL}} \sum_{k=1}^{q} \mathcal{Y}(t_k)^\mathsf{T} \mathcal{Y}(t_k) \tilde{\theta} \\ &= -e^\mathsf{T} Q_e e - \tilde{\theta}^\mathsf{T} Q_\theta \tilde{\theta}, \end{aligned} \tag{3.46}$$

where $Q_e = -(A_m^\mathsf{T} P + P A_m)$ and $Q_\theta = k_{\mathrm{CL}} \sum_{k=1}^{q} \mathcal{Y}(t_k)^\mathsf{T}\ \mathcal{Y}(t_k)$. The matrix Q_e is positive definite since the matrix A_m is designed to be Hurwitz. Furthermore, Q_θ has the same eigenvalues as $\sum_{k=1}^{q} [X(t_{i_k})^\mathsf{T}, U(t_{i_k})^\mathsf{T}, 1]^\mathsf{T} [X(t_{i_k})^\mathsf{T}, U(t_{i_k})^\mathsf{T}, 1]$ due to the definition of Ψ. Thus, Q_θ is also positive definite by Assumption 2. In summary, the derivative of V

$$\dot{V} = \begin{bmatrix} e^\mathsf{T} & \tilde{\theta}^\mathsf{T} \end{bmatrix} \begin{bmatrix} -Q_e & 0 \\ 0 & -Q_\theta \end{bmatrix} \begin{bmatrix} e \\ \tilde{\theta} \end{bmatrix}, \tag{3.47}$$

is negative definite and the prediction error e and $\tilde{\theta}$ converge to zero exponentially while the subsystem is active ($\sigma = i$).

Subsequently, $\dot{V}$ is also shown to be negative for $\sigma \neq i$. The prediction error $e = 0$ while $\sigma \neq i$ due to the state reset. Therefore, the derivative for Lyapunov function $\dot{V}$ transforms into

$$\dot{V} = -\tilde{\theta}^\mathsf{T} Q_\theta \tilde{\theta}, \tag{3.48}$$

for $\sigma \neq i$. This indicates that even if the subsystem is presently inactive, $\tilde{\theta}$ of the corresponding subsystem also converges to zero. Thus, the value of V is strictly decreasing according to the equations (3.47) and (3.48). In addition, the state reset (3.33) decreases V by $e^\mathsf{T} P e$. Therefore, the switching between $\sigma = i$ and $\sigma \neq i$ does not affect the convergence of e and $\tilde{\theta}$. In conclusion, the negative definiteness of $\dot{V}$ justifies that e_i and $\tilde{\theta}_i$ for each subsystem converge to zero exponentially while the subsystem is active or inactive with the update law.

The designed Lyapunov function and its derivatives only show stability in the sense of Lyapunov for all subsystems individually. As reported in [84, 85], the candidate Lyapunov functions have to be Lyapunov-like along all possible trajectories and switching sequences to achieve the stability of the overall switched system. This requires the values of the Lyapunov functions at each activation of the corresponding subsystem to form a decreasing sequence.

In order to show that the candidate Lyapunov functions are Lyapunov-like, we provide some precise notations for the switching process. Denote the instance of the q-th activation of sub-model i by $t_{i,q}^{\mathrm{in}}$, i.e., $\sigma^-(t_{i,q}^{\mathrm{in}}) \neq i$ and $\sigma^+(t_{i,q}^{\mathrm{in}}) = i$. Equivalently, the instance $t_{i,q}^{\mathrm{out}}$ refers to the q-th exit from sub-model i ($\sigma^-(t_{i,q}^{\mathrm{out}}) = i$ and $\sigma^+(t_{i,q}^{\mathrm{out}}) \neq i$). Then, the following

requirement arises for stability in the sense of Lyapunov for switched systems [84] based on these notations.

$$V_i(t_{i,q}^{\text{in}}) \geq V_i(t_{i,q+1}^{\text{in}}), \forall q > 1, \tag{3.49}$$

where V_i is the candidate Lyapunov functions value of the i-th subsystem.

For the reason that the equation (3.46) is negative definite, the value of candidate Lyapunov function fulfills

$$V_i(t_{i,q}^{\text{in}}) > V_i(t_{i,q}^{\text{out}}). \tag{3.50}$$

Furthermore, the state reset mechanism indicates that $e_i(t) = 0$ for the inactive periods $t \in (t_{i,q}^{\text{out}}, t_{i,q+1}^{\text{in}}]$. Thus, the history stack term works and the parameter errors remain decreasing during the inactive periods, i.e., $\tilde{\theta}_i(t_{i,q}^{\text{out}}) > \tilde{\theta}_i(t_{i,q+1}^{\text{in}})$. It follows that

$$\begin{aligned} V_i\left(t_{i,q}^{\text{out}}\right) &= \left[e_i^\top P e_i + \tilde{\theta}_i^\top \Gamma^{-1} \tilde{\theta}_i\right]_{t_{i,q}^{\text{out}}} > \left[\tilde{\theta}_i^\top \Gamma^{-1} \tilde{\theta}_i\right]_{t_{i,q+1}^{\text{in}}} \\ &= V_i\left(t_{i,q+1}^{\text{in}}\right), \end{aligned} \tag{3.51}$$

which shows that the equation (3.49) is satisfied. Naturally, all state errors and parameter errors converge to zero by implementing the proposed update law. □

The update law (3.37) consists of two terms. The first term $-\Gamma \Psi P e_i$ stands on current states, inputs, and prediction errors. The second term is based on selected, integrated history stacks. Therefore, the current and recorded data are used simultaneously in the update law. This property is beneficial to the identification of the PWA system since the estimated parameters of inactive subsystems are also updated based on recorded data. In addition, selecting feasible i-th integrated history stacks for the update law (3.37) is straightforward while knowing the estimated active mode $\hat{\sigma}(t)$.

It is clear that update law and corresponding proof are presented under the condition that $t_{i_k} > \Delta t, \forall i_k$ due to the lower bound of the integration in (3.38). In fact, the update law is effective while Assumption 2 is satisfied. Thus, the integrated history stacks can also be formulated with time instants $t_{i_k} < \Delta t$ as long as Assumption 2 is fulfilled. Meanwhile, the lower bound of the integration $t_{i_k} - \Delta t$ is replaced by zero. For the sake of convenience, we select the time points fulfilling $t_{i_k} > \Delta t$ in our algorithm. Even so, the time instants can still be selected in a wide range to satisfy Assumption 2 as Δt is generally assigned to a small value.

The integral concurrent learning identifier from [75] is generalized to the piecewise affine system by imposing the observer and state reset in our method. Meanwhile, the adoption of the integrated history stack in the identifier bears a major advantage compared to the traditional concurrent learning technique. The integrated history stacks ensure the convergence of estimated parameters of each subsystem without the assumption that the measured derivative matches the real derivative.

3.2.3. History Stack Management in Generalized Integral Concurrent Learning Identifier

In the discussion of the generalized integral concurrent learning identifier, the integrated history stacks are assumed to be invariant so far. However, the parameter convergence is

maintained as long as Q_θ remains positive definite considering the equation (3.47). Alternatively stated, we can arbitrarily replace the elements of the integrated history stack under the condition that Q_θ is positive definite. Note that the elements for replacement are required to be from the same subsystem as the integrated history stack.

All the integrated history stacks are empty at the beginning. Afterwards, the elements are added into the corresponding integrated history stacks until Assumption 2 is fulfilled. The new element should be sufficiently different from the previously recorded elements to fulfil the assumption. For each subsystem, we define the integrated state-input vector at the time instance t as $\Omega(t) = [X^\mathsf{T}(t)\ U^\mathsf{T}(t)]^\mathsf{T}$. Let Ω_{prev} denotes one of the recorded integrated state-input vector. The new element is sufficiently different from the previous one if the following inequality is satisfied

$$\frac{||\Omega(t) - \Omega_{\text{prev}}||^2}{||\Omega(t)||} \geq \kappa, \tag{3.52}$$

where $\kappa \in \mathbb{R}^+$ is a design threshold. Note that the inequality is slightly modified from [86] by replacing the state-input vectors into integrated state-input vectors. To confine the computational complexity, the number of recorded data elements q is generally limited as $n+p+1$. Once the integral history stack is filled with q elements that satisfying Assumption 2, we can update the stacks based on whether replacing the old element with the new one is beneficial. As stated in equation (3.47), the rate of convergence can be characterized by Q_θ. The derivative of the Lyapunov function of an inactive subsystem fulfills the following condition

$$\dot{V} = -\tilde{\theta}^\mathsf{T} Q_\theta \tilde{\theta} \leq -\lambda_{\min}(Q_\theta)\tilde{\theta}^\mathsf{T}\tilde{\theta}. \tag{3.53}$$

Thus, the target of managing the integrated history stack is to maximize the minimum eigenvalue of Q_θ, such that the rate of convergence is maximized. Obviously, it is beneficial to replace the old element with the new one only if this increases the minimum eigenvalue of Q_θ. The methods for accomplishing the task have been proposed by [65] and [86]. In our approach, we apply the eigenvalue maximization method from [65].

3.2.4. Polyhedral Partition Estimation

In this section, we consider the estimation of polyhedral partitions. The straightforward solution is to obtain the polyhedral partition with the state vectors around the estimated switching instants. Then, polyhedral partitions can be estimated through the linear discrimination methods. Some algorithms have already been proposed for solving the problem but most of them are designed for the discrete switching systems [87–89].

However, even if Algorithm 3 can be improved to acquire the switching instants directly via the switching detection, their precision still can not be guaranteed. It follows that the estimated switching instants are not precise enough under the noisy condition due to the adoption of steady-state detectors. The precision of the estimated switching instants is quite strictly since a slightly uncertainty in the instants will cause great deviation of the state vectors. Then, the great biases for the polyhedral partition estimation is unavoidable. Therefore, the hyperplanes obtained via the estimated switching instants are infeasible for the PWA system identification.

To handle this problem, we present another method to estimate the polyhedral partitions after obtaining the estimated switching sequence and subsystem parameters. The method is

built on the designed cost function that includes these estimates. Actually, this problem is quite primary while acknowledging these conditions. As presented in Algorithm 3, there are some time intervals whose recognized active modes are $\hat{\sigma} = 0$. Let $[t_i^0, t_j^0]$ denote the interval with unrecognized active mode while switching from i-th mode to j-th mode. Obviously, the switching time instant locates in the time interval $[t_i^0, t_j^0]$. Therefore, the switching instant $\hat{t}$ can be obtained by searching the minimum of the following cost function.

$$J(t) = \int_{t_i^0}^{t} ||\check{x}_i(\tau) - x(\tau)||_2 d\tau + \int_{t}^{t_j^0} ||\check{x}_j(\tau) - x(\tau)||_2 d\tau, t \in [t_i^0, t_j^0]. \tag{3.54}$$

where $\check{x}_i(t)$ is the state generated by the estimated i-th system. Since the solution of the optimization problem locates inside the interval $[t_i^0, t_j^0]$, the population optimization algorithms such as genetic algorithm, ant colony, or particle swarm optimization are feasible for solving the problem [90]. Following the estimated switching instants, it is convenient to obtain the polyhedral partitions via support vector machine or linear regression since the switching sequence and the state vectors around switching instants are acknowledged.

3.3. Simulation Studies

In this section, a series of simulations are conducted to validate the effectiveness of the proposed online identification approach.

3.3.1. Numerical Experiment

In the numerical experiment, we consider the following PWA system:

$$\dot{x}(t) = \begin{cases} A_1 x(t) + B_1 u(t) + f_1 \text{ if } \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} \in \mathcal{X}_1, \\ A_2 x(t) + B_2 u(t) + f_2 \text{ if } \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} \in \mathcal{X}_2, \\ A_3 x(t) + B_3 u(t) + f_3 \text{ if } \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} \in \mathcal{X}_3. \end{cases} \tag{3.55}$$

The parameters of each subsystem are

$$A_1 = \begin{bmatrix} 0 & 1 \\ -2 & -1 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0 \\ 1.5 \end{bmatrix}, \quad f_1 = \begin{bmatrix} 0 \\ 0.4 \end{bmatrix},$$
$$A_2 = \begin{bmatrix} 0 & 1 \\ -2.5 & -1 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 0 \\ 1.5 \end{bmatrix}, \quad f_2 = \begin{bmatrix} 0 \\ 0.2 \end{bmatrix},$$
$$A_3 = \begin{bmatrix} 0 & 1 \\ -1.5 & -1 \end{bmatrix}, \quad B_3 = \begin{bmatrix} 0 \\ 1.5 \end{bmatrix}, \quad f_3 = \begin{bmatrix} 0 \\ -0.3 \end{bmatrix},$$

and polyhedral partitions $\mathcal{X}_i$ with

$$\mathcal{H}_1 = \begin{bmatrix} 1 & 0 & 0 & -2 \\ -1 & 0 & 0 & -2 \end{bmatrix}, \quad \preceq_{[1]} = \begin{bmatrix} \leq \\ \leq \end{bmatrix},$$
$$\mathcal{H}_2 = \begin{bmatrix} -1 & 0 & 0 & 2 \end{bmatrix}, \quad \preceq_{[2]} = \begin{bmatrix} < \end{bmatrix},$$
$$\mathcal{H}_3 = \begin{bmatrix} 1 & 0 & 0 & 2 \end{bmatrix}, \quad \preceq_{[3]} = \begin{bmatrix} < \end{bmatrix}.$$

In the simulation, the time step is $0.01s$ and the system is excited with the following input signal

$$u(t) = \sum_{k=1}^{2} \sin(\omega_k t) + \bar{u}(t), \tag{3.56}$$

where $\omega_1 = \pi$rad/s, and $\omega_2 = 2\pi$rad/s. The signal $\bar{u}(t)$ switches every $50s$ between the values $\bar{u}_1 = 0$, $\bar{u}_2 = 6$, and $\bar{u}_3 = -6$ to excite the system into the three corresponding partitions. The switching sequence is $1 \to 2 \to 1 \to 3 \to 1 \to 2 \cdots$. In addition, the measured state is corrupted by a state-measurement white noise which is distributed as $\mathcal{N} \sim (0, 10^{-3})$ [44]. Figure 3.4 shows the state space trajectory of the PWA system (3.55) excited by the input (3.56) from $t = 0s$ to $t = 400s$.

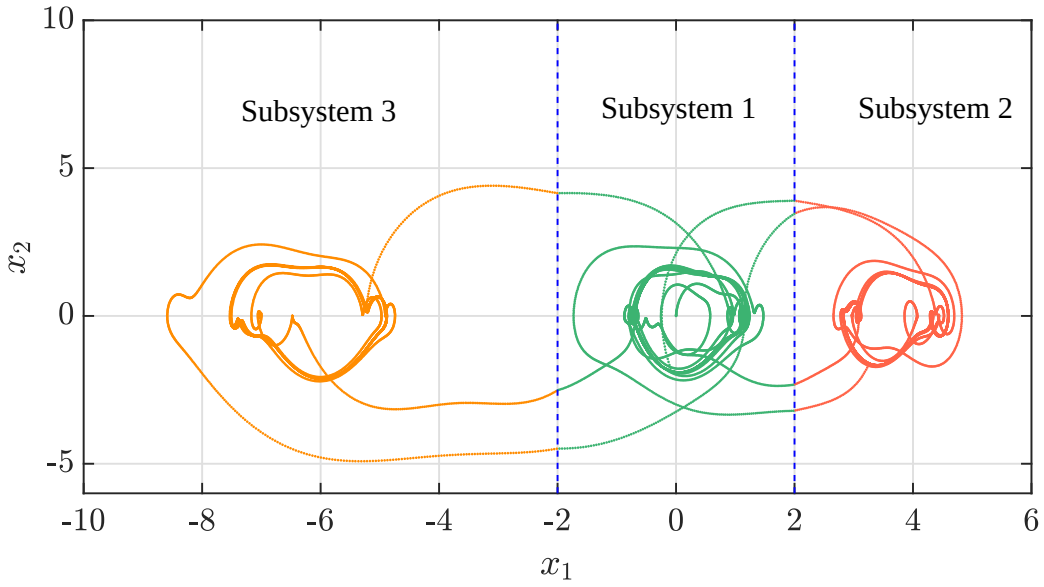


Figure 3.4.: State space trajectory (red green orange) generated by the PWA system (3.55), whose color represents the corresponding subsystem. The true hyperplanes of the system (3.55) are illustrated by the dashed lines.

In the simulation, we set the coefficients for estimating active mode as: the delay shift $t_e = 6s$, the sampling interval $t_s = 0.1s$, the sampling size $N = 6$, the moving data window size $\eta = 2s$, the threshold $\rho = 0.2$, $\gamma = 0.01$, the regularized parameter $\alpha = 10$. Clearly, the input signal and the delay shift t_e in the simulation satisfy the periodic condition in Theorem 2. Also, the piecewise input signal guarantees the activation of each subsystem and the dwell time is long enough for identification.

By implementing Algorithm 1, the STM $\hat{\Phi}(t)$ is estimated. Every entry of the matrix $\hat{\Phi}(t)$ with respect to time is plotted with the solid line in different colors in the Figure 3.5. Note that $\hat{\Phi}(t)$ before $6.6s$ are nonexistent due to the delay between the active mode recognition and the switching occurrence. The figure indicates that the estimated STM $\hat{\Phi}(t)$ is fixed while no switching occurs, whose value reveals the active mode as well.

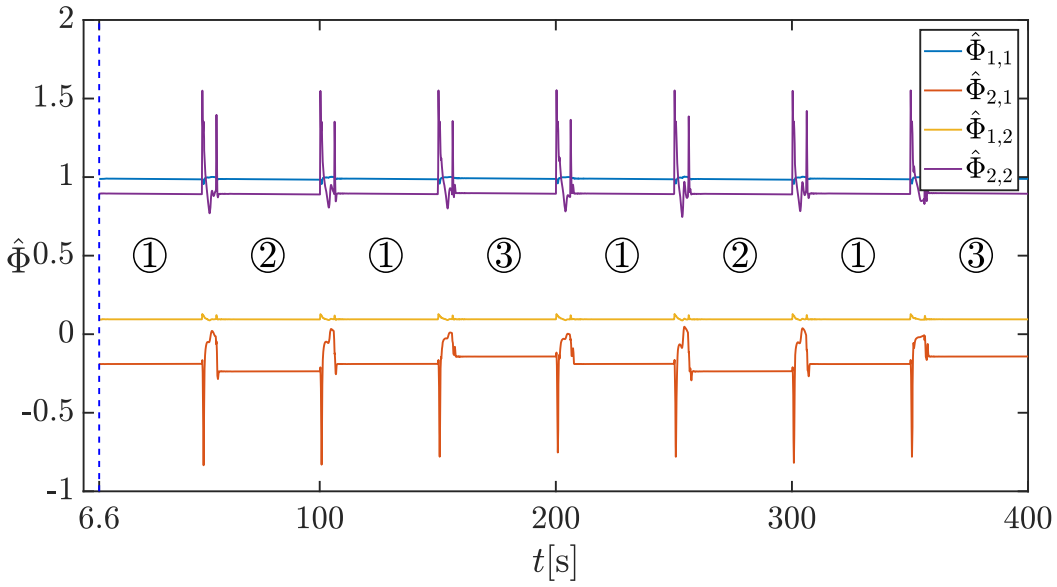


Figure 3.5.: Every entry (red line, blue line, purple line, and yellow line) in estimated state transition matrix $\hat{\Phi}(t)$ from $0s$ to $400s$, whose steady value reveals the mode of active subsystem ,i.e., ①, ②, and ③. Meanwhile, the $\hat{\Phi}(t)$ before $6.6s$ are nonexistent due to insufficient data for their calculation.

Subsequently, the active modes from $t = 0s$ to $t = 400s$ are estimated by implementing the proposed approach. The true and estimated active modes are plotted in Figure 3.6(a) and 3.6(b), respectively. In Figure 3.6(b), the active modes of a short interval after the switching occurs is $\hat{\sigma(t)} = 0$. It follows that there exists a time delay between the occurrence of switching and the correct recognition of the active mode. The correct recognition requires sufficient information from the new subsystem. Thus, it is infeasible to recognized the new active mode after the switching immediately. We assign the estimated active mode into 0 manually in this short interval after the switching occurs. Naturally, the state vectors in the interval is ignored during the subsystem parameter estimation.

For the generalized integral concurrent learning identifier, all initial estimated parameters are set to zero matrices or vectors. The coefficients for the state observer and update law are $\gamma_1 = \gamma_2 = \gamma_3 = 1$, and

$$A_m = \begin{bmatrix} -10 & 0 \\ 0 & -10 \end{bmatrix}, \quad P = \begin{bmatrix} 0.05 & 0 \\ 0 & 0.05 \end{bmatrix}. \tag{3.57}$$

Each integrated history stack contains $q = 4$ elements. Initially, the integrated history stacks of all subsystems are empty. The integrated elements $X(t_{i_k})$ are calculated with the integrated window size $\Delta t = 1s$. Once an integrated history stack is full, the elements are

managed with the method proposed by [65]. For the update law (3.37), the control gain is $k_{\mathrm{CL}} = 0.6$ once the integrated history stack is full and fulfills Assumption 2. Before that, the control gain $k_{\mathrm{CL}} = 0$ and the update law without concurrent learning are used.

In addition, the parameter error $\tilde{\theta}_i$ of each subsystem w.r.t time are depicted in Figure 3.7. Simulation results show a fast convergence of all estimated parameters to true parameters as expected. Moreover, Figure 3.7 illustrates that the adaptation is carried out no matter the corresponding subsystem is active or not.

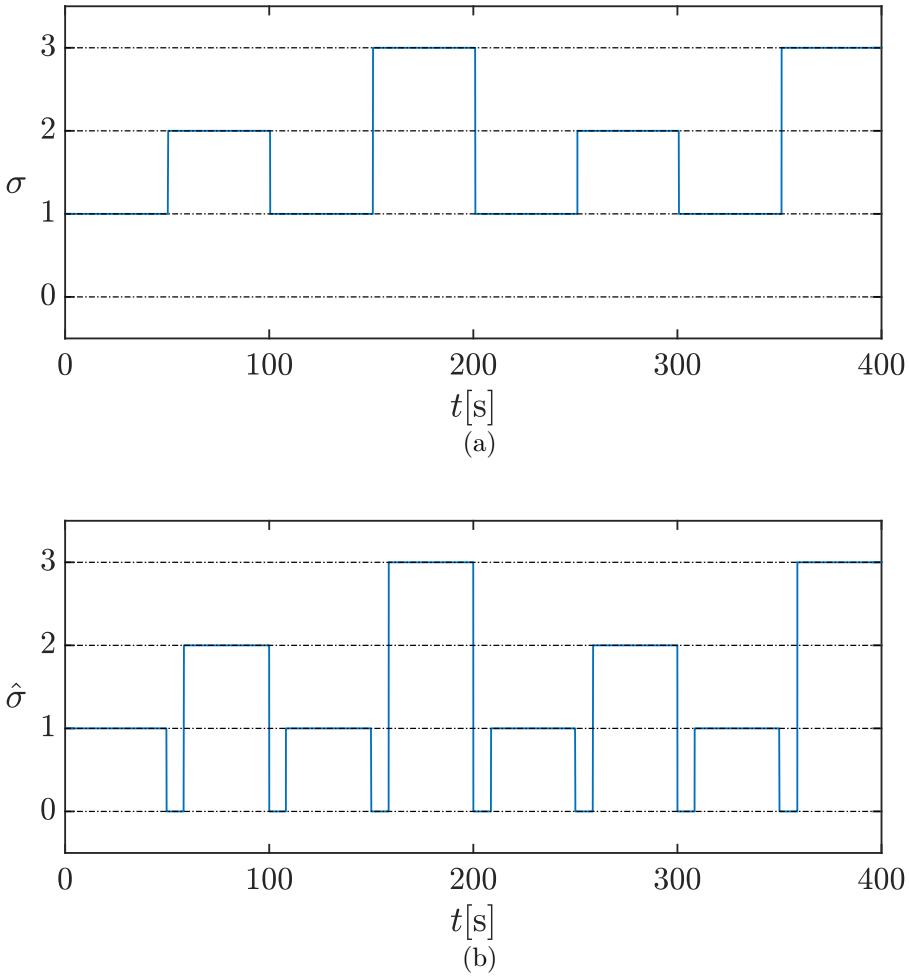


Figure 3.6.: The true active mode $\sigma(t)$ (3.6(a)) and estimated active mode $\hat{\sigma}(t)$ (3.6(b)) of the PWA system in the simulation.

Following the estimation of the subsystem parameters, we identify the switching time instants of the PWA system (3.55) by solving the optimization problem in Section 3.2.4. In the simulation, the sequence of switching instants T_{sw} and the estimated corresponding active mode $\hat{\sigma}$ before and after them are shown in Table 3.1. Then, the parameters of the trained hyperplanes are re-scaled and listed in Table 3.2. These tables indicate that the subsystem parameters and the polyhedral partitions are well-estimated.

The performance of the proposed approach is evaluated by the modified best fit rate (BFR)

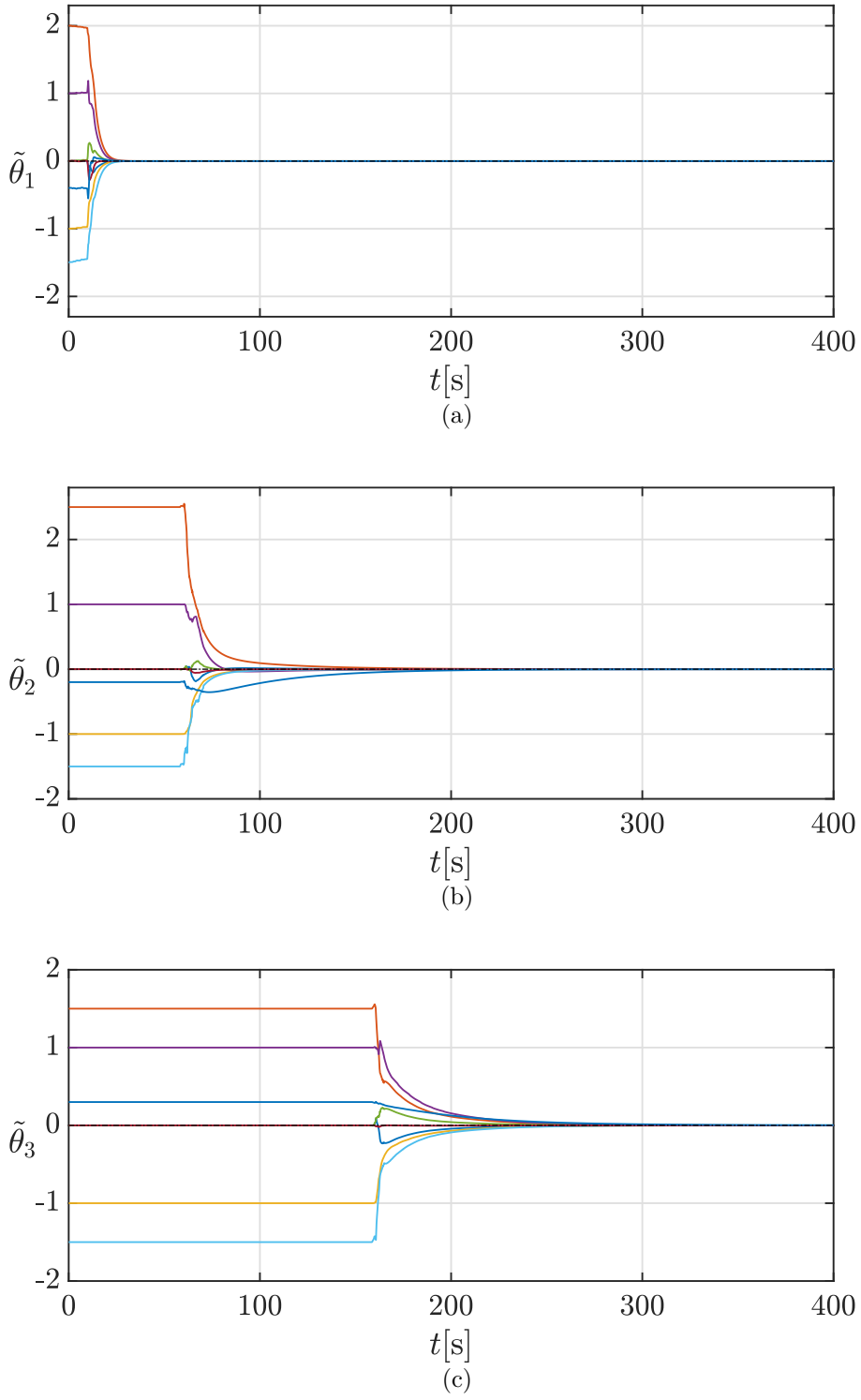


Figure 3.7.: The estimated parameter error $\tilde{\theta}_i$ of each subsystem in the simulation. 3.7(a), 3.7(b), and 3.7(c) are the estimated error of subsystem 1, 2 and 3, respectively. Every entry of the estimated error vector is visualized by solid lines with different colors in every sub-figure.

Table 3.1.: Sequence of switching instants T_{sw} and the corresponding active mode $\hat{\sigma}$ before and after them

$\hat{\sigma}$	T_{sw}[s]	$\hat{\sigma}$	T_{sw}[s]	$\hat{\sigma}$	T_{sw}[s]
1			150.85	2	
	50.54	3			300.76
2			200.99	1	
	100.45	1			351.06
1			251.01	3	

Table 3.2.: True hyperplane parameters (h_1,h_2) and estimated hyperplane parameters ($\hat{h}_1$, $\hat{h}_2$) in the simulation

h_1	$\hat{h}_1$	h_2	$\hat{h}_2$
1	1	1	1
0	-0.014	0	-0.029
0	-0.031	0	-0.04
-2	-2.105	2	1.9577

which is defined as follows

$$\text{BFR} = \max\left\{1 - \frac{||\mathbf{x}_i - \hat{\mathbf{x}}_i||}{||\mathbf{x}_i - \bar{x}\mathbf{1})||}, 0\right\} \times 100\%, \tag{3.58}$$

where i is the index of the entry in state vectors. $\mathbf{x}_i \in \mathbb{R}^N$ is the sequence of the i-th entry in true state vector x. $\hat{\mathbf{x}}_i \in \mathbb{R}^N$ is the sequence of i-th entry in estimated state vector $\hat{x}$. Additionally, $\bar{x} \in \mathbb{R}$ is the mean of $\mathbf{x}_i$ and $\mathbf{1}$ is an all-ones vector with suitable dimension.

The BFR is originally designed to evaluate the performance of the approach based on the true and estimated outputs, which are not provided in our simulation [58]. Therefore, we do slight modification to the equation of BFR and utilize the state vectors to evaluate performance instead of the outputs in the simulation. The replacement can also be regarded as computing the BFR of the PWA system (3.55) while the system is with the output matrix [0 1] or [1 0] and without a direct feed-through.

In the evaluation, the true and estimated PWA systems are excited by the same input signal, which is a random signal between $[-4, 4]$. The error between the state-vector x and the estimated state-vector $\hat{x}$ from $0s$ to $400s$ are plotted in Figure 3.8. Meanwhile, we replace the output sequence with the sequence of each entry x_1 and x_2 in state vectors to calculate the BFR. The BFR values of the estimated $\hat{x}_1$ and $\hat{x}_2$ are 97.21% and 97.77%. The values imply the satisfying accuracy of the proposed approach.

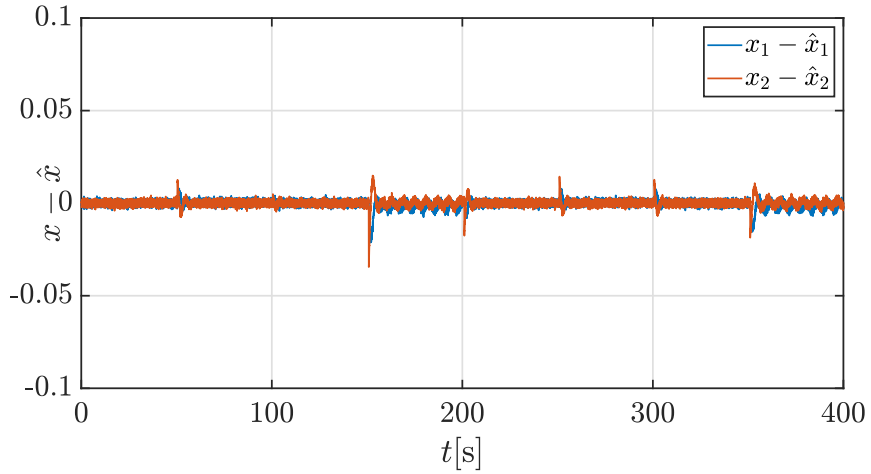


Figure 3.8.: Errors of the true state-space vector x and estimated state-space vector $\hat{x}$ in the evaluation.

We also discuss the choice of α and elaborate that a relatively large value is suitable. To justify that the choice of α is quite loose in our approach, the BFR values of our approach with respect to different α are depicted in Figure 3.9. The figure reveals that a regularized parameter $\alpha \in [0.1\ 19]$ provides the satisfying accuracy for the proposed approach.

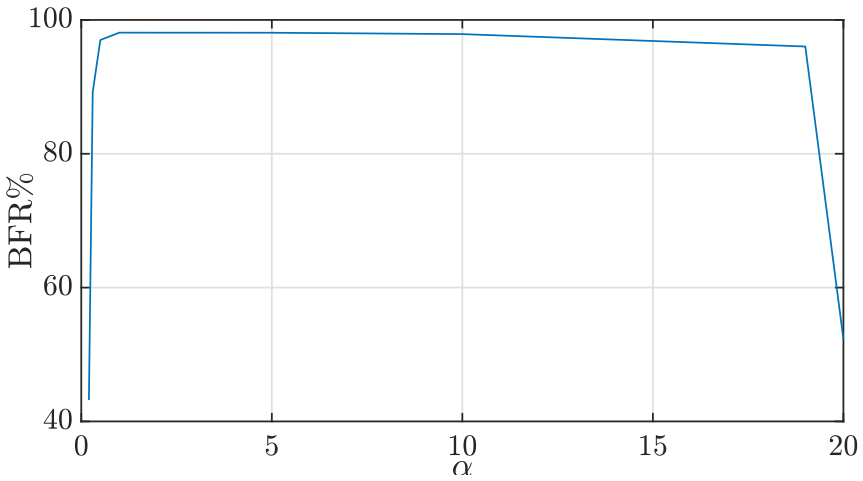


Figure 3.9.: The regularized parameter α with respect to BFR value.

3.3.2. Wheeled Mobile Robot System

In this section, we evaluate the effectiveness of the online identification approach on the wheeled mobile robot (WMR) shown in Figure 3.10. This example is presented in the paper [43] and the following simulations not only evaluate the effectiveness of our approach but also compare it with the online fault detection and identification (FDI) technique proposed in [43]. The WMR is assumed to be rigid and it is driven by a torque T_q to control the

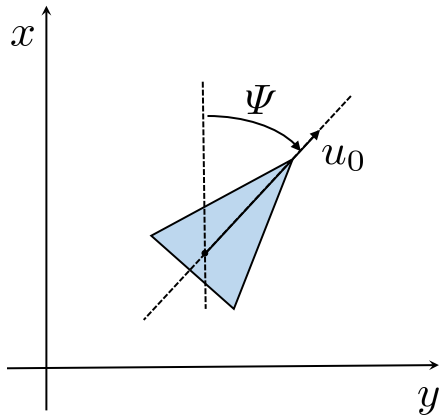


Figure 3.10.: Schematic representation of the wheeled mobile robot

heading angle Ψ. The kinematic and dynamic equations for the WMR are

$$\begin{aligned} \dot{y} &= u_0 \sin\Psi, \\ \dot{\Psi} &= R, \\ \dot{R} &= 0.75\frac{1}{I}T_q, \end{aligned} \tag{3.59}$$

where u_0 is the forward velocity of the robot and assumed to be constant, by designing appropriately a cruise controller. The heading angle Ψ is measured with respect to the positive x-axis in the inertial frame. In the equation (3.59), 0.75 is the unknown actuator effectiveness, and $I = 1\text{kg} \cdot \text{m}^2$ (which is known) corresponds to the moment of inertia of the WMR with respect to the center of its mass. Inspired by this example, the actual system can be considered as the PWA system in the form

$$\dot{x} = \begin{cases} A_1 x + B_1 u + f_1 & \text{if } \begin{bmatrix} x \\ u \end{bmatrix} \in \mathcal{X}_1, \\ A_2 x + B_2 u + f_2 & \text{if } \begin{bmatrix} x \\ u \end{bmatrix} \in \mathcal{X}_2, \end{cases} \tag{3.60}$$

where $[x, u] = [x_1, x_2, x_3, u] = [y, \Psi, R, T]$.

The parameters of each subsystem are

$$A_1 = \begin{bmatrix} 0 & \frac{2}{\pi}u_0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0 \\ 0 \\ \frac{0.75}{I} \end{bmatrix}, \quad f_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix},$$

$$A_2 = \begin{bmatrix} 0 & -\frac{2}{\pi}u_0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0 \\ 0 \\ \frac{0.75}{I} \end{bmatrix}, \quad f_2 = \begin{bmatrix} 2u_0 \\ 0 \\ 0 \end{bmatrix},$$

with $u_0 = 1$. The aforementioned matrices arise from approximating, in the range $[-\pi/2, 3\pi/2]$. As a consequence, the switching surface between the two subsystems is given by

$$\mathcal{H}_1 = \begin{bmatrix} 0 & \frac{2}{\pi} & 0 & 0 & -1 \end{bmatrix}, \quad \preceq_{[1]} = \begin{bmatrix} \leq \end{bmatrix},$$
$$\mathcal{H}_2 = \begin{bmatrix} 0 & -\frac{2}{\pi} & 0 & 0 & 1 \end{bmatrix}, \quad \preceq_{[2]} = \begin{bmatrix} < \end{bmatrix}.$$

In the simulation, we use the same initial state and input signal from [43]. The initial state is taken as $x_0 = [1, \pi/2, 0]^\intercal$ and the input is a series of steering and counter-steering sinusoids at frequency 0.2, 0.8, and 1.6 rad/s.

Meanwhile, the coefficients for our approach is set as: the delay shift $t_e = 1.6s$, the sampling interval $t_s = 0.1s$, the sampling size $N = 6$, the moving data window size $\eta = 2s$, the threshold $\rho = 0.01$, $\gamma = 0.001$, the regularized parameter $\alpha = 0.1$. Clearly, the input signal and the delay shift t_e in the simulation satisfy the periodic condition in Theorem 2. The coefficients for the state observer and update law are the same as the previous example. The simulation results are described in Figure 3.11. The figure shows the estimated parameter error $\tilde{\theta}_i$ of each subsystem w.r.t time. Clearly, the consequences indicate the effectiveness of the proposed approach while identifying the WMR system.

In order to check the consistency of the approach and the possibility of getting trapped into local minima, we have selected many initial estimate $\hat{\theta}(0)$ for the test [43]. The performance of the approach w.r.t different initial estimates are shown in Table 3.3. Table 3.3 shows the

Table 3.3.: Performance depending on the initial estimate

$\mathrm{Var}(\theta - \hat{\theta}(0))$	$\mathrm{Avg}\|\|\theta - \hat{\theta}_{st}\|\|/\|\|\theta\|\|$
0.03	0.21%
0.1	0.41%
0.3	0.53%
1.0	0.49%
3.0	0.30%

distance between the true and the estimated parameters (at steady state) $||\theta - \hat{\theta}_{st}||/||\theta||$ as a function of the variance of $\theta - \hat{\theta}(0)$. The table illustrates that the steady-state distance keeps a similar level, regardless of the initial estimate $\hat{\theta}(0)$. On the contrary, the performance of online FDI in [43] shows that the steady-state distance also increases while the initial estimate is very far from the true parameter. It follows that the online FDI is based on the Gauss-Newton algorithm, which may not converge to the actual parameters. Actually, this is also the drawback of the online FDI technique proposed in [43].

In this section, the effectiveness of the proposed approach is validated via identifying the WMR system. In addition, the comparison between the proposed approach and the online FDI technique is also presented in this section. Compare to the online FDI technique, the proposed approach is not sensitive to the initial parameters.

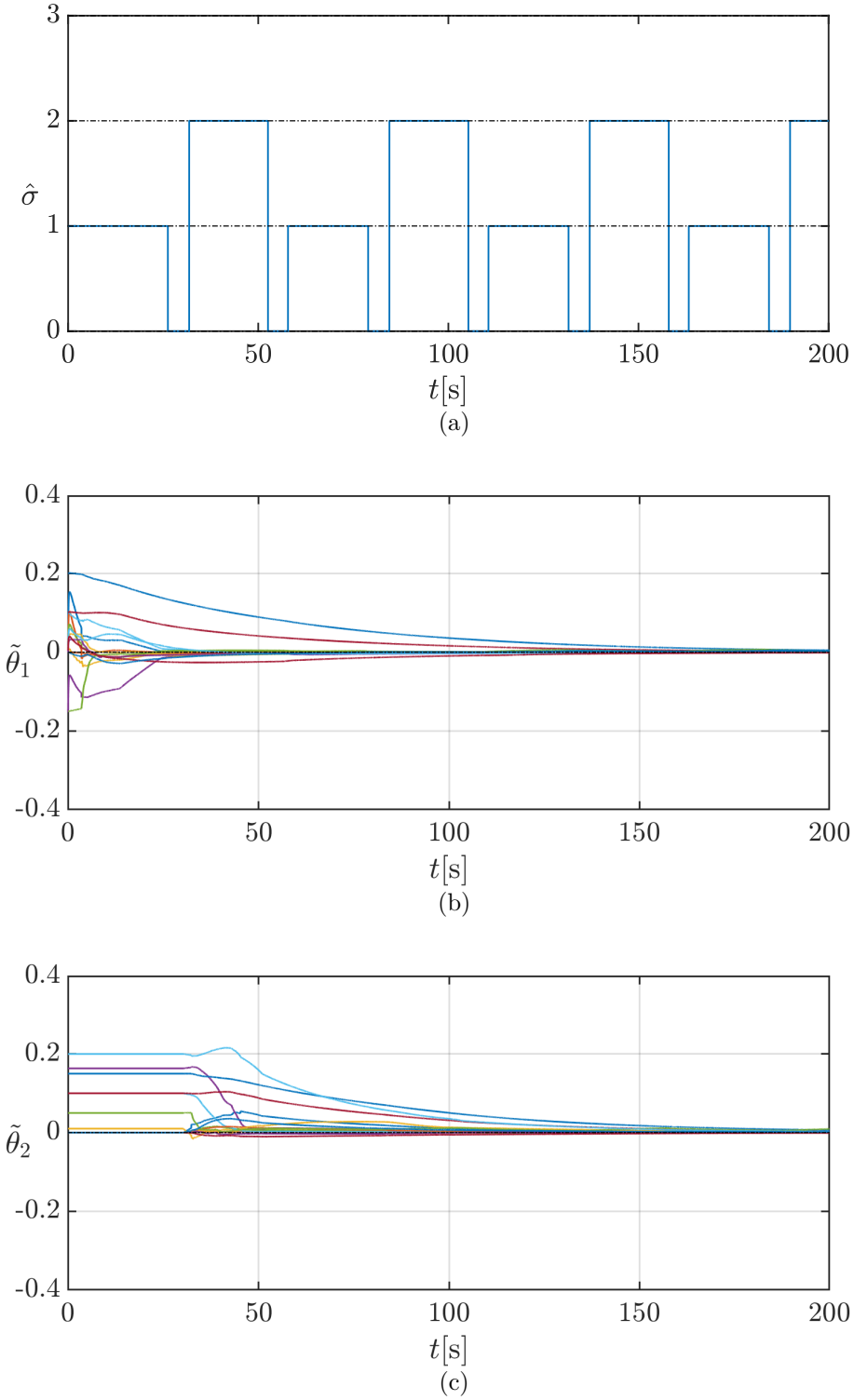


Figure 3.11.: The estimated active mode $\hat{\sigma}(t)$ and the estimated parameter error $\tilde{\theta}_i$ of each subsystem. 3.11(a) is the estimated active mode $\hat{\sigma}(t)$. 3.11(b) and 3.11(c) are the estimated error of subsystem 1 and 2, respectively. Every entry of the estimated error vector is visualized by solid lines with different colors in every sub-figure.

3.3.3. Extreme Situation: Rank-deficient Sampling Matrices

To validate the effectiveness of the proposed refinement strategy, we carry out a simulation with the following PWA system.

$$\dot{x}(t) = \begin{cases} A_1 x(t) + B_1 u(t) + f_1 \text{ if } \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} \in \mathcal{X}_1, \\ A_2 x(t) + B_2 u(t) + f_2 \text{ if } \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} \in \mathcal{X}_2. \end{cases} \tag{3.61}$$

The parameters of each subsystem are

$$A_1 = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 1 \\ 1.5 \end{bmatrix}, \quad f_1 = \begin{bmatrix} 0 \\ 0.4 \end{bmatrix},$$
$$A_2 = \begin{bmatrix} 0 & 1 \\ -2 & -1 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 1 \\ 1.5 \end{bmatrix}, \quad f_2 = \begin{bmatrix} 0 \\ 0.2 \end{bmatrix},$$

and polyhedral partitions $\mathcal{X}_i$ with

$$\mathcal{H}_1 = \begin{bmatrix} 1 & 0 & 0 & -2 \end{bmatrix}, \quad \preceq_{[1]} = \begin{bmatrix} \leq \end{bmatrix},$$
$$\mathcal{H}_2 = \begin{bmatrix} -1 & 0 & 0 & 2 \end{bmatrix}, \quad \preceq_{[2]} = \begin{bmatrix} < \end{bmatrix}.$$

The state matrix of the first subsystem in PWA system (3.61) is **identity matrix**, which is a rare case for real systems. As aforementioned, for the identity state matrix, the row rank of the augmented matrix $[0\ 0\ \varepsilon_0]$ is always 1 regardless of ε_0 and the sampling matrix for the first subsystem is always row rank-deficient. Therefore, we implement the refinement strategy to provide distinct ε_0 for the augmented matrix such that the reformulated matrix is full row rank.

In the simulation, the time step is $0.01s$ and the system is excited with the following input signal

$$u(t) = \sum_{k=1}^{3} \sin(\omega_k t) + \bar{u}(t), \tag{3.62}$$

where $\omega_1 = 2\pi$rad/s, $\omega_2 = 4\pi$rad/s, and $\omega_3 = 8\pi$rad/s. The signal $\bar{u}(t)$ switches every $50s$ between the values $\bar{u}_1 = 0$, $\bar{u}_2 = 6$ and excite the system into the two corresponding partitions. The switching sequence is $1 \to 2 \to 1 \to 2 \cdots$. In addition, the delay shift $t_e = 1s$, the sampling interval $t_s = 0.2s$, the sampling size $N = 5$ are used in the simulation.

The estimated STM without the refinement strategy is shown in Figure 3.12(a). The estimated STM $\hat{\Phi}$ of the first subsystem are not unique due to the row rank-deficient of the augmented matrix $\text{rank}[e^{At_s} - \lambda I \quad \varepsilon_0]$. Therefore, we adopt the refinement strategy after detecting the row rank-deficient $\Xi(t)$. The input signal is adjusted at 3s after the switching occurs. The adjusted input signal is given by

$$u(t) = \sum_{k=1}^{3} \cos(\omega_k t) + \bar{u}(t), \tag{3.63}$$

where $\omega_1 = 2\pi\text{rad/s}$, $\omega_2 = 4\pi\text{rad/s}$, and $\omega_3 = 8\pi\text{rad/s}$. Note that only the phase of the input signal is adjusted to guarantee the unchanged active mode. Meanwhile, the sampling matrices corresponding to the original and adjusted input signals are recorded. These matrices reformulate a full row rank matrix to calculate the initial STM of the first subsystem. Subsequently, $\hat{\Phi}$ is updated based on the initial STM. The estimated STM $\hat{\Phi}$ with the refinement strategy is shown in Figure 3.12(b). Clearly, $\hat{\Phi}$ is unique and can be employed for the active mode recognition with the refinement strategy.

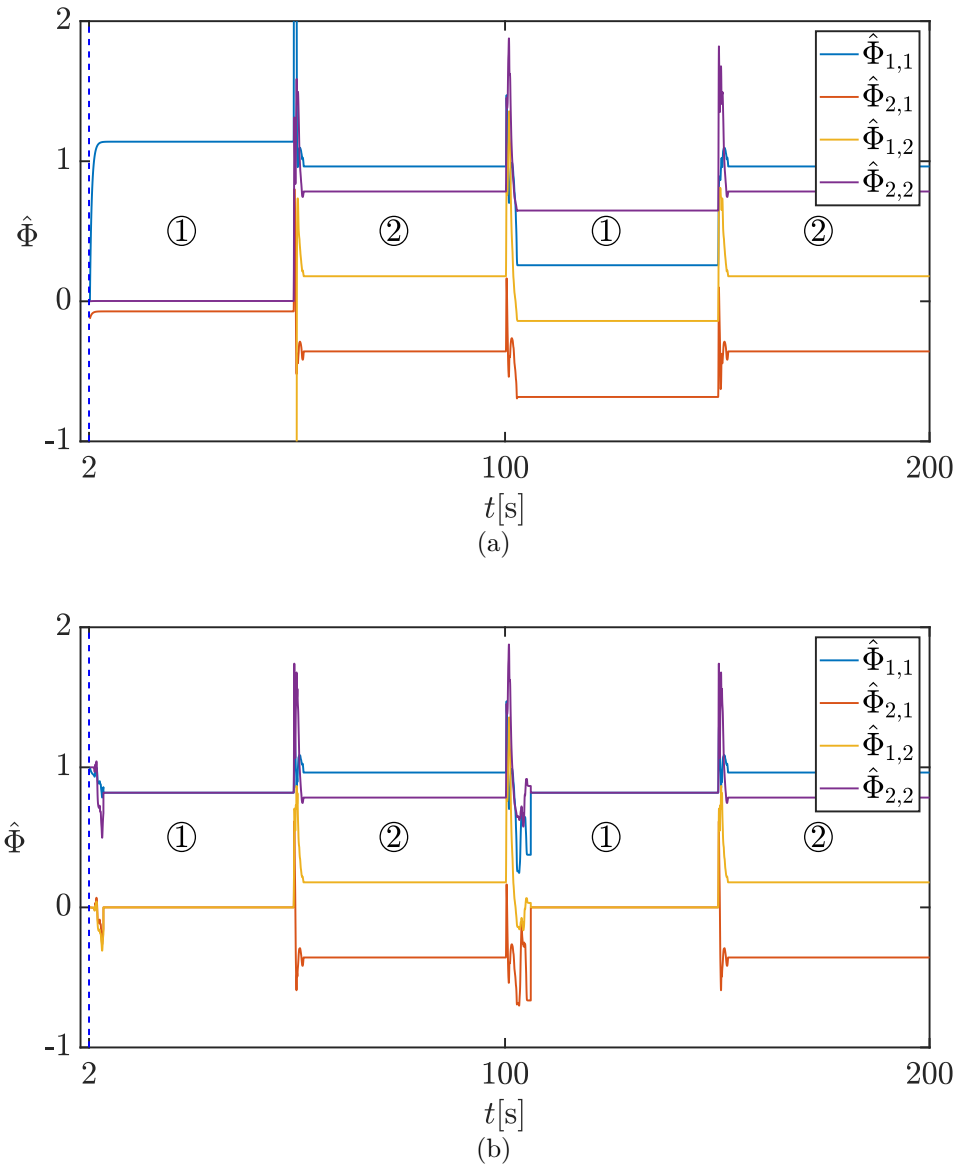


Figure 3.12.: Estimated STM $\hat{\Phi}(t)$ without the refinement strategy (3.12(a)) and with refinement strategy (3.12(b)) from $0s$ to $200s$. The number ①, and ② indicate the active modes. Meanwhile, the $\hat{\Phi}(t)$ before $2s$ are nonexistent due to insufficient data for their calculation.

3.4. Summary

The major concern of this chapter is the online identification of continuous-time PWA systems in state-space form. Taking into consideration the formulation of the continuous-time PWA system, we decompose the identification task into three sub-tasks: recognition of the active mode, estimation of the subsystem parameters, and identification of the polyhedral partitions. Then, we propose a novel approach to cope with the online identification of continuous-time PWA systems.

For the first sub-task, we propose an online algorithm to detect switching and recognize the active mode. The algorithm is evolved from the defined delay error, i.e, the error between the current and previous states. The active mode recognition is then accomplished by analyzing the dynamics of the delay error which is excited by an appropriately designed input signal. Moreover, there is an unavoidable delay between the occurrence of switching and the correct recognition of active mode since the confirmation of the new active mode requires sufficient information from the new subsystem. To guarantee the precision of the subsystem identification, the active mode during the delay is manually assigned as zero and the corresponding state-input vectors are skipped in the subsystem identification. In addition, we discuss the extreme situations that lead to the unrecognized active modes and provide a refinement strategy to handle the situations. The first sub-task is heretofore accomplished.

According to the recognized active mode, the generalized I-CL identifier is derived to identify the parameters of each subsystem. In the identifier, we impose the Luenberger-observer for each subsystem to predict the states. In order to ensure the stability of the hybrid system with the multiple-Lyapunov-functions concept, the predicted state of inactive subsystems needs to be reset and kept equal to the actual state measurements, i.e., state-reset mechanism. The update law for the parameters of each subsystem is then designed based on the predicted states. Stability of the update law is ensured by Lyapunov analysis, while the convergence is guaranteed under the assumption of integrated history stacks.

Subsequently, the third sub-task is accomplished by optimizing the proposed cost function while knowing the switching sequence and the estimated subsystem parameters. The optimization can be summarized as searching for the optimal switching instants such that the state error is minimized. The state-input vectors before and after the optimal switching instants are then extracted to estimated the polyhedral partitions.

Following the description of the proposed method, we do a series of simulations to verify the effectiveness of our method. The first and second simulations show the capability of the proposed method via a numerical experiment and a WMR system, respectively. Then, the performances of the refinement strategy to handle the extreme situations are illustrated in the third simulation. In addition, the proposed framework is generalized while identifying other continuous-time hybrid systems with linear/affine subsystems, such as PWL systems, SLSs, and PWA systems. The proposed active mode recognition algorithm identifies their discrete dynamics behaviors while the generalized I-CL identifier confirms their continuous ones. Note that the limitation of identifying the linear/affine hybrid systems is caused by the delay errors in the proposed active mode recognition algorithm.

In summary, we propose a novel framework for online identification of continuous-time PWA systems, including the number of subsystems, parameters of each subsystem, and polyhedral partition of each subsystem. The effectiveness of the approach is verified in the extensive simulations.

Online Identification of Switched Nonlinear Systems

4.

In this chapter, we describe the enhanced framework for the online identification of switched nonlinear systems. The dynamics of the SNSs are decided by the nonlinear subsystem and the switching signals. Thus, the main challenge of the online identification framework is how to handle the estimation of these nonlinear subsystems. The framework for the online identification of continuous-time SNSs includes the switching signal estimation and the subsystem identification. The switching signal estimation is decomposed into the switching detection and the active mode recognition tasks, which are accomplished via the proposed algorithm. Meanwhile, the identification of subsystems not only requires to obtain the parameters but also their model structures. Therefore, we use the modified I-CL identifier to update the subsystem parameters and propose a refinement strategy to realized the MSS of the nonlinear subsystems.

4.1. Introduction to Switched Nonlinear Systems and Problem Formulation

The switched nonlinear system is a representative class of hybrid system, whose dynamics is determined by the nonlinear subsystems and the switching signal. Compare to PWA systems, the SNSs require fewer subsystems to describe a complex system, whereas their nonlinear subsystems are more complicated. In some scenarios, the small number of local subsystems greatly alleviates the computational complexity of identifying these complex systems. In addition, nonlinear hybrid systems in the state-space form are more intuitive to depict the internal states of the systems so as to design the corresponding control strategies.

4.1.1. Continuous-Time Switched Nonlinear Systems

The continuous-time switched nonlinear systems in state-space form are defined as follows

$$\dot{x}(t) = \begin{cases} f_1(x(t), u(t)), & \text{if } \sigma(t) = 1, \\ \vdots & \vdots \\ f_s(x(t), u(t)), & \text{if } \sigma(t) = s, \end{cases} \tag{4.1}$$

where $x \in \mathbb{R}^{n_x}$, $u \in \mathbb{R}^{n_u}$ are the state and input vectors, respectively. $\sigma(t) \in \{1, 2, \ldots, s\}$ is the switching signal that determines the sequence of activation among s subsystems. Correspondingly, each nonlinear subsystem $f_i(\cdot)$ can be expressed as a linear combination of basis functions $\varphi_j(x(t), u(t)), j = 1, \ldots, n_b$ with the respective parameter vectors $\kappa_{i,1}, \ldots, \kappa_{i,n_b}$ [91],

i.e.,

$$f_i(x(t), u(t)) = \sum_{j=1}^{n_b} \kappa_{i,j} \varphi_j(x(t), u(t)), \tag{4.2}$$

where $\kappa_{i,j} \in \mathbb{R}^{n_x}$. For the sake of convenience, we use the notation φ_j to represent the j-th basis function. Then, the i-th nonlinear subsystem can be written as the following matrix form

$$\dot{x}(t) = f_i(x(t), u(t)) = \Theta_i^\mathsf{T} \psi(t), \tag{4.3}$$

where all basis functions are collected in the regression vector $\psi(t) = [\varphi_1, \ldots, \varphi_{n_b}]^\mathsf{T} \in \mathbb{R}^{n_b}$ and the corresponding parameter vector $\Theta_i = [\kappa_{i,1}, \ldots, \kappa_{i,n_b}]^\mathsf{T} \in \mathbb{R}^{n_b \times n_x}$.

Typical basis functions are the linear basis function, Gaussian radial basis function, and polynomial basis function, which are selected based on the application scenarios [92, 93]. Without loss of generality, let the first n_x basis functions be the n_x entries of x, i.e., $\psi(t) = [x^\mathsf{T}(t), \varphi_{n_x+1}, \ldots, \varphi_{n_b}]$. This is a technical transformation regardless of the identification results. If the entries of the state x are not contained in the nonlinear subsystem, the corresponding parameters will converge to 0 in the identification procedure.

4.1.2. Problem Formulation and State of the Art Limitations

Following the re-written formulation of the nonlinear subsystems, the identification problem of the switched nonlinear system is now presented as follows.

Problem 3. Given a group of basis functions $\varphi_j, j = 1, \ldots, n_b$, input signal u, and measured state x in the switched nonlinear system (4.1), design an online algorithm such that estimates of the number of subsystems s, switching signal $\sigma(t)$, parameters of each subsystem Θ_i are guaranteed to converge to the true values.

In the last decades, switched nonlinear systems and the corresponding control methods have drawn considerable attention among various hybrid systems due to the ability of modelling practical systems [94–97]. The identification of switched nonlinear systems has been extensively studied as the foundation of analyzing and controlling these systems [98–100].

In the last few decades, the identification of linear or affine hybrid systems has been spotlighted due to their universal approximation properties and simple interpretations [49, 101]. However, the satisfying accuracy of describing the complex systems with linear or affine hybrid systems is generally built on the high number of local subsystems (and switchings among them) [26]. Consequently, the computational complexity of the identification methods greatly aggravates along with the increasing number of switchings. In addition, the identified linear or affine hybrid models prevent describing the intrinsic mechanism of the complex systems and hinder their physical interpretations [45]. Therefore, the identification of switched nonlinear systems has received considerable attention in these years.

For the switched nonlinear system identification, the majority of the works concentrate on handling the discrete-time switched nonlinear systems in input-output form, especially the switched nonlinear autoregressive exogenous (ARX) models [91, 102–105]. In the paper of Lauer [102], the identification of hybrid nonlinear ARX systems is recast as an unconstrained nonlinear continuous optimization problem and solved efficiently by using standard optimization methods. Breschi employs a probabilistic framework using categorical and Bernoulli distributions to identify the switched nonlinear ARX systems [104]. Very recently,

a new methodology named hybrid-sparse identification of nonlinear dynamics is developed to identify separate nonlinear dynamical regimes by imposing information theory [105]. Nevertheless, the nonlinear hybrid models in the state-space form are more intuitive to depict the internal states of the systems so as to design the corresponding control strategies [44]. Furthermore, as pointed out by Garnier [25], direct identification of continuous-time models based on sampled data can outperform the discrete-time models in case of rapidly or irregularly sampled data.

From another perspective, the aforementioned identification methods for switched nonlinear systems are offline, which suffer from high computational costs owning to the increasing number of samples. In order to solve this problem, an attempt to deal with switched nonlinear system identification recursively is reported in the paper of Bianchi [91]. This paper introduces an iterative randomized approach for the segmentation of time-ordered data observed from switched nonlinear ARX (SNARX) models. The iterative procedure is only adopted to refine the switching instants from an initial guess. In fact, the approach still offers an offline algorithm in each segmentation and the reduction of the combinatorial complexity is limited. Meanwhile, The paper of Goldar proposes a concurrent learning based identification method for the continuous-time switched nonlinear systems [74]. However, it is not a strictly online identification method due to the assumption that the switching signal is known. In general, online identification methods are capable to identify the switching signal and handle the data in sequence, which is more efficient than offline methods. Besides, the online identification method for switched nonlinear systems is also the foundation of adaptive control. The adaptive control is a control method that using the controller which must adapt to a controlled system [106–108]. It is widely used to control the nonlinear or linear systems with varying parameters and initially uncertain [109], especially for aircraft flies [110–112]. In this regard, we focus on the online identification approach for the continuous-time switched nonlinear systems in state-space form.

It is worth noticing that most works for the online identification of continuous-time affine hybrid systems are realized via the concurrent learning technique [44,73,74]. The concurrent learning technique can be summarized as using the recorded and current data concurrently for adaption in the framework of model reference adaptive control. The technique is especially promising in the field of online continuous-time affine hybrid systems identification due to providing a continuous adaptation regardless of which subsystem is activated. However, the main drawback of concurrent learning technique, i.e., the necessity of state derivatives, is also inherited by these works. A novel integral concurrent learning method is presented that circumvents the drawback while maintaining parameter convergence properties, which is limited to the adaptive control of the linearly parameterized systems [75]. In this chapter, we concentrate on extending the integral concurrent learning method to achieve the identification of continuous-time switched nonlinear systems.

4.2. Enhanced Framework for Online Identification of Continuous-Time SNSs

In this section, we propose a novel framework for the online identification of continuous-time switched nonlinear systems in state-space form. The number of subsystems, the switching sequence, and the nonlinear subsystems are identified through the proposed framework. The

identification framework consists of the algorithms for the switching signal estimation, and the subsystem identification. In addition, we design the refinement strategy for the model structure selection (MSS) to improve the proposed framework. Note that the employment of the MSS strategy is optional. The overview of the framework is illustrated through the following block diagram.

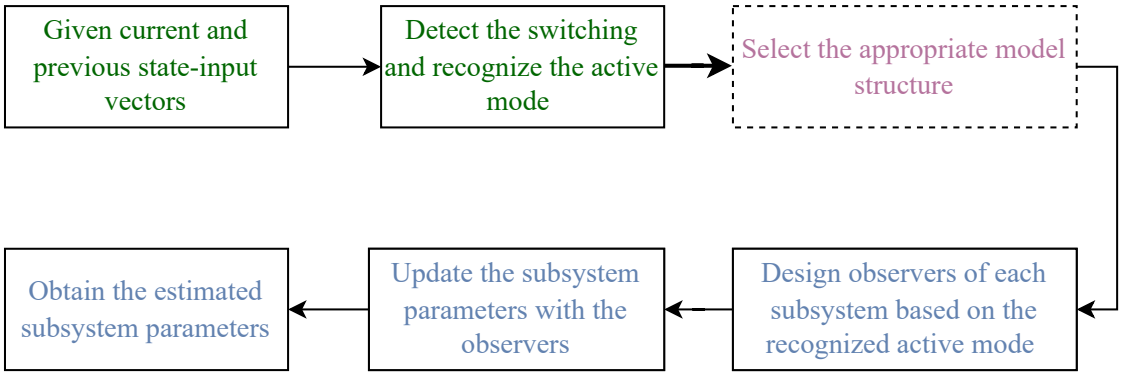


Figure 4.1.: The overview of the proposed online identification framework. The different color blocks indicate different stages in the proposed framework. The dotted line block implies the optional strategy.

Specifically, A theorem based on the projection matrix in statistics is introduced to address the identification of the number of subsystems and the switching sequence. Derived from the theorem, an online algorithm is designed to estimate the active mode, and the estimation of subsystems number is also hinged on the algorithm. Following the estimation of the active mode, the integral concurrent learning method is extended to identify the nonlinear subsystem, whose convergence proof is provided in the section. The extended identifier also retains the advantage of the traditional integral concurrent learning method, i.e., removing the need to estimate state derivatives [75]. Additionally, a refinement strategy for MSS is introduced to improve the efficiency of the proposed approach. The strategy can be integrated into the active mode estimation algorithm to determine the model structure of each subsystem before identifying the nonlinear subsystems.

The solution of Problem 3 is presented in the following sections. Specifically, the identification procedure for the switched nonlinear system comprises two stages. The first stage (Section 4.2.1) is proposed to estimate the switching signal $\sigma(t)$, including the tasks of switching detection and mode recognition. To achieve the objective, the dwell time, being the time between consecutive switchings, is required to satisfy the following assumption in the identification.

Assumption 3. The switching signal causes repeated activation of all subsystems. In addition, there exists a positive τ_Δ such that the minimum dwell time fulfills $\tau_{\text{dwell}} > \tau_\Delta$.

The delay between the switching occurrence and the correct active mode recognition is unavoidable conforming to the designed algorithm in the first stage. Therefore, Assumption 3 for minimum dwell-time should be fulfilled, where τ_Δ is the time required to recognize the active mode correctly. Note that the value of τ_Δ will be stated later. Following the estimated switching signal $\hat{\sigma}(t)$ from the first stage, the integral concurrent learning identifier is extended to identify the parameters of each subsystem in the second stage (Section 4.2.2).

4.2.1. Switching Signal Estimation

In this section, we propose the recursive algorithm for estimating the switching sequence, including the switching detection and active mode recognition. The algorithm is inspired by the data-driven projection subspace method and the principle is to generate the residual as the mode switching indicator [89]. Compare to the original projection subspace method for discrete-time systems, the proposed method is designed for continuous-time systems. Meanwhile, the assumption of the state matrix A is relaxed according to the equation (4.3) [113].

Before introducing the algorithms, it is worth noticing that the modes of switched nonlinear systems are required to be discernible. Indeed, if two modes are not discernible, it is not possible to determine which one is active and estimate the switching signal. The definition of discernability is stated as follows [89].

Definition 4. Two modes i, j are discernible for all inputs in a time interval $[0, T]$ $(T > 0)$, if for all initial states and for the same input applied in the modes i and j, the states x in the two modes i and j are different.

In the algorithm, the sampling matrices comprised by the regression vectors $\psi(t)$ are the foundation. For a time sequence $t_{i_1}, t_{i_2}, \ldots, t_{i_N}$ that satisfying $\sigma(\tau) = i, \forall\tau \in [\max(t_{i_j} - \Delta t, 0), t_{i_j}]$, where $\Delta t \in \mathbb{R}^+$ is the size of integration window, the following sampling matrices are obtained

$$\begin{aligned} \Upsilon_i &= \begin{bmatrix} \delta(t_{i_1}) & \delta(t_{i_2}) & \ldots & \delta(t_{i_N}) \end{bmatrix} \in \mathbb{R}^{n_x \times N}, \\ \Phi_i &= \begin{bmatrix} \phi(t_{i_1}) & \phi(t_{i_2}) & \ldots & \phi(t_{i_N}) \end{bmatrix} \in \mathbb{R}^{n_b \times N}, \end{aligned} \tag{4.4}$$

where

$$\begin{aligned} \delta(t) &= x(t) - x(\max(t - \Delta t, 0)), \\ \phi(t) &= \int_{\max(t-\Delta t,0)}^{t} \psi(\tau) d\tau. \end{aligned} \tag{4.5}$$

The matrices Φ_i and Υ_i are formed by the inputs and outputs of the integrated equation (4.3), i.e., ϕ and δ, respectively. Following the sampling matrices, we acquire the crucial prerequisite for the switching detection and active mode recognition, i.e., $\Upsilon_i = \Theta_i^\mathsf{T} \Phi_i$. Additionally, the difficulties of measuring the state derivatives are avoided.

The sampling matrix Φ_i is prescribed to obtain n_b linearly independent vectors, i.e., the number of independent rows of matrix is fewer than its number of independent columns. This condition can be achieved easily by assigning a large sampling size N such that $N \gg n_b$. It is obvious that the matrix $\Phi_i \Phi_i^\mathsf{T}$ is invertible while Φ_i is full row rank and the inverse of matrix $\Phi_i \Phi_i^\mathsf{T}$ is denoted by G_i.

Subsequently, we append the current error state $\delta(t)$ and the current integrated basis function $\phi(t)$ into the corresponding sampling matrices and obtain

$$\Upsilon(t) = \begin{bmatrix} \Upsilon_i & \delta(t) \end{bmatrix}, \ \Phi(t) = \begin{bmatrix} \Phi_i & \phi(t) \end{bmatrix}. \tag{4.6}$$

Then, the projection matrix is formulated as $\Pi(t) = I - \Phi^\mathsf{T}(t) \left(\Phi(t) \Phi^\mathsf{T}(t) \right)^{-1} \Phi(t)$ and the condition $\Upsilon(t)\Pi(t) = 0$ is fulfilled while the current $\delta(t)$ and $\phi(t)$ are also generated by the i-th subsystem. Thus, we define the last column vector of the matrix $\Upsilon(t)\Pi(t)$ as follows

$$\epsilon(t) = \delta(t) - \Upsilon_i \Phi_i^\mathsf{T} M(t) \phi(t) - \delta(t) \phi^\mathsf{T}(t) M(t) \phi(t), \tag{4.7}$$

where

$$M(t) = G_i - \frac{G_i\phi(t)\phi^{\mathsf{T}}(t)G_i}{1+\phi^{\mathsf{T}}(t)G_i\phi(t)}. \tag{4.8}$$

The residual $\epsilon(t) \in \mathbb{R}^{n_x}$ is the current residual that reveals whether the switching occurs at time t or not. Following the definition of $\epsilon(t)$, the following theorem is proposed for the switching detection.

Theorem 4. Suppose that all the inputs are nonzero. For the current time t, if the active modes are fixed during $[t-\Delta t,\ t]$, i.e., $\sigma(\tau) = i, \forall\tau \in [t-\Delta t,\ t]$ where $i = 1, 2, \ldots, s$, then the residuals fulfills

$$\epsilon(\tau) = \mathbf{0}, \forall\tau \in [t-\Delta t,\ t]. \tag{4.9}$$

Otherwise, $\exists\tau \in [t-\Delta t,\ t]$ such that $\epsilon(\tau) \neq \mathbf{0}$

Proof. By integrating the equation (4.3), the sampling matrices Θ_i and Φ_i satisfy

$$\Upsilon_i = \Theta_i^{\mathsf{T}}\Phi_i. \tag{4.10}$$

Meanwhile, if the current integrated state vector and regression vector are also generated by the i-th subsystem, then

$$\Upsilon(t) = \Theta_i^{\mathsf{T}}\Phi(t), \tag{4.11}$$

where

$$\Upsilon(t) = \begin{bmatrix}\Upsilon_i & \delta(t)\end{bmatrix},\ \Phi(t) = \begin{bmatrix}\Phi_i & \phi(t)\end{bmatrix}. \tag{4.12}$$

By the formulation of projection matrix and the equation (4.11), the following condition holds:

$$\Upsilon(t)\Pi(t) = \Theta_i^{\mathsf{T}}\Phi(t)\Pi(t) = \mathbf{0}, \tag{4.13}$$

where

$$\Pi(t) = I - \Phi^{\mathsf{T}}(t)\left(\Phi(t)\Phi^{\mathsf{T}}(t)\right)^{-1}\Phi(t). \tag{4.14}$$

Denote $M(t) = \left(\Phi(t)\Phi^{\mathsf{T}}(t)\right)^{-1}$. According to lock matrix multiplication rules and Sherman–Morrison–Woodbury formula [114], we have

$$M(t) = \left(\Phi(t)\Phi^{\mathsf{T}}(t)\right)^{-1} = G_i - \frac{G_i\phi(t)\phi^{\mathsf{T}}(t)G_i}{1+\phi^{\mathsf{T}}(t)G_i\phi(t)}. \tag{4.15}$$

Then, the equation (4.13) can be rewritten as

$$\begin{aligned}
&\Upsilon(t)\Pi(t)\\
&= \Upsilon(t)\left(I - \Phi^{\mathsf{T}}(t)M(t)\Phi(t)\right)\\
&= \Upsilon_i(t)\left(I - \begin{bmatrix}\Phi_i & \phi(t)\end{bmatrix}^{\mathsf{T}} M(t)\begin{bmatrix}\Phi_i & \phi(t)\end{bmatrix}\right)\\
&= \Upsilon(t)\left(I - \begin{bmatrix}\Phi_i^{\mathsf{T}}\\ \phi^{\mathsf{T}}(t)\end{bmatrix} M(t)\begin{bmatrix}\Phi_i & \phi(t)\end{bmatrix}\right)\\
&= \Upsilon(t)\left(I - \begin{bmatrix}\Phi_i^{\mathsf{T}}M(t)\Phi_i & \Phi_i^{\mathsf{T}}M(t)\phi(t)\\ \phi(t)^{\mathsf{T}}M(t)\Phi_i & \phi(t)^{\mathsf{T}}M(t)\phi(t)\end{bmatrix}\right)\\
&= \Upsilon(t) - \begin{bmatrix}\Upsilon_i & \delta(t)\end{bmatrix}\begin{bmatrix}\Phi_i^{\mathsf{T}}M(t)\Phi_i & \Phi_i^{\mathsf{T}}M(t)\phi(t)\\ \phi^{\mathsf{T}}(t)M(t)\Phi_i & \phi^{\mathsf{T}}(t)M(t)\phi(t)\end{bmatrix}\\
&= \Upsilon(t) - \Big[\Upsilon_i\Phi_i^{\mathsf{T}}M(t)\Phi_i + \delta(t)\phi^{\mathsf{T}}(t)M(t)\Phi_i \mid\\
&\Upsilon_i\Phi_i^{\mathsf{T}}M(t)\phi(t) + \delta(t)\phi^{\mathsf{T}}(t)M(t)\phi(t)\Big].
\end{aligned} \tag{4.16}$$

Later, we define $\epsilon(t)$ as the last column of the matrix $\Upsilon(t)\Pi(t)$, i.e.,

$$\epsilon(t) = \delta(t) - \Upsilon_i\Phi_i^{\mathsf{T}}M(t)\phi(t) + \delta(t)\phi^{\mathsf{T}}(t)M(t)\phi(t). \tag{4.17}$$

The residual $\epsilon(t)$ represents the current information of the projection of $\Upsilon(t)$ on $\Pi(t)$. If $\sigma(\tau) = i, \forall\tau \in [t-\Delta t, t]$, $\epsilon(t)$ is a zero vector according to the equation (4.13) under noiseless condition.

On the contrary, if the switching occurs at $\hat{t} \in [t-\Delta t, t]$ and the mode switches from i to j, the equation (4.13) is now transformed into

$$\begin{aligned}
\Upsilon(t)\Pi(t) &= \Theta_i^{\mathsf{T}}\left[\Phi_i \quad \int_{t-\Delta t}^{\hat{t}}\psi(\tau)d\tau + \int_{\hat{t}}^{t}\psi(\tau)d\tau\right]\Pi(t)\\
&\quad + (\Theta_j^{\mathsf{T}} - \Theta_i^{\mathsf{T}})\left[0 \quad \dots \quad 0 \quad \int_{\hat{t}}^{t}\psi(\tau)d\tau\right]\Pi(t)\\
&= \Theta_i^{\mathsf{T}}\Phi(t)\Pi(t)\\
&\quad + (\Theta_j^{\mathsf{T}} - \Theta_i^{\mathsf{T}})\left[0 \quad \dots \quad 0 \quad \int_{\hat{t}}^{t}\psi(\tau)d\tau\right]\Pi(t)\\
&= (\Theta_j^{\mathsf{T}} - \Theta_i^{\mathsf{T}})\left[0 \quad \dots \quad 0 \quad \int_{\hat{t}}^{t}\psi(\tau)d\tau\right]\Pi(t).
\end{aligned} \tag{4.18}$$

Then, the last column of the matrix $\Upsilon(t)\Pi(t)$ is

$$\begin{aligned}
\epsilon(t) &= h_{NN}(t)(\Theta_j^{\mathsf{T}} - \Theta_i^{\mathsf{T}})\int_{\hat{t}}^{t}\psi(\tau)d\tau\\
&= h_{NN}(t)\left(x(t) - x(\hat{t}) - x_i(t) + x(\hat{t})\right)\\
&= h_{NN}(t)\left(x(t) - x_i(t)\right),
\end{aligned} \tag{4.19}$$

where

$$h_{NN}(t) = 1 - \phi(t)^{\mathsf{T}} M(t)\phi(t), \tag{4.20}$$

and $x(t)$ and $x_i(t)$ implies the current state and the state at t if no switching occurs, respectively. Consider the assumption that the modes of the switched nonlinear system are discernible, the following condition holds:

$$x(t) - x_i(t) \neq \mathbf{0}. \tag{4.21}$$

Therefore, whether the residual $\epsilon(t)$ is a zero vector is determined by $h_{NN}(t)$, the last diagonal element of $\Pi(t)$. The residual fulfills $\epsilon(t) \neq \mathbf{0}$ as long as $h_{NN}(t) \neq 0$. Considering the formulation of projection matrix $\Pi(t)$, it can be regraded as the residual maker matrix in the least squares regression of $\Upsilon^{\mathsf{T}}(t)$ on $\Phi^{\mathsf{T}}(t)$ in statistics [115].

Subsequently, let us assume that $h_{NN}(t) = 0$ at first. It is obvious that the matrix $\Pi(t)$ is both symmetric $\left(\Pi(t) = \Pi^{\mathsf{T}}(t)\right)$ and idempotent $(\Pi(t) = \Pi^2(t))$ conforming to the equation (4.14). Thus, all entries in the last row and column of $\Pi(t)$ are zeros as long as $h_{NN}(t) = 0$. Since $\Pi(t)\Upsilon^{\mathsf{T}}(t)$ are the residuals in the least squares regression of $\Upsilon^{\mathsf{T}}(t)$ on $\Phi^{\mathsf{T}}(t)$, it is clear that the entries of last row of $\Upsilon^{\mathsf{T}}(t)$, i.e, $\delta^{\mathsf{T}}(t)$ are perfect fit on the last row of $\Phi^{\mathsf{T}}(t)$, i.e., $\phi^{\mathsf{T}}(t)$ in the linear regression and every residual is zero [115].

Under the noiseless condition, the matrix Φ_i and Υ_i are the perfect fit for each other in the linear regression for the reason that they are generated by the same subsystem. However, the parameters between the vectors $\phi^{\mathsf{T}}(t)$ and $\delta^{\mathsf{T}}(t)$ are different from the parameters between Φ_i and Υ_i. Therefore, a perfect fit for the entries of $\delta^{\mathsf{T}}(t)$ and $\phi^{\mathsf{T}}(t)$ in the regression is impossible under noiseless condition. Hence, the condition $h_{NN}(t) \neq 0$ is satisfied while switching occurs.

In summary, the following condition holds:

$$\epsilon(t) \neq \mathbf{0}, \tag{4.22}$$

while the switching occurs and the residual $\epsilon(t)$ can be employed as a criterion for switching detection. □

Theorem 4 endeavors to provide a criterion $\epsilon(t)$, which varies with the occurrence of switching, for the switching detection. According to the equation (4.19), the value of $\epsilon(t)$ is affected by $h_{NN}(t)$, the last diagonal element of $\Pi(t)$. Clearly, the value of $\epsilon(t)$ and $h_{NN}(t)$ are non-zero while switching occurs under the noiseless condition. However, in real applications, the measured states are corrupted by noises, which may result in $h_{NN}(t) = 0$ while switching occurs in an extreme case. It is unlikely but not impossible. Even though, we can still handle it by monitoring $h_{NN}(t)$ in advance via equation (4.20) and skip the mode recognition procedure by assigning the active mode to 0 while $h_{NN}(t) = 0$.

It is inapplicable to directly detect switching by judging whether $\epsilon(t)$ equals zero in real applications. It follows that the measured states and inputs are generally corrupted by noises. According to the proof, if no switching occurs, the residual $\epsilon(t)$ follows $E[\epsilon] \approx 0$ while the measurement noise is zero-mean Gaussian noise. Note that $E[\cdot]$ represents the mathematical expectation.

Therefore, the switching detection task turns into detecting a mean change in the residual, which can be solved via finite moving average algorithm (FMA) [116]. The outline of the FMA algorithm is calculating the mathematical expectation of residual in a window and

comparing it with a pre-defined threshold. The FMA algorithm depends on two coefficients: a window size η for mathematical expectation computation and a threshold vector $\zeta \in \mathbb{R}^{n_x}$ for comparison. The work of Zouari provides us the standard for choosing the window size η and the threshold ζ for switching detection [88]. The window size of FMA is chosen as $\eta = N \cdot t_s$. The threshold ζ is set based on data-driven residuals generation for each mode, which is given by

$$\zeta_{[j*]} = 0.9\sqrt{\text{var}[\epsilon_{[j*]}]}, j = 1, 2, \ldots, n_x, \tag{4.23}$$

when the active mode is fixed in the time window $[t - \eta, t]$. In Equation (4.23), $\text{var}[\epsilon_{[j*]}]$ denotes the variance of the j-th entries of ϵ in the window. Note that $\zeta_{[j*]}$ and $\epsilon_{[j*]}$ in the equation denote the j-th entry of the vectors ζ and ϵ, respectively. There are still some other standards for choosing the threshold and window size η and other methods for the mean change detection. However, we do not concentrate on the mean change detection in our work. We only select a feasible and convenient method to achieve it. The FMA method [116, 117] is adopted here only for the sake of ease in implementation and its effectiveness in detecting small shifts in the mean.

Subsequently, Algorithm 4 is proposed to detect switching and recognize the active mode based on Theorem 4.

In Algorithm 4, a fixed sampling interval t_s is used to generate the sampling matrices and $t_{i_1} \sim t_{i_N}$ of the equation (4.4) are assigned specific values. For instance, the sampling matrices of the first subsystem fulfill $t_{1_1} = t_0 - (N-1)t_s, t_{1_2} = t_0 - (N-2)t_s, \ldots, t_{1_N} = t_0$. Note that the choices of t_s and N only need to fulfill that there are n_b linearly independent vectors in Φ_i, which can be realized without effort. After obtaining the sampling matrices, the expectation of the residual $E[\epsilon]$ is computed for switching detection. It is worth noticing that $E[\epsilon]$ in Algorithm 4 is the expectation in window $[t - \eta, t]$.

Indeed, another challenge for designing the algorithm is how to determine whether the active mode after switching is a new one or a repetitive one. To achieve this, the sampling matrices are recorded from the subsystems whose modes have already been recognized, i.e., Υ_i and Φ_i for $i = 1, 2, \ldots \hat{s}$ in the algorithm (see line 18 in Algorithm 4). Note that $\hat{s}$ is the number of subsystems whose modes have been recognized. Subsequently, for each i, we calculate the residual ϵ_i corresponding to Υ_i and Φ_i with the vectors $\phi(t)$ and $\delta(t)$ which are generated by the current subsystem (see line 12 in Algorithm 4). If the expectation of the residual in the window fulfills $E[\epsilon_i] < \zeta$, the current subsystem is a repetitive one and the active mode is i. Otherwise, the current subsystem is a new one and the corresponding sampling matrices will be stored as well. Eventually, the switching signal mode $\hat{\sigma}(t)$ is estimated online by implementing Algorithm 4.

In Algorithm 4, there is a delay $\tau_\Delta = Nt_s + \Delta t + \eta$ between the occurrence of switching and the correct estimation of the active mode. For instance, if the i-th subsystem is activated at time instant t_j, the active mode is estimated as i at the time instant $t_j + \tau_\Delta$ via Algorithm 4. Therefore, we set the estimated active mode $\hat{\sigma}$ as zero during this delay period and the corresponding regression vectors are not applied in the estimation of subsystem parameters. In addition, the delay is exactly the time required to recognize the active mode correctly in Assumption 3. It follows that the minimum dwell time needs to satisfy $\tau_{\text{dwell}} > \tau_\Delta$. Consider the effect of delay τ_Δ, the stage of estimating the subsystem parameters will be skipped during the interval $(t_j, t_j + \tau_\Delta)$ since the active mode is zero, i.e., $\hat{\sigma}(t) = 0, \forall t \in (t_j, t_j + \tau_\Delta)$. Therefore, the identification procedure is less time-consuming if we choose a small t_s, η, and Δt. Fortunately, the identification results remain the same regardless of the choice of t_s, η,

and Δt due to the recursive estimation of the subsystem parameters.

4.2.2. Estimation of Subsystem Parameters

In this section, we focus on the online identification of each subsystem and propose the extension of integral concurrent learning adaptive identifier. The original method was firstly documented for adaptive control of linear systems [75]. In the section, the original method is extended to accomplish the identification of switched nonlinear systems. In the proposed identifier, we design the state observer for each subsystem and impose the state-reset mechanism to achieve the identification. Before introducing our identifier, we firstly define the integrated history stacks.

Algorithm 4 Recognition of Active Mode

Input: integration window size Δt, sampling interval t_s, sampling size N, window size η.
Output: The estimated number of subsystems $\hat{s}$, estimated active mode $\hat{\sigma}(t)$, and sequence of switching instants T_{sw}.

Let $\hat{s} = 1$, $\hat{\sigma} = 1$,$j = 0$, $\tau_\Delta = Nt_s + \Delta t + \eta$, Flag=True, $T_{sw} = \emptyset$.
$\Upsilon_1 = [\delta(t_0 - (N-1)t_s), \ldots, \delta(t_0 - t_s), \delta(t_0)]$
$\Phi_1 = [\phi(t_0 - (N-1)t_s), \ldots, \phi(t_0 - t_s), \phi(t_0)]$
Set the threshold ζ with the computed ϵ in the window $[t_0 - \eta, t_0]$
$\langle$Model structure selection via the refinement strategy (Section 4.2.3)$\rangle$
while $t \geq t_0$ **do**
  if $E[\epsilon] \geq \zeta$ and Flag is True **then**
    $j \leftarrow j + 1$, $\hat{\sigma} = 0$;
    $t_j \leftarrow t$, $T_{sw} \leftarrow T_{sw} \cup t$, Flag$\leftarrow$False;
  end if
  if $t \in (t_j, t_j + \tau_\Delta)$ **then**
    $\hat{\sigma} \leftarrow 0$;
  else if $t = t_j + \tau_\Delta$ **then**
    Calculate the residual $\epsilon_i(t)$ with respect to Υ_i and Φ_i for $i = 1, \ldots, \hat{s}$.
    if $\exists E[\epsilon_i] < \zeta$ for $i = 1, 2, \ldots, \hat{s}$ **then**
      $\hat{\sigma} \leftarrow i$;
    else
      $\hat{\sigma} \leftarrow \hat{s} + 1$;
      $\Upsilon_{\hat{\sigma}} = [\delta(t - (N-1)t_s), \ldots, \delta(t - t_s), \delta(t)]$
      $\Phi_{\hat{\sigma}} = [\phi(t - (N-1)t_s), \ldots, \phi(t - t_s), \phi(t)]$
      $\langle$Model structure selection via the refinement strategy (Section 4.2.3)$\rangle$
      $\hat{s} \leftarrow \hat{s} + 1$;
    end if
    Flag$\leftarrow$True
  end if
  Identify the subsystems according to $\hat{\sigma}(t)$ (Section 4.2.2)
end while
Return: $\hat{s}$, $\hat{\sigma}(t)$, and T_{sw}

Definition 5. The q-element ($q \in \mathbb{N}^+$) integrated history stack of subsystem i is defined as

$$\mathbf{H}_i = [\phi(t_{i_1}), \phi(t_{i_2}), \ldots, \phi(t_{i_q})], \tag{4.24}$$

where t_{i_k} is the k-th recorded time instant while the i-th subsystem is active, and

$$\sigma(\tau) = i \text{ for } \tau \in [t_{i_k} - \Delta t, t_{i_k}], \forall k = 1, \ldots, q. \tag{4.25}$$

The integrated history stacks $\mathbf{H}_i$, $i = 1, 2, \ldots, s$ are composed by the recorded integrated regression vectors, whose integration window size Δt is generally user-defined. In the extended integral concurrent learning adaptive identifier, the elements of each integrated history stack are selected such that there exist n_b linearly independent vectors. This requirement guarantees that sufficient information for the identification is contained in the integrated history stacks. Also, it provides a relaxed PE condition to ensure the convergence of estimated parameters.

Remark 5. Consider the demands of the integrated history stacks in Definition 5, the sampling matrix Φ_i is perfectly suitable to be the integrated history stack of the i-th subsystem for the proposed identifiers. Therefore, we directly employ Φ_i as the integrated history stacks for the i-th identifier in our approach, i.e. $\mathbf{H}_i = \Phi_i$ and $q = N$.

As aforementioned, the entries of x are involved in the regression vector ψ through a technical transformation. Hence, the i-th nonlinear subsystem can be transformed into

$$\dot{x} = A_i x + \overline{\Theta}_i^{\mathsf{T}} \overline{\psi}, \tag{4.26}$$

where

$$\Theta_i = \begin{bmatrix} A_i & \overline{\Theta}_i^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}}, \tag{4.27}$$

$$\psi = \begin{bmatrix} x^{\mathsf{T}} & \overline{\psi}^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}}, \tag{4.28}$$

and the parameters $A_i \in \mathbb{R}^{n_x \times n_x}$ and $\overline{\Theta}_i \in \mathbb{R}^{(n_b - n_x) \times n_x}$. The vector $\overline{\psi} \in \mathbb{R}^{(n_b - n_x)}$ is obtained by extracting the state vector $x(t)$ from regression vector $\psi(t)$.

According to the equation (4.26), we design a state observer to predict the states of the i-th subsystem

$$\dot{\hat{x}}_i = A_m \hat{x}_i + (\hat{A}_i - A_m) x + \hat{\overline{\Theta}}_i^{\mathsf{T}} \overline{\psi}, \quad \text{if } \hat{\sigma}(t) = i, \tag{4.29}$$

where $\hat{A}_i \in \mathbb{R}^{n_x \times n_x}$ and $\hat{\overline{\Theta}}_i \in \mathbb{R}^{(n_b - n_x) \times n_x}$ are the estimated parameters. $A_m \in \mathbb{R}^{n_x \times n_x}$ is a stable (Hurwitz) matrix which guarantees $\hat{x}_i$ does not diverge. It follows that there always exists a symmetric, positive definite $P \in \mathbb{R}^{n_x \times n_x}$ satisfying

$$A_m^{\mathsf{T}} P + P A_m = -Q_e, \tag{4.30}$$

where $Q_e \in \mathbb{R}^{n_x \times n_x}$ is a positive definite matrix [118]. Generally, the initial state in the observer is set as a null vector.

By the designed state observer (4.29), the i-th prediction error is defined as $e_i = \hat{x}_i - x$. The update law for the i-the estimated parameter is built on e_i. Meanwhile, the prediction error e_i is unrelated to the estimation process while the i-th subsystem is inactive. Thus,

we impose the discrete state reset mechanism to deal with the inactive situation [44]. The predicted state $\hat{x}_i$ is reset and set equal to the true state x while the corresponding subsystem is inactive, i.e.,

Figure 3.3 illustrates the state reset and shows how the state prediction for the i-th subsystem pauses while $\hat{\sigma}(t) \neq i$. The state reset ensures that the prediction error $e_i = 0$ while the i-th subsystem is inactive. In other words, the information from other subsystems is unrelated to the estimation of the i-th subsystem parameters.

In order to obtain a unified update law, the estimated and true parameters are reconstructed into vectors. We denote

$$\theta_i = \begin{bmatrix} \mathrm{vec}\,(A_i)^\mathsf{T} & \mathrm{vec}\,(\overline{\Theta}_i)^\mathsf{T} \end{bmatrix}^\mathsf{T} \in \mathbb{R}^{n_b n_x}, \tag{4.31}$$

$$\hat{\theta}_i = \begin{bmatrix} \mathrm{vec}\,(\hat{A}_i)^\mathsf{T} & \mathrm{vec}\,(\hat{\overline{\Theta}}_i)^\mathsf{T} \end{bmatrix}^\mathsf{T} \in \mathbb{R}^{n_b n_x}. \tag{4.32}$$

Besides, we introduce the gain matrix for the update law

$$\Psi = \psi \otimes I_{n_x}, \quad \Gamma = \begin{bmatrix} \gamma_1 I_{n_x n_x} & 0 \\ 0 & \gamma_2 I_{(n_b - n_x) n_x} \end{bmatrix} \tag{4.33}$$

where $\gamma_1, \gamma_2 \in \mathbb{R}^+$ are the gain. Moreover, I_{n_x}, $I_{n_x n_x}$ and $I_{(n_b - n_x) n_x}$ are identity matrices of dimension n_x, $n_x n_x$ and $(n_b - n_x) n_x$, respectively. Consequently, the update law of the extended integrated concurrent learning identifier for the switched nonlinear system identification is designed as follows.

Theorem 5. Consider the switched nonlinear system (4.1) with the estimated active mode $\hat{\sigma}(t)$. Let the states of each subsystem be predicted with the corresponding observer (4.29). Then, the update law of the i-th subsystem

$$\dot{\hat{\theta}}_i = -\Gamma \Psi P e_i + \frac{k_{\mathrm{CL}}}{2} \Gamma \sum_{k=1}^{N} \mathcal{Y}(t_{i_k}) \Big(\delta(t_{i_k}) - \mathcal{Y}^\mathsf{T}(t_{i_k}) \hat{\theta}_i \Big), \tag{4.34}$$

where for any $k = 1, \ldots, N$,

$$\mathcal{Y}(t_{i_k}) = \phi(t_{i_k}) \otimes I_{n_x}, \tag{4.35}$$

ensures that e_i converges to zero exponentially and the estimated parameters $\hat{\theta}_i$ converge to the true parameters θ_i. In the equation (4.34), $k_{\mathrm{CL}} \in \mathbb{R}^+$ is the gain and I_{n_x} is the identity matrix of dimension n_x.

Proof. The proof for theorem 5 is similar to the proof of theorem 4. Therefore, we only provide a brief version in this proof and spotlight the differences while proving Theorem 5. In the proof, we also neglect the index i (i.e. $\mathcal{Y}(t) \triangleq \mathcal{Y}_i(t)$ or $\theta \triangleq \theta_i$)

By recalling the definitions of θ in the equation (4.31), the state equation of one subsystem of the switched nonlinear system is written as

$$\dot{x} = (\psi \otimes I_{n_x})^\mathsf{T} \theta. \tag{4.36}$$

Integrating the equation (4.36) yields

$$\begin{aligned}\int_{\max(0,t-\Delta t)}^{t} \dot{x}(\tau)d\tau &= \int_{\max(0,t-\Delta t)}^{t} (\psi(\tau)\otimes I_{n_x})^{\mathsf{T}}\theta d\tau \\ &= \left(\int_{\max(0,t-\Delta t)}^{t} \psi(\tau)d\tau \otimes I_{n_x}\right)^{\mathsf{T}} \theta.\end{aligned} \tag{4.37}$$

Then, substituting (4.5) into (4.37), we obtain

$$\delta(t) = (\phi(t)\otimes I_{n_x})^{\mathsf{T}}\theta = \mathcal{Y}^{\mathsf{T}}(t)\theta. \tag{4.38}$$

With the equation (4.38), the update law of the subsystem takes the form

$$\begin{aligned}\dot{\hat{\theta}} &= -\Gamma\Psi Pe + \frac{k_{\mathrm{CL}}}{2}\Gamma\sum_{k=1}^{N}\mathcal{Y}(t_k)\Big(\delta(t_k) - \mathcal{Y}^{\mathsf{T}}(t_k)\hat{\theta}\Big) \\ &= -\Gamma\Psi Pe - \frac{k_{\mathrm{CL}}}{2}\Gamma\sum_{k=1}^{N}\mathcal{Y}(t_k)\mathcal{Y}^{\mathsf{T}}(t_k)\tilde{\theta},\end{aligned} \tag{4.39}$$

where

$$\tilde{\theta} = \hat{\theta} - \theta. \tag{4.40}$$

Additionally, the derivative of the prediction error for the active subsystem along the equations (4.26) and (3.31) is given by

$$\begin{aligned}\dot{e} = \dot{\hat{x}} - \dot{x} &= A_m e + (\hat{A} - A)x + (\hat{\overline{\Theta}} - \overline{\Theta})^{\mathsf{T}}\overline{\psi} \\ &= A_m e + \Psi^{\mathsf{T}}\tilde{\theta},\end{aligned} \tag{4.41}$$

where

$$\Psi = \psi \otimes I_{n_x}. \tag{4.42}$$

Consider the same quadratic candidate Lyapunov function used in the proof of Theorem2

$$V = \begin{bmatrix} e^{\mathsf{T}} & \tilde{\theta}^{\mathsf{T}}\end{bmatrix}\begin{bmatrix} P & 0 \\ 0 & \Gamma^{-1}\end{bmatrix}\begin{bmatrix} e \\ \tilde{\theta}\end{bmatrix}. \tag{4.43}$$

In light of the equations (4.39) and (4.41), the time derivative of V yields

$$\begin{aligned}\dot{V} &= \dot{e}^{\mathsf{T}}Pe + e^{\mathsf{T}}P\dot{e} + \dot{\tilde{\theta}}^{\mathsf{T}}\Gamma^{-1}\tilde{\theta} + \tilde{\theta}^{\mathsf{T}}\Gamma^{-1}\dot{\tilde{\theta}} \\ &= -e^{\mathsf{T}}Q_e e - \tilde{\theta}^{\mathsf{T}}Q_\theta\tilde{\theta},\end{aligned} \tag{4.44}$$

where $Q_e = -(A_m^{\mathsf{T}}P + PA_m)$ and $Q_\theta = k_{\mathrm{CL}}\sum_{k=1}^{N}\mathcal{Y}(t_k)\,\mathcal{Y}^{\mathsf{T}}(t_k)$. It is obvious the formulation of $\dot{V}$ is similar to the equation (4.44). Therefore, the successive proof is also similar to the proof of Theorem 2, which will not be presented in detail here. In summary, the state error e_i and the estimation error $\tilde{\theta}_i$ of parameters for $i = 1, 2\ldots s$ converge to zero exponentially. □

The update law (4.34) is constructed by two terms. The first and second terms are based on the prediction error e_i and integrated history stack Φ_i, respectively. While the i-th subsystem is activated, both of them are effective to update the estimated parameters. On the other hand, the estimated parameters are also updated by the second term while the i-th subsystem is inactive and $e_i = 0$. Thus the current and recorded data are used simultaneously in the update law. The update law is beneficial to the switched system identification since the estimated parameters of each subsystem are updated regardless of the activation condition.

Remark 6. In the extended integral concurrent learning identifier, the convergence rate of the parameters is determined by the minimum eigenvalues of Q_θ (See Proof). It is thus intuitive to update the integrated history stack such that the minimum eigenvalue $\eta_{\min}$ of Q_θ is maximized, as this maximizes the rate of convergence. The strategies for updating the integrated history stack are mature and have been widely used [44, 119]. They can be summarized as replacing each column vectors of Φ_i and re-calculate the minimum eigenvalue of $\sum_{k=1}^{N} \phi(t_{i_k})\phi(t_{i_k})^\mathsf{T}$. Then, among the original and replaced integrated history stack, the one that corresponds to the maximum $\lambda_{\min}$ is the new integrated history stack. However, the update strategy for history stacks owns a heavy computational burden, especially when the regression vector is high dimensional. In fact, the parameter convergence is still maintained even if the integrated history stack is not updated. The update strategy is only proposed to improve the convergence rate of the identification. Therefore, the adoption of the update strategies in our approach is flexible considering the number of basis functions.

4.2.3. Refinement Strategy for Model Structure Selection

In the proposed identification approach, the regression vectors which are generated based on the basis functions participant in the whole procedure. Although the choice of basis functions for the switched nonlinear system is important for identification, it is not the main concern of our dissertation. In fact, the limitation of the proposed approach is that the nonlinear terms of the system must be included in the selected basis functions. Therefore, we select a vast number of basis functions for the identification in the proposed approach. The vast number of basis functions guarantee the accuracy, whereas leads to high dimension regression vectors. Some of the basis functions may be redundant for the system and increase the computation burden for online identification. The estimated parameters with respect to the redundant basis functions will also be updated and converge to zero in our approach.

Although the proposed approach is online, the computational burden is still heavy while the regression vectors are high dimensional. Therefore, we propose a strategy based on the inputs and states to solve the MSS problem of the nonlinear subsystems. Specifically, the purpose of MSS is to determine which basis function is redundant and which one is indispensable for the system to be identified.

In order to reduce the computational burden in high dimensional situation, we present a theorem for MSS based on the aforementioned sampling matrices Υ_i and Φ_i.

Theorem 6. Let P_i denote the term $\Phi_i\Phi_i^\mathsf{T}$, if the j-th basis function is redundant, then the residual matrix R_m satisfies

$$R_m = \Upsilon_i \left(I - ((\Phi_i)_{[\bar{j}*]})^\mathsf{T} G_{i,\bar{j}} (\Phi_i)_{[\bar{j}*]} \right) = \mathbf{0}, \tag{4.45}$$

where

$$G_{i,\bar{j}} = \left(G_i + \frac{(G_i r_j)(e_j^\mathsf{T} G_i)}{1 - e_j^\mathsf{T} G_i r_j} \right)_{[\bar{j},\bar{j}]}, \tag{4.46}$$

and

$$r_j = (P_i)_{[*j]} - e_j; \tag{4.47}$$

Otherwise, $S \neq \mathbf{0}$.

Proof. In the proof, we assume that no switching occurs while selecting model structure and neglecting the index i in the proof for sake of convenience, such as $\Phi = \Phi_i, \Upsilon = \Upsilon_i, G_{\bar{j}} = G_{i,\bar{j}}, \kappa_j = \kappa_{i,j}$.

The inverse of the matrix $P_{[\bar{j},\bar{j}]} = \Phi_{[\bar{j}*]}(\Phi_{[\bar{j}*]})^\mathsf{T}$ plays an important role throughout the proof. Therefore, we need to prove that the sub-matrix $P_{[\bar{j},\bar{j}]}$ is invertible at first. As aforementioned, $P = \Phi\Phi^\mathsf{T}$ is invertible, whose inverse matrix is G. If the j-th diagonal elements of P and G are not null, then $P_{[\bar{j},\bar{j}]}$ is invertible, whose inverse can be computed via G [120]. Consider the formulations of P and G, both of them are positive-definite matrices and the diagonal elements are all positive. Thus, the sub-matrix $P_{[\bar{j},\bar{j}]}$ is invertible.

According to the inverse of block matrix and the property of elementary transformation, it is obvious

$$(P_{[\bar{j},\bar{j}]})^{-1} = ((P - r_j e_j^\mathsf{T})^{-1})_{[\bar{j},\bar{j}]}, \tag{4.48}$$

where

$$r_j = (P_i)_{[*j]} - e_j. \tag{4.49}$$

Then, the inverse of $P_{[\bar{j},\bar{j}]}$, which is denoted by $G_{\bar{j}}$, is computed based on the Sherman-Morrison formula [121]:

$$G_{\bar{j}} = ((P - r_j e_j^\mathsf{T})^{-1})_{[\bar{j},\bar{j}]} = \left(G_i + \frac{(G_i r_j)(e_j^\mathsf{T} G_i)}{1 - e_j^\mathsf{T} G_i r_j} \right)_{[\bar{j},\bar{j}]}. \tag{4.50}$$

We extract the entries which are generated by the j-th basis function from sampling matrix, i.e., the j-th row of the Φ_i. Then, the equation (4.10) is transformed into

$$\Upsilon = (\Theta_{[\bar{j}*]})^\mathsf{T} \Phi_{[\bar{j}*]} + \kappa_j \Phi_{[j*]}. \tag{4.51}$$

Therefore, if the j-th basis function is redundant, the parameter $\kappa_j = \mathbf{0}$ and it yields

$$\Upsilon = (\Theta_{[\bar{j}*]})^\mathsf{T} \Phi_{[\bar{j}*]}. \tag{4.52}$$

Thus, we denote $R_m = \Upsilon \left(I - (\Phi_{[\bar{j}*]})^\mathsf{T} G_{\bar{j}} \Phi_{[\bar{j}*]} \right)$, which is given by

$$\begin{aligned} R_m &= \Upsilon \left(I - (\Phi_{[\bar{j}*]})^\mathsf{T} G_{\bar{j}} \Phi_{[\bar{j}*]} \right) \\ &= \Upsilon - \Upsilon (\Phi_{[\bar{j}*]})^\mathsf{T} (\Phi_{[\bar{j}*]} (\Phi_{[\bar{j}*]})^\mathsf{T})^{-1} \Phi_{[\bar{j}*]} \\ &= \Upsilon - (\Theta_{[\bar{j}*]})^\mathsf{T} \Phi_{[\bar{j}*]} \\ &= \mathbf{0}. \end{aligned} \tag{4.53}$$

On the contrary, if the j-th basis function is indispensable, the matrix R_m takes the form

$$\begin{aligned} R_m &= \Upsilon \left(I - (\Phi_{[\bar{j}*]})^\mathsf{T} G_{\bar{j}} \Phi_{[\bar{j}*]} \right) \\ &= (\Theta_{[\bar{j}*]})^\mathsf{T} \Phi_{[\bar{j}*]} \left(I - (\Phi_{[\bar{j}*]})^\mathsf{T} G_{\bar{j}} \Phi_{[\bar{j}*]} \right) \\ &+ \kappa_j \Phi_{[j*]} \left(I - (\Phi_{[\bar{j}*]})^\mathsf{T} G_{\bar{j}} \Phi_{[\bar{j}*]} \right) \\ &= \kappa_j \Phi_{[j*]} - \kappa_j \Phi_{[j*]} (\Phi_{[\bar{j}*]})^\mathsf{T} G_{\bar{j}} \Phi_{[\bar{j}*]}. \end{aligned} \tag{4.54}$$

Since $\kappa_j \in \mathbb{R}^{n_x}$ and $\Phi_{[j*]} \in \mathbb{R}^{1 \times N}$, the only condition that fulfills $R_m = \mathbf{0}$ is that the vector $(\Phi_{[j*]})^\mathsf{T}$ is in the space spanned by the eigenvectors of $(\Phi_{[\bar{j}*]})^\mathsf{T} G_{\bar{j}} \Phi_{[\bar{j}*]}$, whose eigenvalues are 1.

We define the basis vectors of the eigenspace of $(\Phi_{[\bar{j}*]})^\mathsf{T} G_{\bar{j}} \Phi_{[\bar{j}*]}$ corresponding to eigenvalues 1 as $v_1, v_2 \ldots, v_p$. Obviously, we have

$$(\Phi_{[\bar{j}*]})^\mathsf{T} G_{\bar{j}} \Phi_{[\bar{j}*]} V = V, \tag{4.55}$$

where

$$V = \left[v_1, v_2 \ldots v_p \right]. \tag{4.56}$$

Since there exists a matrix $Q = G_{\bar{j}} \Phi_{[\bar{j}*]} V$ such that $(\Phi_{[\bar{j}*]})^\mathsf{T} Q = V$, the column space of $(\Phi_{[\bar{j}*]})^\mathsf{T}$ and V satisfy:

$$\mathrm{Cols}((\Phi_{[\bar{j}*]})^\mathsf{T}) \supseteq \mathrm{Cols}(V). \tag{4.57}$$

The vector $(\Phi_{[j*]})^\mathsf{T}$ is the j-th column of Φ^T. For the reason that the matrix Φ is full row rank, the vector $(\Phi_{[j*]})^\mathsf{T}$ is linear independent to the other columns of Φ^T. In other words, the vector $(\Phi_{[j*]})^\mathsf{T}$ is not in the column space of $(\Phi_{[\bar{j}*]})^\mathsf{T}$. According to the equation (4.56) and (4.57), the vector $(\Phi_{[j*]})^\mathsf{T}$ is also not in the eigenspace of $(\Phi_{[\bar{j}*]})^\mathsf{T} G_{\bar{j}} \Phi_{[\bar{j}*]}$, whose eigenvalues are 1. Therefore, the matrix $R_m \neq \mathbf{0}$ as long as the j-th basis function is indispensable.

In summary, the matrix R_m can be employed as a criterion for MSS. If the j-th basis function is indispensable, the residual matrix $R_m \neq \mathbf{0}$, otherwise, $R_m = \mathbf{0}$.

□

Theorem 6 can be summarized as calculating the corresponding residual with the projection matrix after deleting either basis function from the nonlinear subsystem. The proposed theorem provides the criterion R_m for the MSS strategy. In order to apply Theorem 5 to a practical nonlinear system, we adopt a similar method to FMA for the MSS. According to Equations (4.16) and (4.53), the residual matrix R_m and residual ϵ share similar formulations. Indeed, while the j-th basis function is redundant, the column vectors of R_m are exactly the residuals ϵ. Therefore, we compute the mathematical expectation of the column vector of R_m and compare it with the same threshold ζ. The basis function is redundant if the expectation exceeds the threshold; otherwise, it is indispensable. Note that we adopt the Sherman-Morrison formula to calculate $G_{i,\bar{j}}$, i.e., the inverse of matrix P after deleting the j-th basis function, for lower computation burden.

The refinement for MSS is introduced to raise the efficiency of the method, whose employment is determined by the application scenarios. If the number of basis functions is small, the refinement may not be essential for the identification.

4.3. Simulation Studies

In this section, a series of simulations are conducted to validate the effectiveness of the proposed online identification approach. The switched nonlinear system for simulations is selected as follows:

$$\begin{aligned} \dot{x}_1 &= f_{1,\sigma(t)}(x,u), \\ \dot{x}_2 &= f_{2,\sigma(t)}(x,u), \end{aligned} \tag{4.58}$$

where

$$\begin{aligned} f_{1,1} &= x_2 + 0.5u - 0.3x_1^3 + 0.4, & f_{2,1} &= -2x_1 - x_2 + 1.5u + 0.3x_1^3, \\ f_{1,2} &= 0.5u - 0.5x_1^3 - 0.3x_1x_2u, & f_{2,2} &= 1.5u - 0.3x_2^3 + 0.5x_1x_2u, \\ f_{1,3} &= x_2 - 0.3x_1x_2u + 0.5, & f_{2,3} &= -x_1 - x_2 + 1.5u + 0.4x_1x_2u. \end{aligned} \tag{4.59}$$

In the simulation, the time step is $0.01s$ and the system is excited by the following input signal:

$$u(t) = \sin(3t) + 0.5\sin(2t) - \sin(t). \tag{4.60}$$

The input signal is designed to be adequately rich both in frequency as well as in amplitude so that the relaxed PE condition can be easily fulfilled [122]. The switching signal $\sigma(t)$ switches every $100s$ and obeys the switching sequence $1 \to 2 \to 3 \to 1 \to \ldots$. Every state of the system, i.e., x_1, x_2, is corrupted by a zero mean Gaussian measurement noise of variance 0.005. Figure 4.2 shows the state space trajectory of the switched nonlinear system (4.58) excited by the input (4.60) from $t = 0s$ to $t = 600s$.

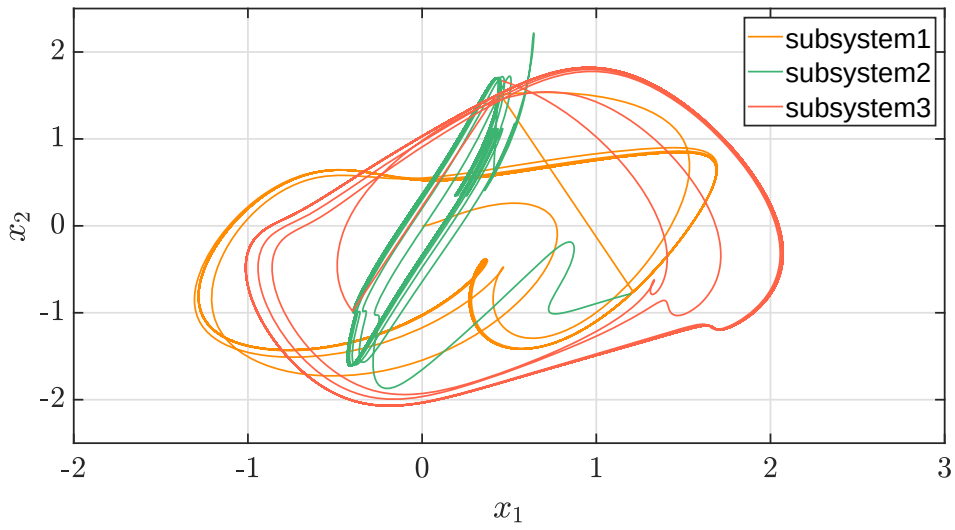


Figure 4.2.: State space trajectory generated by the switched nonlinear system (4.58), whose color represents the corresponding subsystem.

In the simulation, we set the coefficients for Algorithm 4 as: the integration window size $\Delta t = 0.1s$, the sampling interval $t_s = 0.5s$, the sampling size $N = 15$. The basis functions for the identification are determined as $\varphi_1(x,u) = x_1$, $\varphi_2(x,u) = x_2$, $\varphi_3(x,u) = u$, $\varphi_4(x,u) = 1$, $\varphi_5(x,u) = x_1^3$, $\varphi_6(x,u) = x_2^3$, $\varphi_7(x,u) = x_1x_2u$. By implementing Algorithm 4, the active modes from $t = 0s$ to $t = 600s$ are estimated and depicted in Figure 4.3(b). Meanwhile, the true active modes are plotted in Figure 4.3(a) for comparison. In Figure 4.3(b), there

is a short interval whose estimated active mode is zero after every switching. This interval indicates the delay between the switching occurrence and the correct estimation of the active mode. Moreover, the state and regression vectors during these intervals are ignored in the identification procedure due to the unknown active modes. However, the existence of zero is infeasible in the estimated switching signal. Thus, we replace the zero active modes with the corresponding correctly estimated active mode in the estimated switching signal. For instance, the active mode of the interval around $100s$ is alternated from 0 to 2 in the estimated switching signal.

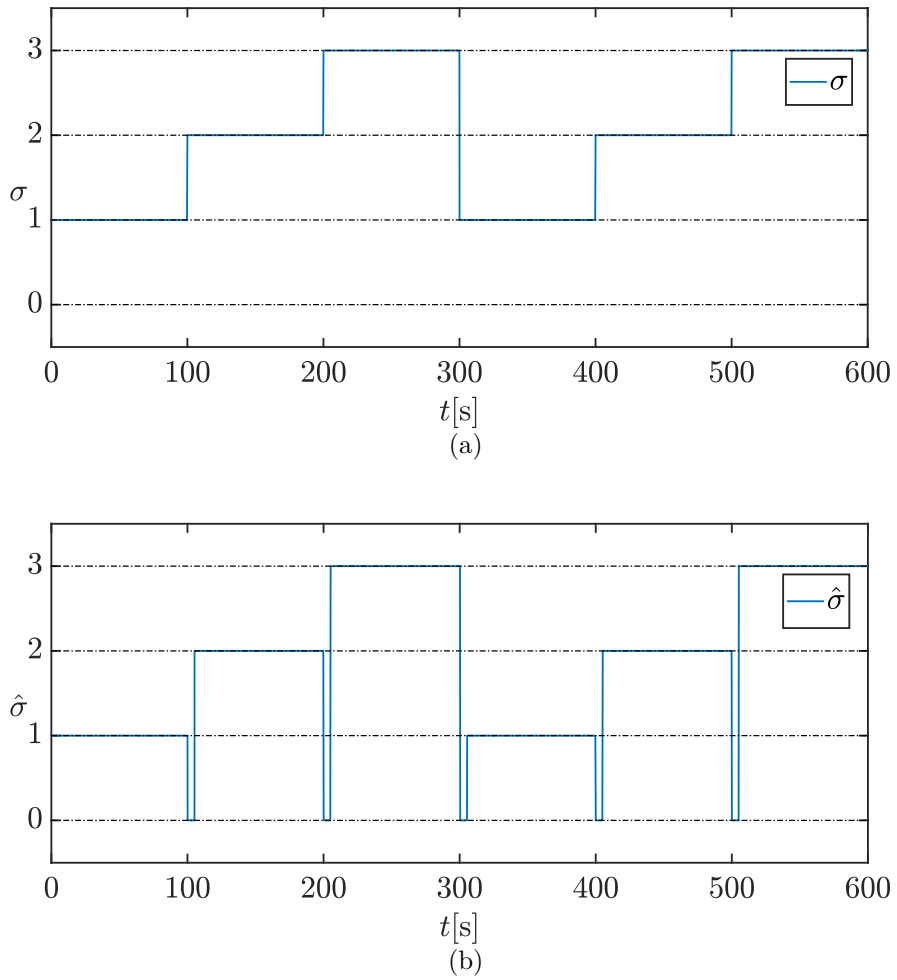


Figure 4.3.: The true active mode $\sigma(t)$ (4.3(a)) and estimated active mode $\hat{\sigma}(t)$ (4.3(b)) of the switched nonlinear system in the simulation.

For the extended integral concurrent learning identifier, all initial estimated parameters are set to zeros. The coefficients for the state observer and update law are $\gamma_1 = \gamma_2 = 4$, and

$$A_m = \begin{bmatrix} -10 & 0 \\ 0 & -10 \end{bmatrix}, \quad P = \begin{bmatrix} 0.05 & 0 \\ 0 & 0.05 \end{bmatrix}. \tag{4.61}$$

As presented in Remark 5, the sampling matrix Φ_i is employed as the i-th integrated history stack for the identification of subsystems. For the update law (4.34), the control gain is

$k_{\mathrm{CL}} = 0.8$ once the sampling matrix Φ_i is obtained. Before that, the control gain $k_{\mathrm{CL}} = 0$ and the update laws without concurrent learning are adopted.

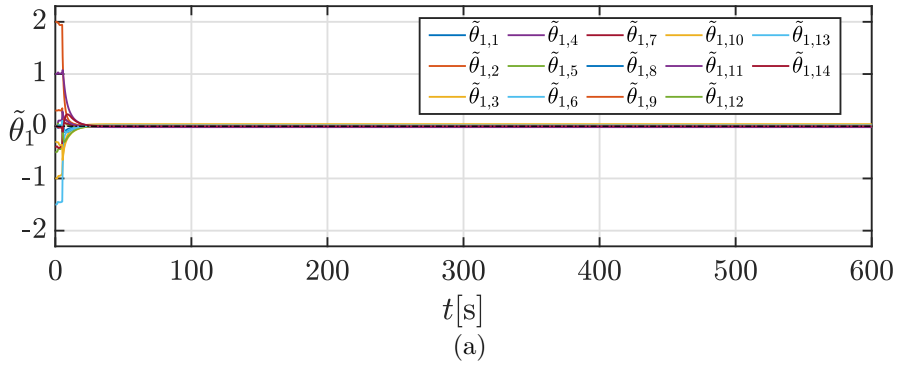
By implementing the update law, the estimated parameter error $\tilde{\theta}_i = \hat{\theta}_i - \theta_i \in \mathbb{R}^{14}$ of each subsystem w.r.t time is depicted in Figure 4.4, respectively. In these figures, the legend $\tilde{\theta}_{i,j}$ implies the j-th entry of the parameter $\tilde{\theta}_i$. Simulation results show a fast convergence of all estimated subsystem parameters to true parameters as expected. Meanwhile, it also illustrates that the adaptation is carried out no matter the corresponding subsystem is active or not. However, the estimated parameter errors do not converge to zeros perfectly due to the measurement Gaussian noise. Thus, the values of true and estimated parameters are listed in Table 4.1. The simulation results indicate the subsystem parameters are well-estimated.

Table 4.1.: True parameters (θ_1,θ_2,θ_3) and estimated parameters ($\hat{\theta}_1$,$\hat{\theta}_2$,$\hat{\theta}_3$) in the simulation

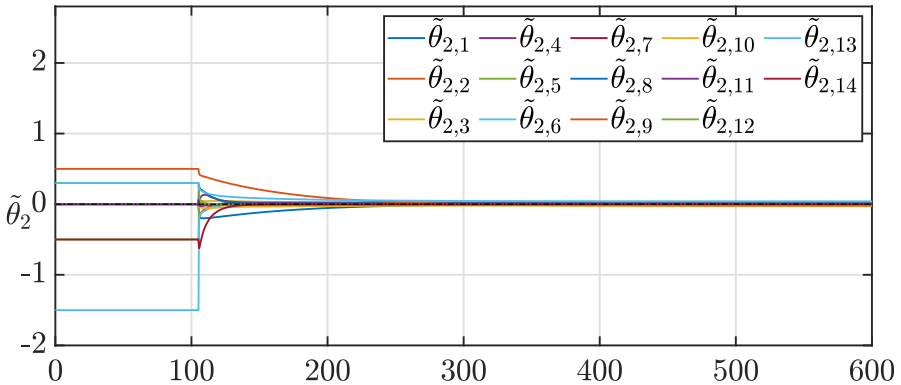
θ_1	$\hat{\theta}_1$	$\hat{\theta}_2$	$\hat{\theta}_2$	$\hat{\theta}_3$	$\hat{\theta}_3$
0	0.00	0	0.01	0	0
-2	-2.00	0	0.00	-1	-1.03
1	1.04	0	0.00	1	0.99
-1	-1.00	0	0.02	-1	-0.98
0.5	0.48	0.5	0.51	0	-0.00
1.5	1.49	1.5	1.50	1.5	1.52
0.4	0.39	0	0	0.5	0.51
0	0	0	0	0	0.00
-0.3	-0.31	-0.5	-0.53	0	0.00
0.3	0.31	0	-0.02	0	0.03
0	-0.02	0	0	0	0.01
0	0.01	-0.3	-0.31	0	-0.01
0	0.02	-0.3	-0.26	-0.3	-0.3
0	0	0.5	0.5	0.4	0.43

In the evaluation, the true and estimated switched nonlinear systems are excited by the same input signal, which is a random signal between $[-2, 2]$. Each entry of the true state-vector x and the estimated state-vector $\hat{x}$ from $0s$ to $400s$ are plotted in Figure 4.5. In addition, the performance of the identification approach is evaluated by the modified (BFR), i.e., Equation (3.58). The BFR values of the estimated $\hat{x}_1$ and $\hat{x}_2$ are 96.24% and 94.36%. The figure and values imply the satisfactory performance of the proposed approach.

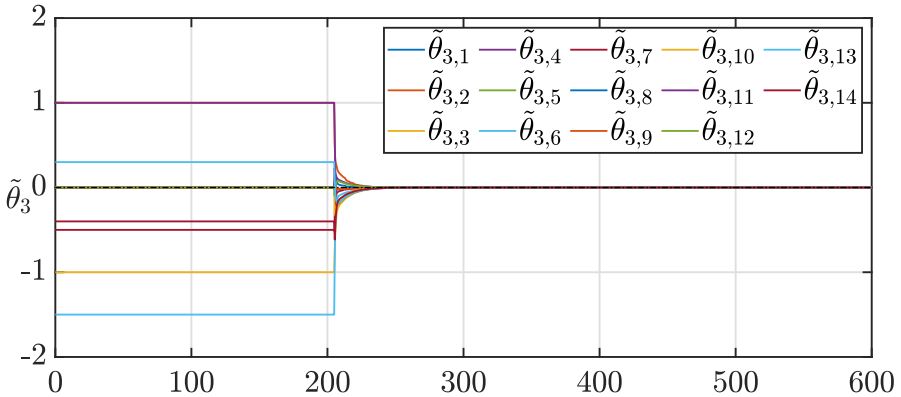
In the simulation, the refinement strategy for the MSS is also demonstrated. The existence of the basis functions being determined via the strategy is listed in Table 4.2. In Table 4.2, the model structure of each subsystem is revealed by check-mark and cross-mark. The check-mark indicates that the basis function is indispensable for the subsystem. Otherwise, the



(a)

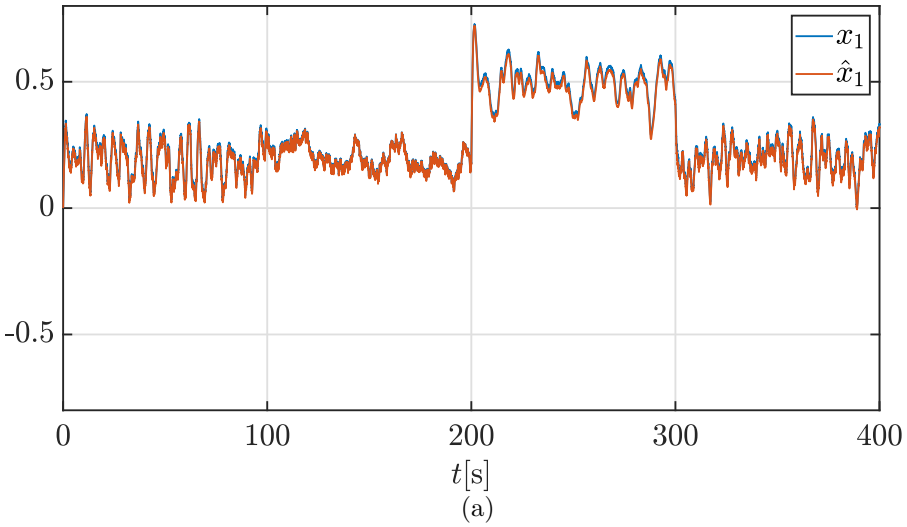


(b)

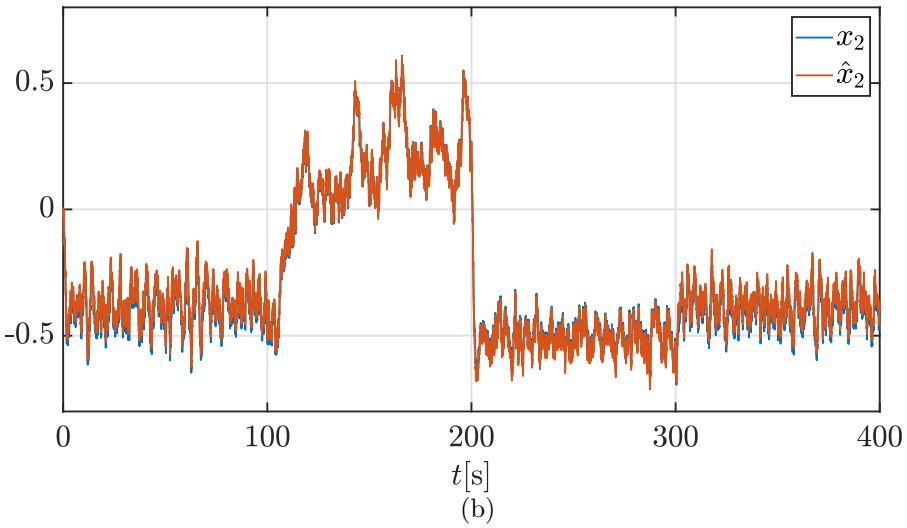


(c)

Figure 4.4.: The errors of estimated and true parameters $\tilde{\theta}_i$ in subsystem 1, 2 and 3, respectively. Every entry of the error vector is visualized by solid lines with different colors in every sub-figure.



(a)



(b)

Figure 4.5.: The entries of the true state-space vector x and the estimated state-space vector $\hat{x}$ while excited by the same random signal between $[-2, 2]$.

corresponding mark is a cross-mark. The effectiveness of the proposed refinement strategy is proved through the simulation.

In addition, the Sherman-Morrison formula is adopted in the strategy to calculate the inverse matrix. To justify the higher efficiency of using the formula, we record the time costs for MSS while using and not using the Sherman-Morrison formula, respectively. In the simulation, the average time for MSS is 3.02s (using Sherman-Morrison formula) and 3.65s (without using Sherman-Morrison formula) after running the algorithm 500 times.

To make it more intuitive, we enlarge the selected basis function and make the regression vector $\phi(t) \in \mathbb{R}^{15}$, the average time of MSS are now 3.05s (using Sherman-Morrison formula) and 4.45s (without using Sherman-Morrison formula). Thus, the effectiveness of Sherman-Morrison formula is confirmed through the recorded time.

Table 4.2.: The estimated and true model structure of the switched nonlinear system

Basis	subsystem 1		subsystem 2		subsystem 3	
function	estimated	true	estimated	true	estimated	true
x_1	✓	✓	✗	✗	✓	✓
x_2	✓	✓	✗	✗	✓	✓
u	✓	✓	✓	✓	✓	✓
1	✓	✓	✗	✗	✓	✓
x_1^3	✓	✓	✓	✓	✗	✗
x_2^3	✗	✗	✓	✓	✗	✗
x_1x_2u	✗	✗	✓	✓	✓	✓

4.4. Summary

In this chapter, we enhance the online framework to accomplish the online identification of switched nonlinear systems. The enhanced framework consists of two stages: estimation of the active mode and identification of subsystem parameters.

In the first stage, we propose an algorithm to detect switching and recognize the active mode recursively. The number of the subsystem is also determined with the algorithm. The algorithm is an extension of the subspace-projection method and the residual computed with the projection matrix is selected as the criterion for switching detection. After the switching is detected, the active mode estimation is then achieved via the recorded sampling matrices of the recognized subsystems. The residuals computed with the recorded sampling matrices reveal whether the active mode after switching is a new one or a repetitive one. The proof of the active mode estimation algorithm is presented in the chapter.

According to the estimated active mode, the extended integral concurrent learning identifier is derived to identify each subsystem in the second stage. The identifier is similar to the

one proposed in Chapter 3 and the same state-reset mechanism is employed to guarantee the convergence of the estimated parameters. The major difference is the adoption of regression vectors instead of state vectors in the identifier. The theoretical proof of the extended integral concurrent learning identifier is also described.

In the aforementioned algorithms, the regression vectors which are generated based on the basis functions are the foundations. Generally, we select a vast number of basis functions for the identification in the proposed approach to ensure accuracy. It is essential to select a proper model structure of the SNS since the high dimensional regression vectors increase the computational burden. Therefore, we propose a strategy for MSS as a refinement of the proposed approach. In the refinement strategy, the computed residual matrices are adopted as criteria to discriminate whether a basis function is redundant or dispensable for the subsystem. Naturally, we present the analytical proof of the refinement strategy in this chapter. It is worth noticing that the strategy is an option for the proposed methods since the extended I-CL identifiers ensure the convergence of the parameters regardless of MSS. The users could decide the adoption of the refinement strategy considering the computational burden.

Some simulations are also provided in the chapter after describing the proposed methods. The effectiveness of our approach and the corresponding refinement strategy is verified in extensive numerical experiments. In addition, the proposed framework is also applicable while identifying other continuous-time hybrid systems, such as PWNS, SLS, and PWA systems. It is a generalized framework for hybrid system identification regardless of the subsystem formulations. It follows that the hybrid system arises when the continuous dynamic behaviors interact with the discrete ones. The active mode recognition algorithm identifies its discrete dynamic behaviors while the extended I-CL identifier confirms its continuous ones. For the hybrid systems with state-dependent switching rules, the cost function from Chapter 3 is applicable to estimate their polyhedral partitions. In fact, these aforementioned algorithms can be combined freely according to the type of hybrid systems to be identified.

In summary, the number of subsystems, switching signal, and the parameters of each subsystem are precisely estimated online by implementing the proposed approach. The limitation of the proposed approach is that the nonlinear terms of the system must be included in the selected basis functions. Future works will concentrate on solving this limitation.

5. Conclusions and Outlook

In this dissertation, we provide novel methods for the identification of representative hybrid systems, the PWARX systems, continuous-time PWA systems, and continuous-time SNSs. In the beginning, we take an overview of the essential concepts, concerning factors, development paradigms, and approaches for hybrid system identification in previous works. Hinged on this, we firstly propose an efficient offline identification method for the PWARX systems, including a cluster-based algorithm for the initialization and a modified self-training algorithm for the parameter and polyhedral partition estimation. The method aims at achieving the identification efficiently by improving the traditional algorithm for polyhedral partitions estimation.

Subsequently, we develop the framework for the online identification of continuous-time hybrid systems. As an attempt to fill the void of the online identification methods for the hybrid systems, our work is targeted to propose an online, generalized framework to identify the hybrid systems. The framework contains the algorithms for switching detection and active mode recognition, subsystem parameter identification, and polyhedral partition estimation. The essence of the algorithm for active mode recognition is designing an appropriate input signal and analyzing the dynamics of the defined delay error. In addition, we also propose a strategy to handle the extreme situations which lead to the unrecognized active mode. Based on the recognized active mode, the generalized I-CL identifier is presented to online estimate the subsystem parameters. The polyhedral partitions are estimated through the designed cost function that evolved from the estimated switching sequence and subsystem parameters.

However, the proposed framework is limited to identify the hybrid system with linear/affine subsystems. Therefore, we enhance the framework to achieve the identification of the hybrid system with nonlinear subsystems, i.e., switched nonlinear systems. To the best of our knowledge, there are no works for the online identification of continuous-time nonlinear hybrid systems in state-space form. The enhancements can be summarized as designing a new active mode recognition algorithm and a refinement strategy for MSS for the framework. It follows that the delay error dynamics, being the foundation of the active mode recognition algorithm for PWA systems, are inapplicable if the subsystem formulation contains nonlinear terms. Therefore, we propose a new active mode recognition algorithm derived from the projection subspace method. The generalized I-CL identifier is still employed to online estimate the subsystem parameters. The refinement strategy for MSS is presented and can be integrated into the framework for online identification.

We validate the effectiveness of these novel identification methods through numerical simulations. The corresponding refinement strategies for handling rank-deficient extreme situations and selecting the model strictures are also verified through the simulation.

5.1. Conclusions

By proposing novel approaches for the online identification of hybrid systems and validating them in simulations and experiments, we provide reliable and applicable solutions to the three questions described in Section 1.1. The correspondence between our solutions and the questions is specifically interpreted as follows.

Solution to question 1: How to efficiently achieve the offline identification of the PWARX system?

In Chapter 2, we propose a novel offline identification method for PWARX systems to solve this question. The conventional offline methods for the PWARX system identification generally adopt the SVM algorithm and neglect the computational complexity in the polyhedral partition estimation process. It follows that the SVM algorithm is capable to estimate the hyperplanes precisely whereas bringing heavy computation burdens for a large data set. Therefore, we consider the drawback of the existing methods and present a novel semi-supervised learning-based approach to cope with the identification method efficiently. Our method contains two proposed algorithms: the cluster-based algorithm and the modified self-training algorithm. The cluster-based algorithm can be considered as an initialization procedure to obtain the initial conditions, including the number of subsystems, initial parameters, and initial data sets. The cluster-based algorithm is also practical for the initialization of other identification methods. These initial conditions are then applied in the self-training SVM algorithm to update the subsystem parameters and the hyperplanes that constructing the polyhedral partitions. The convergence proof of the estimated hyperplanes is provided in the chapter as well. In the novel approach, we argue that only a small amount of the labeled data points and a large amount of the unlabeled data points are adequate to estimate the precise hyperplanes. Only the points in the initial data sets and the support vectors are labeled during the update process.

We also consider the computational complexity of the proposed modified self-training SVM algorithm and compare it with the original SVM algorithm. It is validated that the proposed approach is effective with low computational complexity through numerical experiments. Moreover, the accuracy comparisons with some existing approaches are also provided in the section and reveal a satisfying accuracy of the proposed method.

Solution to question 2: How to accomplish the online identification of the continuous-time PWA system?

In Chapter 3, we solve this problem by developing a novel framework for the online identification of continuous-time PWA systems. Different from the conventional identification methods with limitations, the proposed online framework can identify the continuous-time PWA systems with an arbitrary number of subsystems by solely using the measured state-input vectors. The major challenge to achieve such responsiveness is to decouple the subsystem dynamics and switching process through the measurements. We tackle this issue by inferring the delay error and analyzing its dynamics. The delay errors are formulated into the sampling matrix and then provide a criterion for the switching detection and active mode recognition. Clearly, the subsystem dynamics and switching process are decoupled

after estimating which subsystem the state vector belongs to, i.e., active mode. In addition, we discuss the extreme situations that lead to the unrecognized active modes and provide a refinement strategy to handle these situations. Subsequently, the subsystem parameter is updated with the proposed generalized I-CL identifier at the insistence of the designed Luenberger-observer for each subsystem. The convergence proof of the identifier is presented. Later, we transform the estimation problem of polyhedral partitions into an optimization problem by proposing a cost function. The simulation results show that the proposed framework provides a satisfying accuracy for the online identification of continuous-time PWA systems. Moreover, a WMR system is identified to validate the effectiveness of the proposed framework in real scenario.

The proposed framework is generalized for the identification of linear/affine hybrid systems, such as PWL systems, switched linear systems, and switched affine systems. It follows that the hybrid system arises when the continuous dynamic behaviors interact with the discrete ones. The active mode recognition algorithm decouples the interaction between the continuous and discrete dynamic behaviors. The generalized I-CL identifier estimates the continuous dynamics. In addition, the estimation process of polyhedral partition is optional in the framework, considering the switching mechanism of target hybrid systems. Note that the proposed framework is limited to linear/affine hybrid systems. The reason is that the delay errors dynamics are inapplicable for the active mode recognition while there are nonlinear terms in the hybrid systems. This limitation is also the motivation for solving question 3.

In summary, we propose a novel framework for online identification of continuous-time PWA systems, including the number of subsystems, parameters, and polyhedral partition of each subsystem. The effectiveness of the approach is proved in the extensive numerical simulations.

Solution to question 3: How to carry out the online identification of the continuous-time switched nonlinear system?

In Chapter 4, we focus on the online identification of the continuous-time SNS. Considering the formulation of the SNS, it is intuitive that the framework in Chapter 3 needs to be enhanced before applying them for the SNS. For instance, the delay error for active mode recognition is built on the affine subsystem of the PWA system, which is nonlinear in the SNSs. In addition, the model structure selection is also unavoidable for SNS identification. To resolve these issues, we enhance the framework for the online identification in Chapter 3 by replacing the original active mode recognition algorithm and integrating the MSS strategy. Firstly, the new algorithm for active mode recognition is designed based on the subspace projection. The technical assumption is also replaced by a common discernible assumption in the field of identification. The generalized I-CL identifier is then still employed for the estimation of the subsystem parameters. Furthermore, we develop a refinement strategy for the MSS of the subsystems, which can be integrated into the framework to select a proper model structure before estimating the subsystem parameters. The effectiveness of the enhanced framework is justified via the numerical experiment and the accuracy of the estimates is satisfying. Also, the improvement of implementing the refinement strategy is validated in the simulation. It is clear that the proposed framework can now be generalized to identify the nonlinear hybrid systems, such as PWNSs, and SLSs. The reason is that the

active mode recognition algorithm is now capable to handle the nonlinear subsystems and the MSS strategy also copes with the heavy computational burden caused by high-dimensional regression vectors. Note that the estimation process of polyhedral partition from Chapter 3 can also be integrated into the framework, considering the switching mechanism of target hybrid systems.

In summary, this dissertation provides novel offline and online identification approaches for representative hybrid systems, i.e., PWARX systems, continuous-time PWA systems, and continuous-time SNSs. The semi-supervised learning-based approach concerns the polyhedral partition estimation process and is designed to improve the efficiency of the offline identification of PWARX systems (See Section 2). Considering the continuous-time hybrid systems in state-space form, i.e., PWA system and switched nonlinear system, we develop the generalized frameworks for the online identification of them. We decompose the online identification into three tasks: a) active mode recognition, b) subsystem identification, and c) switching signal estimation/polyhedral partition estimation. The algorithms for solving these tasks are presented and proved in Chapter 3 and 4. These algorithms can also be combined to achieve the identification of tremendous hybrid systems, such as PWL systems, switched linear/affine systems, and piecewise nonlinear systems. Moreover, the effectiveness of the proposed approach and frameworks are revealed through numerical experiments. Nevertheless, there are still several issues in the online identification that are not perfectly solved in this dissertation, which will be detailedly interpreted in Section 5.2, with an outlook of the possible improvements.

5.2. Improvements and Future Research Directions

While the proposed methods provide a novel perspective to cope with the hybrid system identification, it also leaves several open problems for potential improvements. Some of the major concerns are specifically discussed as follows.

Generalization of the coefficients in the offline identification method.

The major remaining issue for the proposed offline identification method for the PWARX system is how to choose appropriate coefficients to acquire a satisfying accuracy. Even though we have provided the qualitative relationship between the performance of the approach and these coefficients, the main challenge is that the qualitative relationship is not adequate to acquire generalized coefficients that provide satisfying performance for the identification. Improvements include acquiring the generalized coefficients for the satisfying accuracy or redesign the approach to reduce the coefficients that need to be tuned.

Dwell time of the hybrid system.

As mentioned in Section 4.2, there is a delay between the occurrence of switching and the estimation of active mode correctly. The reason is that we can confirm the active mode after switching only when the residuals/expectations are in a steady-state or in a small range, whose calculations are based on the data in a time interval. However, the existence of the delay requires that the dwell time is long enough such that the parameters of the subsystems

can be updated after recognizing the active mode. This feature may still lead to some difficulties while identifying the hybrid system with fast switching. Therefore, improving the algorithm to shorten the delay or totally eliminate it could be future research directions.

Robustness of the online identification methods.

The generalized I-CL identifier is the major component of the framework for online identification. However, the identifier is sensitive to the estimated active mode. In other words, if the active mode is wrongly estimated, the generalized I-CL identifier may not guarantee the convergence of the estimated subsystem parameters. To handle this problem, we assign zero to the active modes which can not be confirmed and skip their state-inputs vectors while using the generalized I-CL identifier to update parameters. However, this countermeasure may be inapplicable while the measurement noise is intense. It follows that switching detection and active mode recognition are achieved by comparing the calculated residuals/expectations with the pre-defined thresholds. The intense measurement noises may lead to large uncertainties and the active mode which is wrongly estimated. As a result, the subsystem parameters identified by the generalized I-CL identifier may not be converged. Therefore, the robustness of the proposed framework needs to be analyzed and improved in future works. To the best of our knowledge, the relationship between the online switching signal estimation and the measurement noise has been researched in some papers. However, their research is built on the acknowledgment of subsystem models, which is infeasible in system identification. Thus, the robustness of online switching detection and active mode estimation is still a major challenge in future research directions.

A. Support Vector Machine (SVM)

For a classification problem, the output of the training process is a discriminant, a vector function that determines the class of the sample to be tested. The prediction is conducted usually by explicitly applying the discriminant. A classification model is referred to as a parametric method if its discriminant depends on a parameterized probabilistic distribution. The training of parametric models renders an estimation problem for the distribution parameters, where statistic methods are used. Otherwise, the classification model is called a non-parametric method, where the discriminant is obtained by solving an optimization problem. In this section, we introduce background knowledge of SVM algorithm [123], which is a representative non-parametric classification method used in the polyhedral estimation of hybrid system.

Support Vector Machine (SVM) is a non-parametric classifier with an excellent capability of generalization. From a geometric point of view, SVM creates a pair of hyperplane for each two-class data set. The training process of the classifier tends to create the widest separation between the hyperplane pair. As a result, SVM is quite insensitive to the distribution of the data set since its structure only depends on a small cluster of training samples. Nonlinear kernel functions allow the mapping of samples into higher-dimensional spaces, such that linearly inseparable problems can be solved.

Consider a data set $\mathcal{D} = \{(\mathbf{x}_k, \ell_k)\}_{k=1}^N$ with cardinality $N \in \mathbb{N}^+$, where each data point is a pair consisting of a vector $\mathbf{x}_k$ and a label ℓ_k. The fundamental task of SVM is to train a classifier (w, b) to separate the samples with labels, where $w \in \mathbb{R}^n$ and $b \in \mathbb{R}$ are the weight and intercept of the classifier, respectively.

Equivalently, training an SVM classifier is to solve the following quadratic optimization problem.

$$\max_{0 \le \alpha_k \le C} \left(\min_{w,b} \ \frac{1}{2}||w||^2 + \sum_{k=1}^{N} \alpha_k (1 - \ell_k (w^{\mathsf{T}} \mathbf{x}_k + b)) \right), \tag{A.1}$$

where α_k, $k = 1, \ldots, N$ are the Lagrange multipliers and C is the penalty parameter. According to the Lagrange duality principle, the optimization is transformed into

$$\begin{aligned} \alpha^* = \arg\min_{\alpha} & \left(-\frac{1}{2} \sum_{k=1}^{N} \sum_{j=1}^{N} \alpha_k \alpha_j \ell_k \ell_j \mathbf{x}_k^{\mathsf{T}} \mathbf{x}_j + \sum_{k=1}^{N} \alpha_k \right), \\ \text{s.t.} \ & \sum_{k=1}^{N} \alpha_k \ell_k = 0, 0 \leqslant \alpha_k \leqslant C, \forall k = 1, 2, \cdots, N. \end{aligned} \tag{A.2}$$

By solving the optimization problem, the weight of the hyperplane is

$$w = \sum_{k=1}^{N} \alpha_k \ell_k \mathbf{x}_k. \tag{A.3}$$

The intercept can be easily calculated through the weight.

By Karush-Kuhn-Tucker (KKT) conditions [124], it requires that

$$\alpha_k[\ell_k(w^\mathsf{T}\mathbf{x}_k + b) - 1 + \xi_k] = 0 \text{ for all } k = 1, 2, \ldots, N, \tag{A.4}$$

where $\xi_k \geq 0$ are the slack variables which measure the degree of misclassification. Since there holds $\ell_k(w^\mathsf{T}\mathbf{x}_k + b) \geq 1 - \xi_k$ for all $k = 1, 2, \ldots, N$, it is straightforward that the data point with positive α_k implies $\ell_k(w^\mathsf{T}\mathbf{x}_k + b) \leq 1$. In this regard, support vectors are defined as data points which correspond to non-zero α_k in [55]. Equivalently, we define the support vectors (SVs) as follows.

Definition 6. Consider the SVM classifier (w, b) trained with data set $\mathcal{D} = \{(\mathbf{x}_k, \ell_k)\}_{k=1}^N$, the support vectors are the data points satisfying

$$\ell_k(w^\mathsf{T}\mathbf{x}_k + b) \leq 1. \tag{A.5}$$

By definition 6, the SVs are the data points lie on or in the margins of the SVM classifier which are defined as $w^\mathsf{T}\mathbf{x}_k + b = \pm 1$. It follows that only a small number of the training data points end up as SVs and the majority lie outside of the classifier margins [48]. For a data set $\mathcal{D}$ for SVM training, we denote the set of support vectors as $\mathcal{D}^{SV}$. The data points in $\mathcal{D} \setminus \mathcal{D}^{SV}$ are defined as non-support vectors (non-SVs). In addition, the trained classifier is fully determined by SVs in SVM [125].

Another concept in SVM algorithm is the geometric margin of trained classifiers [126] which is defined as follows.

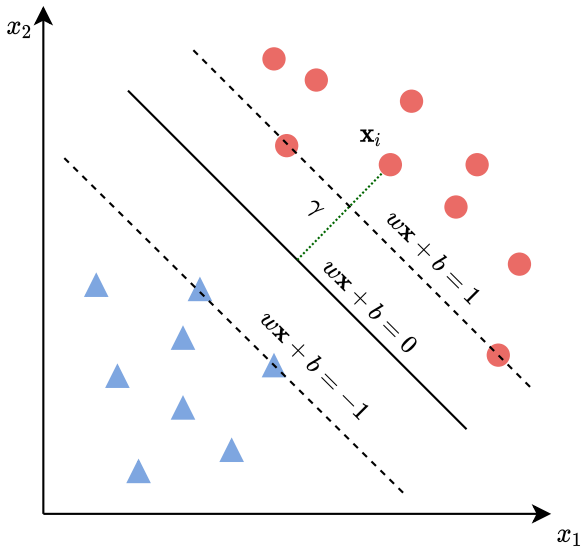


Figure A.1.: The geometric margin γ between a data point $\mathbf{x}_i$ and the decision boundary.

Definition 7. For the classifier (w, b) trained by data set $\mathcal{D} = \{(\mathbf{x}_k, \ell_k)\}_{k=1}^N$, the geometric margin between a certain data point $(\mathbf{x}_k, \ell_k)$ and the classifier is given by

$$\gamma(w, b, \mathbf{x}_k) = \frac{|w^\mathsf{T}\mathbf{x}_k + b|}{||w||}. \tag{A.6}$$

As depicted in the equation, the geometric margin is defined as the distance from a data point to the decision boundary. Figure A.1 illustrates the geometric margin between a data point $\mathbf{x}_i$ and the decision boundary. Note that the definition of the geometric margin in the foundation of the proposed modified self-training SVM algorithm.

Although the classifier introduced above is designed for binary classification, the concepts are the same for the multi-class classification. The reason is that the multi-class classification problems are generally considered as a collection of several binary classification problems [127].

Background on the Identification of Continuous-Time Hybrid Systems

B.

This appendix provides necessary backgrounds on the stability of switch systems, which are needed throughout the dissertation. Important stability notions for switched and hybrid systems are summarized in this Appendix. Note that both the continuous-time PWA system and switched nonlinear system can be categorized as switched systems.

B.1. Lyapunov's Stability Theory

The concept of Lyapunov stability is a fundamental tool in the analysis of dynamic systems. In this thesis, we assume that the basic Lyapunov stability concepts are known [128]. The stability definitions for switched systems presented in [85] are straightforward modifications of the standard stability concepts in [128].

Let $x = 0$ be an equilibrium point of the switched system. Then, the switched system is *stable* in the sense of Lyapunov if, for each ϵ, there exists a $\delta = \delta(\epsilon) > 0$ such that

$$\|x(0)\| \leq \delta \quad \Rightarrow \quad \|x(t)\| \leq \epsilon \quad \forall t \geq 0 \tag{B.1}$$

The switched system is *globally asymptotically stable* if it is stable and there exists $T(\epsilon, \delta) > 0$ for arbitrary positive numbers ϵ and δ such that

$$\|x(0)\| \leq \delta \quad \Rightarrow \quad \|x(t)\| \leq \epsilon \quad \forall t \geq T \tag{B.2}$$

Moreover, the switched system is called *globally exponentially stable* if there exist constants $c > 0$ and $\lambda > 0$ such that the state of switched system for all initial conditions satisfies the inequality

$$\|x(t)\| \leq c\|x(0)\|e^{-\lambda t} \quad \forall t \geq 0 \tag{B.3}$$

It is worth noticing that asymptotic or exponential stability is referred to as *uniform* if stability is not only asymptotic or exponential for a particular switching signal but over the whole set of all switching signals. Hence, the notion *global uniform asymptotic stability* (GUAS) implies independence between the global asymptotic stability and the choice of the switching signal σ.

As mentioned above, the stability of switched systems is heavily determined by the switching signal. This condition yields two levels of stability. First, The target of proving the *stability* under arbitrary switching is to find conditions such that the asymptotic stability of a switched system is guarantee for all possible switching signals σ. This approach is essential if the switching mechanism is unconstrained, unknown, or too complicated to be modeled.

Moreover, a necessary condition of the asymptotic stability of a switched system is that all individual subsystems must be asymptotically stable. Note that stability under arbitrary switching is highly desirable as it implies that the switching can be ignored while analysing the switched systems. However, it is obvious that only few classes of switched systems allow stability under arbitrary switching.

Considering the switched systems which are not asymptotically stable for arbitrary switching, it is necessary to figure out which condition switching signals fulfill to ensure that switched system is asymptotically stable. This approach is referred to as stability under constrained switching and can be further divided according to the set of subsystems. If at least some subsystems are asymptotically stable, it suffices to characterize as well as possible the class of switching signals that preserve asymptotic stability for the switched system. The existence of such switching sequences is guaranteed as one could just keep a stable subsystems active for all times. In case none of the subsystems is asymptotically stable, the task is to construct a stabilizing switching sequence. This task is considerably more challenging and is not in the scope of this work. The interested reader is referred to [85] for more details on the later case.

B.2. Common and Multiple Lyapunov Functions

The existence of a function that qualifies as a Lyapunov function for all subsystems is the foundation of the stability of a switched system under arbitrary switching. Such a Lyapunov function is called a common Lyapunov function for the family of subsystems. Particularly, a common Lyapunov function for switched systems is a positive definite ($V(x) > 0, \forall x \neq 0$) and continuously differential function $V : \mathbb{R}^n \to \mathbb{R}$, if there exists a positive definite, continuous function $W : \mathbb{R}^n \to \mathbb{R}$, such that

$$\dot{V}(x) = \frac{\partial V(x)}{\partial x} g_i(x) \leq -W(x) < 0, \quad \forall x, \quad \forall i \in \mathcal{I} \tag{B.4}$$

The relationship between the asymptotic stability of a switched system and the existence of a common Lyapunov function is described in the following theorem.

Theorem 7. "If all subsystems of a switched system share a radially unbounded common Lyapunov function, then the switched system is *globally uniformly asymptotically stable* (GUAS)." [[85], Th.2.1]

The idea behind this theorem is that the common Lyapunov function V will decrease in time in any subsystem i. The rate of decrease of V is characterized by W and is thus not affected by switching. Hence, the asymptotic stability is uniform with respect to σ, which can thus be arbitrary. For the special case in which $V(x)$ and $W(x)$ are quadratic in x, the theorem implies global uniform exponential stability. Even though the common Lyapunov function benefits a lot while analyzing the stability of switched systems, the difficulties of searching one for switched systems severely limits its adoption. Therefore, we turn into the multiple Lyapunov functions for assistance.

As pointed out before, it might not be possible to prove asymptotic stability of a switched system under arbitrary switching. In that case, one can proceed by constraining the switching signal and work with multiple Lyapunov functions.

Suppose that all systems are globally asymptotically stable and an individual Lyapunov function V_i is known for each subsystem $i \in \mathcal{I}$. The target is then to prove stability for a constrained switching signal σ based on the switched Lyapunov function V_σ. Before the corresponding analysis, we provide a more precise notation for σ. We denote $t_j, j = 1, 2, \ldots$ as the switching instants. Between two consecutive switching instants, the switching signal is constant, i.e. $\sigma(t) = \sigma(t_j), t_j \leq t < t_{j+1}$.

There exist cases that the values of the Lyapunov functions V_i coincide at all switching times t_j, i.e., $V_{\sigma(t_j-1)}(t_j) = V_{\sigma(t_j)}(t_j)$. It is clear that V_σ is a continuous Lyapunov function for the switched systems in this case. Then, the stability of the switched system directly follows from such a continuous Lyapunov function. Figure B.1(a) visualizes this scenario for a switched system with two subsystems.

A continuous Lyapunov function is a rare special case of multiple Lyapunov functions that mainly occurs for state-dependent switching signals and specially constructed local Lyapunov functions. Commonly, V_δ is discontinuous at switching times. It follows that the individual Lyapunov functions V_i is antagonistic. Generally, the value of the currently active ($\sigma(t) = i$) Lyapunov function V_i decreases, while the values of currently inactive Lyapunov functions V_j increase as long as $\sigma \neq j$. Figure B.1(b) gives an example for such discontinuous Lyapunov functions. Despite the possible increase of V_σ at every switch, multiple Lyapunov

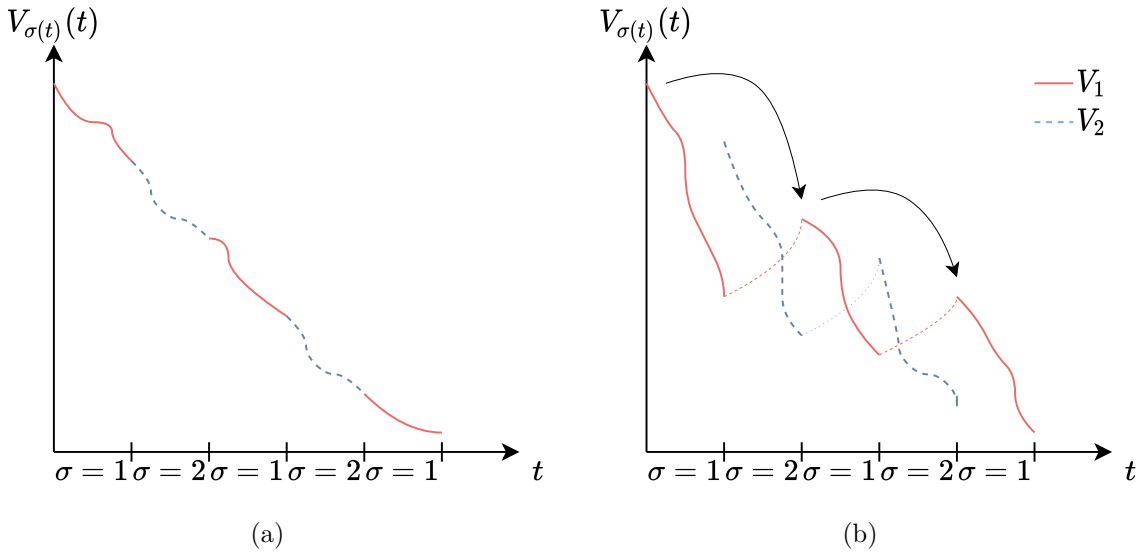


Figure B.1.: Multiple Lyapunov functions V_i form a switched Lyapunov function V_σ, (a) continuous Lyapunov function, (b) discontinuous Lyapunov function.

functions still is capable to prove the asymptotic stability. The idea is to analyze the values of each Lyapunov function V_i individually at the beginning of each interval in which the corresponding subsystem is active, i.e. in which $\sigma(t) = i$. As shown by the following theorem, the switched system is asymptotically stable if the values of the Lyapunov functions at each activation form a decreasing sequence.

Theorem 8 (Multiple Lyapunov Functions)**.** Let the subsystems be a finite family of globally asymptotically stable systems, and let V_i, $i \in \mathcal{I}$ be a family of corresponding radially unbounded Lyapunov functions. Suppose that there exists a family of positive definite

continuous functions $W_i, i \in \mathcal{I}$ with the property that for every pair of switching times $(t_j, t_n), j < n$ such that $\sigma(t_j) = \sigma(t_n) = i \in \mathcal{I}$ and $\sigma(t_k) \neq i$ for $t_j < t_k < t_n$, we have

$$V_i\left(x\left(t_l\right)\right) - V_i\left(x\left(t_j\right) \leq -W_i\left(x\left(t_j\right)\right)\right. \tag{B.5}$$

Then the switched system is globally asymptotically stable." [[85], Th.3.1]

According to the theorem of multiple Lyapunov functions, the proof of the I-CL identifiers for continuous-time PWA systems and SNSs are accomplished.

The Popov-Belevitch-Hautus Test for Controllability

C.

This appendix provides a brief description of PBH controllability test, which is the foundation of solving extreme conditions in Chapter 3.

Suppose we have a linear system described by the state equation

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t), \\ x(0) &= x_0. \end{aligned} \tag{C.1}$$

Controllability and observability are two critical properties while studying the linear systems. Generally speaking, controllability measures the ability of a particular actuator configuration to control all the states of the system; conversely, observability measures the ability of the particular sensor configuration to supply all the information necessary to estimate all the states of the system. In the appendix, we concerns the controllability of linear system. In particular, a system is said to be completely controllable, if it is possible to transfer the system state from any initial state $x(t_0)$ to any desired state $x(t)$ in specified finite time by a control vector $u(t)$.

We assume that the readers have the primary knowledge of the controllability and observability. Then, the theorem for judging whether a linear system is controllable is as follows.

Theorem 9 (Controllability)**.** The linear system (C.1) is controllable if and only if the rank of $[B\ AB\ \dots\ A^{n-1}B]$ equals to n, i.e., $\text{rank}[B\ AB\ \dots\ A^{n-1}B] = n$.

Note that n is the order of the system (C.1). Obviously, the theorem for judging controllability is hard to implement while the system order is high. Therefore, the PBH controllability tests give us alternative ways to test for controllability of a linear system. We state the test as a theorem.

Theorem 10 (PBH test)**.** (A, B) is controllable, if and only if there exists no left eigenvector of A orthogonal to the columns of B.

We will prove the sufficiency and necessity of the statement separately.

i) If there exists a left eigenvector of A orthogonal to the columns of B, then (A, B) is uncontrollable:

Proof. Suppose that $w \in \mathbb{C}^n$, $w \neq 0$ is a left eigenvector of A, that is orthogonal to the columns of B. Then

$$w^{\mathsf{T}}A = \lambda w^{\mathsf{T}}, \quad w^{\mathsf{T}}B = 0 \tag{C.2}$$

so

$$w^{\mathsf{T}}[B\ AB\ \dots\ A^{n-1}B] = 0 \tag{C.3}$$

and therefore

$$\text{rank}([B\ AB\ \dots\ A^{n-1}B]) < 0 \tag{C.4}$$

which means the controllability matrix has linearly dependent rows, and is not controllable. □

ii) If (A, B) is not controllable, there exists a left eigenvector of A orthogonal to the columns of B:

Proof. Suppose

$$\text{rank}([B\ AB\ \dots\ A^{n-1}B]) < 0 \tag{C.5}$$

we can change coordinates as follows

$$\tilde{A} = T^{-1}AT = \begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ 0 & \tilde{A}_{22} \end{bmatrix}, \quad \tilde{B} = T^{-1}B = \begin{bmatrix} \tilde{B}_{11} \\ 0 \end{bmatrix} \tag{C.6}$$

Let λ be an eigenvalue of $\tilde{A}_{22}$, and w_{22} be an associated left eigenvector. Define

$$w = T^{-T} \begin{bmatrix} 0 \\ w_{22} \end{bmatrix} \neq 0 \tag{C.7}$$

and so

$$\begin{aligned} w^T A &= \begin{bmatrix} 0 \\ w_{22} \end{bmatrix}^T T^{-1} \left(T \begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ 0 & \tilde{A}_{22} \end{bmatrix} T^{-1} \right) \\ &= \begin{bmatrix} 0 & w_{22}^T \tilde{A}_{22} \end{bmatrix} T^{-1} \\ &= \begin{bmatrix} 0 & \lambda w_{22}^T \end{bmatrix} T^{-1} \\ &= \lambda \begin{bmatrix} 0 & w_{22}^T \end{bmatrix} T^{-1} \\ &= \lambda w^T \end{aligned} \tag{C.8}$$

Similarly,

$$\begin{aligned} w^T B &= \begin{bmatrix} 0 \\ w_{22} \end{bmatrix}^T T^{-1} \left(T \begin{bmatrix} \tilde{B}_{11} \\ 0 \end{bmatrix} \right) \\ &= \begin{bmatrix} 0 & w_{22}^T \end{bmatrix} \begin{bmatrix} \tilde{B}_{11} \\ 0 \end{bmatrix} \\ &= 0. \end{aligned} \tag{C.9}$$

□

According to Theorem 10, the rank of $[B\ AB\ \dots\ A^{n-1}B]$ is determined by the relationship between the left eigenvector of A and the columns of B.

Bibliography

[1] M. J. Keeling, P. Rohani, and B. T. Grenfell, "Seasonally forced disease dynamics explored as switching between attractors," *Physica D: Nonlinear Phenomena*, vol. 148, no. 3-4, pp. 317–335, 2001.

[2] P. Holmes, R. J. Full, D. Koditschek, and J. Guckenheimer, "The dynamics of legged locomotion: Models, analyses, and challenges," *SIAM review*, vol. 48, no. 2, pp. 207–304, 2006.

[3] I. Dobson, B. A. Carreras, V. E. Lynch, and D. E. Newman, "Complex systems analysis of series of blackouts: Cascading failure, critical points, and self-organization," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 17, no. 2, p. 026103, 2007.

[4] C. R. He, W. B. Qin, N. Ozay, and G. Orosz, "Optimal gear shift schedule design for automated vehicles: Hybrid system based analytical approach," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 6, pp. 2078–2090, 2017.

[5] H. Li, A. Dimitrovski, J. B. Song, Z. Han, and L. Qian, "Communication infrastructure design in cyber physical systems with applications in smart grids: A hybrid system framework," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1689–1708, 2014.

[6] P. Varaiya, "Smart cars on smart roads: problems of control," *IEEE Transactions on Automatic Control*, vol. 38, no. 2, pp. 195–207, 1993.

[7] Y. Wei, J. H. Park, J. Qiu, and H. Jung, "Reliable output feedback control for piecewise affine systems with markov-type sensor failure," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 65, no. 7, pp. 913–917, 2017.

[8] J. Qiu, Y. Wei, and L. Wu, "A novel approach to reliable control of piecewise affine systems with actuator faults," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 64, no. 8, pp. 957–961, 2016.

[9] Y. Wei, J. Qiu, P. Shi, and M. Chadli, "Fixed-order piecewise-affine output feedback controller for fuzzy-affine-model-based nonlinear systems with time-varying delay," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 64, no. 4, pp. 945–958, 2016.

[10] Y. Wei, H. Yu, H. R. Karimi, and Y. H. Joo, "New approach to fixed-order output-feedback control for piecewise-affine systems," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 9, pp. 2961–2969, 2018.

[11] F. Lauer, G. Bloch, *et al.*, "Identification of linear hybrid systems: A geometric approach," in *2013 American Control Conference*, pp. 830–835, IEEE, 2013.

[12] F. Lauer and G. Bloch, "Hybrid system identification," in *Hybrid System Identification*, pp. 77–101, Springer, 2019.

[13] N. N. Nandola and S. Bhartiya, "Hybrid system identification using a structural approach and its model based control: An experimental validation," *Nonlinear Analysis: Hybrid Systems*, vol. 3, no. 2, pp. 87–100, 2009.

[14] C. Feng, C. M. Lagoa, and M. Sznaier, "Hybrid system identification via sparse polynomial optimization," in *Proceedings of the 2010 American Control Conference*, pp. 160–165, IEEE, 2010.

[15] T. P. Bohlin, *Practical grey-box process identification: theory and applications*. Springer Science & Business Media, 2006.

[16] A. J. Van Der Schaft and J. M. Schumacher, *An Introduction to Hybrid Dynamical Systems*, vol. 251. Springer London, 2000.

[17] 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*, vol. 1, pp. 167–172, IEEE, 2003.

[18] A. L. Juloski, S. Weiland, and W. Heemels, "A bayesian approach to identification of hybrid systems," *IEEE Transactions on Automatic Control*, vol. 50, no. 10, pp. 1520–1533, 2005.

[19] A. L. Juloski, W. Heemels, G. Ferrari-Trecate, R. Vidal, S. Paoletti, and J. Niessen, "Comparison of four procedures for the identification of hybrid systems," in *International Workshop on Hybrid Systems: Computation and Control*, pp. 354–369, Springer, 2005.

[20] R. Vidal, "Identification of pwarx hybrid models with unknown and possibly different orders," in *Proceedings of the 2004 American Control Conference*, vol. 1, pp. 547–552, IEEE, 2004.

[21] Z. Lassoued and K. Abderrahim, "An experimental validation of a novel clustering approach to pwarx identification," *Engineering Applications of Artificial Intelligence*, vol. 28, pp. 201–209, 2014.

[22] Z. Lassoued and K. Abderrahim, "New approaches to identification of pwarx systems," *Mathematical Problems in Engineering*, vol. 2013, 2013.

[23] O. Yahya, Z. Lassoued, and K. Abderrahim, "Identification of pwarx model based on outer bounding ellipsoid algorithm," in *2020 20th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*, pp. 173–178, IEEE, 2020.

[24] X. Zhu, "Semi-supervised learning literature survey," Tech. Rep. 1530, Computer Sciences, University of Wisconsin-Madison, 2005.

[25] H. Garnier, "Direct continuous-time approaches to system identification. overview and benefits for practical applications," *European Journal of Control*, vol. 24, pp. 50–62, 2015.

[26] A. Garulli, S. Paoletti, and A. Vicino, "A survey on switched and piecewise affine system identification," *IFAC Proceedings Volumes*, vol. 45, no. 16, pp. 344–355, 2012.

[27] E. R. Kalman, "Phase-plane analysis of automatic control systems with nonlinear gain elements," *Transactions of the American Institute of Electrical Engineers, Part II: Applications and Industry*, vol. 73, no. 6, pp. 383–390, 1955.

[28] L. O. Chua and A.-C. Deng, "Canonical piecewise-linear representation," *IEEE Transactions on Circuits and Systems*, vol. 35, no. 1, pp. 101–111, 1988.

[29] L. O. Chua, C. A. Desoer, and E. S. Kuh, *Linear and nonlinear circuits*. McGraw-Hill College, 1987.

[30] E. Sontag, "Nonlinear regulation: The piecewise linear approach," *IEEE Transactions on Automatic Control*, vol. 26, no. 2, pp. 346–358, 1981.

[31] E. D. Sontag, "Interconnected automata and linear systems: A theoretical framework in discrete-time," in *International Hybrid Systems Workshop*, pp. 436–448, Springer, 1995.

[32] C. Y. Lai, C. Xiang, and T. H. Lee, "Data-based identification and control of nonlinear systems via piecewise affine approximation," *IEEE Transactions on Neural Networks*, vol. 22, no. 12, pp. 2189–2200, 2011.

[33] O. Yahya, Z. Lassoued, and K. Abderrahim, "Predictive control based on fuzzy supervisor for pwarx hybrid model," *International Journal of Automation and Computing*, vol. 16, no. 5, pp. 683–695, 2019.

[34] C. Y. Lai, C. Xiang, and T. H. Lee, "Identification and control of nonlinear systems using piecewise affine models," in *Control Automation Robotics & Vision (ICARCV), 2010 11th International Conference on*, pp. 2259–2265, IEEE, 2010.

[35] M. Torchio, L. Magni, R. D. Braatz, and D. Raimondo, "Design of piecewise affine and linear time-varying model predictive control strategies for advanced battery management systems," *Journal of The Electrochemical Society*, vol. 164, no. 4, p. A949, 2017.

[36] E. F. Camacho, D. R. Ramírez, D. Limón, D. M. De La Peña, and T. Alamo, "Model predictive control techniques for hybrid systems," *Annual reviews in control*, vol. 34, no. 1, pp. 21–31, 2010.

[37] H. Ohlsson and L. Ljung, "Identification of switched linear regression models using sum-of-norms regularization," *Automatica*, vol. 49, no. 4, pp. 1045–1050, 2013.

[38] W.-X. Zhao and T. Zhou, "Weighted least squares based recursive parametric identification for the submodels of a pwarx system," *Automatica*, vol. 48, no. 6, pp. 1190–1196, 2012.

[39] 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.

[40] 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.

[41] E. Khanmirza, M. Nazarahari, and A. Mousavi, "Identification of piecewise affine systems based on fuzzy pca-guided robust clustering technique," *EURASIP Journal on Advances in Signal Processing*, vol. 2016, no. 1, p. 131, 2016.

[42] L. Bottou and C.-J. Lin, "Support vector machine solvers," *Large-scale Kernel Machines*, vol. 3, no. 1, pp. 301–320, 2007.

[43] 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.

[44] S. Kersting and M. Buss, "Recursive estimation in piecewise affine systems using parameter identifiers and concurrent learning," *International Journal of Control*, vol. 92, no. 6, pp. 1264–1281, 2019.

[45] Y. Yuan, X. Tang, W. Zhou, W. Pan, X. Li, H.-T. Zhang, H. Ding, and J. Goncalves, "Data driven discovery of cyber physical systems," *Nature Communications*, vol. 10, no. 1, pp. 1–9, 2019.

[46] R. Saravanakumar, H. S. Kang, C. K. Ahn, X. Su, and H. R. Karimi, "Robust stabilization of delayed neural networks: Dissipativity-learning approach," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 3, pp. 913–922, 2018.

[47] U. Ekong, H.-K. Lam, B. Xiao, G. Ouyang, H. Liu, K. Y. Chan, and S. H. Ling, "Classification of epilepsy seizure phase using interval type-2 fuzzy support vector machines," *Neurocomputing*, vol. 199, pp. 66–76, 2016.

[48] G. Cauwenberghs and T. Poggio, "Incremental and decremental support vector machine learning," in *Advances in Neural Information Processing Systems*, pp. 409–415, 2001.

[49] V. Breschi, D. Piga, and A. Bemporad, "Piecewise affine regression via recursive multiple least squares and multicategory discrimination," *Automatica*, vol. 73, pp. 155–162, 2016.

[50] 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.

[51] M. Vaezi and A. Izadian, "Piecewise affine system identification of a hydraulic wind power transfer system," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 6, pp. 2077–2086, 2015.

[52] V. Barnett and T. Lewis, *Outliers in statistical data*. Wiley, 1974.

[53] S. Chatterjee and A. S. Hadi, *Sensitivity Analysis in Linear Regression*, vol. 327. John Wiley & Sons, 2009.

[54] Y. Li, C. Guan, H. Li, and Z. Chin, "A self-training semi-supervised svm algorithm and its application in an eeg-based brain computer interface speller system," *Pattern Recognition Letters*, vol. 29, no. 9, pp. 1285–1294, 2008.

[55] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[56] 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.

[57] M. Tabatabaei-Pour, K. Salahshoor, and B. Moshiri, "A modified k-plane clustering algorithm for identification of hybrid systems," in *2006 6th World Congress on Intelligent Control and Automation*, vol. 1, pp. 1333–1337, IEEE, 2006.

[58] R. Tóth, *Modeling and Identification of Linear Parameter-varying Systems*, vol. 403. Berlin: Springer, 2010.

[59] S. Paoletti, J. Roll, A. Garulli, and A. Vicino, "On the input-output representation of piecewise affine state space models," *IEEE Transactions on Automatic Control*, vol. 55, no. 1, pp. 60–73, 2009.

[60] F. Rosenqvist and A. Karlström, "Realisation and estimation of piecewise-linear output-error models," *Automatica*, vol. 41, no. 3, pp. 545–551, 2005.

[61] S. Paoletti, A. L. Juloski, G. Ferrari-Trecate, and R. Vidal, "Identification of hybrid systems a tutorial," *European Journal of Control*, vol. 13, no. 2-3, pp. 242–260, 2007.

[62] L. Bako, F. Lauer, G. Bloch, *et al.*, "Identification of mimo switched state-space models," in *2013 American Control Conference*, pp. 71–76, IEEE, 2013.

[63] K. M. Pekpe and S. Lecoeuche, "Online clustering of switching models based on a subspace framework," *Nonlinear Analysis: Hybrid Systems*, vol. 2, no. 3, pp. 735–749, 2008.

[64] L. Bako, G. Mercère, R. Vidal, and S. Lecoeuche, "Identification of switched linear state space models without minimum dwell time," *IFAC Proceedings Volumes*, vol. 42, no. 10, pp. 569–574, 2009.

[65] S. Kersting and M. Buss, "Concurrent learning adaptive identification of piecewise affine systems," in *53rd IEEE Conference on Decision and Control*, pp. 3930–3935, IEEE, 2014.

[66] S. Kersting and M. Buss, "Online identification of piecewise affine systems," in *2014 UKACC International Conference on Control*, pp. 86–91, IEEE, 2014.

[67] L. Bako, K. Boukharouba, E. Duviella, and S. Lecoeuche, "A recursive identification algorithm for switched linear/affine models," *Nonlinear Analysis: Hybrid Systems*, vol. 5, no. 2, pp. 242–253, 2011.

[68] J. Wang and T. Chen, "Online identification of switched linear output error models," in *2011 IEEE International Symposium on Computer-Aided Control System Design (CACSD)*, pp. 1379–1384, IEEE, 2011.

[69] R. Vidal, "Recursive identification of switched arx systems," *Automatica*, vol. 44, no. 9, pp. 2274–2287, 2008.

[70] S. Boyd and S. S. Sastry, "Necessary and sufficient conditions for parameter convergence in adaptive control," *Automatica*, vol. 22, no. 6, pp. 629–639, 1986.

[71] G. Chowdhary and E. Johnson, "Concurrent learning for convergence in adaptive control without persistency of excitation," in *49th IEEE Conference on Decision and Control (CDC)*, pp. 3674–3679, IEEE, 2010.

[72] S. Xue, B. Luo, D. Liu, and Y. Yang, "Constrained event-triggered H_∞ control based on adaptive dynamic programming with concurrent learning," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2020.

[73] G. De La Torre, G. Chowdhary, and E. N. Johnson, "Concurrent learning adaptive control for linear switched systems," in *2013 American Control Conference*, pp. 854–859, IEEE, 2013.

[74] S. N. Goldar, M. Yazdani, and B. Sinafar, "Concurrent learning based finite-time parameter estimation in adaptive control of uncertain switched nonlinear systems," *Journal of Control, Automation and Electrical Systems*, vol. 28, no. 4, pp. 444–456, 2017.

[75] A. Parikh, R. Kamalapurkar, and W. E. Dixon, "Integral concurrent learning: Adaptive control with parameter convergence using finite excitation," *International Journal of Adaptive Control and Signal Processing*, vol. 33, no. 12, pp. 1775–1787, 2019.

[76] G. H. Golub, P. C. Hansen, and D. P. O'Leary, "Tikhonov regularization and total least squares," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 1, pp. 185–194, 1999.

[77] S. Gazzola and J. G. Nagy, "Generalized arnoldi–tikhonov method for sparse reconstruction," *SIAM Journal on Scientific Computing*, vol. 36, no. 2, pp. B225–B247, 2014.

[78] A. N. Tikhonov, A. Goncharsky, V. Stepanov, and A. G. Yagola, *Numerical methods for the solution of ill-posed problems*, vol. 328. Springer Science & Business Media, 2013.

[79] A. Buccini, M. Donatelli, and L. Reichel, "Iterated tikhonov regularization with a general penalty term," *Numerical Linear Algebra with Applications*, vol. 24, no. 4, p. e2089, 2017.

[80] M. Kim, S. H. Yoon, P. A. Domanski, and W. V. Payne, "Design of a steady-state detector for fault detection and diagnosis of a residential air conditioner," *International Journal of Refrigeration*, vol. 31, no. 5, pp. 790–799, 2008.

[81] S. Narasimhan, C. S. Kao, and R. Mah, "Detecting changes of steady states using the mathematical theory of evidence," *AIChE Journal*, vol. 33, no. 11, pp. 1930–1932, 1987.

[82] R. M. Bethea and R. R. Rhinehart, *Applied Engineering Statistics*, vol. 121. CRC Press, 1991.

[83] B. K. Ghosh and J. Rosenthal, "A generalized popov-belevitch-hautus test of observability," *IEEE Transactions on Automatic Control*, vol. 40, no. 1, pp. 176–180, 1995.

[84] M. S. Branicky, "Multiple lyapunov functions and other analysis tools for switched and hybrid systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 4, pp. 475–482, 1998.

[85] D. Liberzon, *Switching in systems and control*. Springer Science & Business Media, 2003.

[86] G. Chowdhary, T. Yucelen, M. Mühlegg, and E. N. Johnson, "Concurrent learning adaptive control of linear systems with exponentially convergent bounds," *International Journal of Adaptive Control and Signal Processing*, vol. 27, no. 4, pp. 280–301, 2013.

[87] A. Hakem, K. M. Pekpe, and V. Cocquempot, "Parameter-free method for switching time estimation and current mode recognition," in *2010 Conference on Control and Fault-Tolerant Systems (SysTol)*, pp. 879–884, IEEE, 2010.

[88] T. Zouari, K. M. Pekpe, V. Cocquempot, and M. Ksouri, "Active mode recognition of switched nonlinear systems: application to fault detection and isolation," *Asian Journal of Control*, vol. 16, no. 2, pp. 345–357, 2014.

[89] A. Hakem, V. Cocquempot, and K. M. Pekpe, "Switching time estimation and active mode recognition using a data projection method," *International Journal of Applied Mathematics and Computer Science*, vol. 26, no. 4, pp. 827–840, 2016.

[90] D. Simon, *Evolutionary optimization algorithms*. John Wiley & Sons, 2013.

[91] F. Bianchi, M. Prandini, and L. Piroddi, "A randomized two-stage iterative method for switched nonlinear systems identification," *Nonlinear Analysis: Hybrid Systems*, vol. 35, p. 100818, 2020.

[92] C. Simpkins, "System identification: Theory for the user," *IEEE Robotics & Automation Magazine*, vol. 19, pp. 95–96, 06 2012.

[93] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P.-Y. Glorennec, H. Hjalmarsson, and A. Juditsky, "Nonlinear black-box modeling in system identification: a unified overview," *Automatica*, vol. 31, no. 12, pp. 1691 – 1724, 1995.

[94] X. Lin, X. Li, and J. H. Park, "Output-feedback stabilization for planar output-constrained switched nonlinear systems," *International Journal of Robust and Nonlinear Control*, vol. 30, no. 5, pp. 1819–1830, 2020.

[95] Z.-E. Lou and J. Zhao, "Immersion-and invariance-based adaptive stabilization of switched nonlinear systems," *International Journal of Robust and Nonlinear Control*, vol. 28, no. 1, pp. 197–212, 2018.

[96] J.-L. Wu, "Singular l2-gain control for switched nonlinear control systems under arbitrary switching," *International Journal of Robust and Nonlinear Control*, vol. 30, no. 10, pp. 4149–4163, 2020.

[97] H. Wang, P. X. Liu, and B. Niu, "Robust fuzzy adaptive tracking control for nonaffine stochastic nonlinear switching systems," *IEEE Transactions on Cybernetics*, vol. 48, no. 8, pp. 2462–2471, 2017.

[98] 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.

[99] A. Abate, R. C. Hillen, and S. Aljoscha Wahl, "Piecewise affine approximations of fluxes and enzyme kinetics from in vivo 13c labeling experiments," *International Journal of Robust and Nonlinear Control*, vol. 22, no. 10, pp. 1120–1139, 2012.

[100] G. Pillonetto, "A new kernel-based approach to hybrid system identification," *Automatica*, vol. 70, pp. 21–31, 2016.

[101] L. Bako, "Identification of switched linear systems via sparse optimization," *Automatica*, vol. 47, no. 4, pp. 668–677, 2011.

[102] F. Lauer, G. Bloch, and R. Vidal, "Nonlinear hybrid system identification with kernel models," in *49th IEEE Conference on Decision and Control (CDC)*, pp. 696–701, IEEE, 2010.

[103] G. Bloch, F. Lauer, *et al.*, "Reduced-size kernel models for nonlinear hybrid system identification," *IEEE Transactions on Neural Networks*, vol. 22, no. 12, pp. 2398–2405, 2011.

[104] F. Bianchi, M. Prandini, and L. Piroddi, "A randomized approach to switched nonlinear systems identification," *IFAC-PapersOnLine*, vol. 51, no. 15, pp. 281–286, 2018.

[105] N. Mangan, T. Askham, S. Brunton, J. Kutz, and J. Proctor, "Model selection for hybrid dynamical systems via sparse regression," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 475, 2019.

[106] H. Liang, G. Liu, H. Zhang, and T. Huang, "Neural-network-based event-triggered adaptive control of nonaffine nonlinear multiagent systems with dynamic uncertainties," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–12, 2020.

[107] H. Li, Y. Wu, and M. Chen, "Adaptive fault-tolerant tracking control for discrete-time multiagent systems via reinforcement learning algorithm," *IEEE Transactions on Cybernetics*, vol. 51, no. 3, pp. 1163–1174, 2021.

[108] Y. Pan, P. Du, H. Xue, and H.-K. Lam, "Singularity-free fixed-time fuzzy control for robotic systems with user-defined performance," *IEEE Transactions on Fuzzy Systems*, 2020.

[109] H. Wang, P. X. Liu, J. Bao, X.-J. Xie, and S. Li, "Adaptive neural output-feedback decentralized control for large-scale nonlinear systems with stochastic disturbances," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 3, pp. 972–983, 2019.

[110] W. He, X. Mu, L. Zhang, and Y. Zou, "Modeling and trajectory tracking control for flapping-wing micro aerial vehicles," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 1, pp. 148–156, 2021.

[111] W. He, T. Meng, X. He, and C. Sun, "Iterative learning control for a flapping wing micro aerial vehicle under distributed disturbances," *IEEE Transactions on Cybernetics*, vol. 49, no. 4, pp. 1524–1535, 2018.

[112] Z. Liu, Z. Han, Z. Zhao, and W. He, "Modeling and adaptive control for a spatial flexible spacecraft with unknown actuator failures," *Science China Information Sciences*, 2020.

[113] A. Hakem, K. M. Pekpe, and V. Cocquempot, "Fault detection and isolation for switching systems using a parameter-free method," in *Diagnostics and Prognostics of Engineering Systems: Methods and Techniques*, pp. 98–118, IGI Global, 2013.

[114] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge: Cambridge university press, 2012.

[115] W. H. Greene, *Econometric Analysis*. India: Pearson Education, 2003.

[116] P. Bertrand and G. Fleury, "Detecting small shift on the mean by finite moving average," *International Journal of Statistics and Management System*, vol. 3, pp. 56–73, 01 2008.

[117] M. Basseville, I. V. Nikiforov, *et al.*, *Detection of abrupt changes: theory and application*, vol. 104. Englewood Cliffs: Prentice hall, 1993.

[118] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*. Philadelphia: SIAM, 1994.

[119] G. Chowdhary and E. Johnson, "A singular value maximizing data recording algorithm for concurrent learning," in *Proceedings of the 2011 American Control Conference*, pp. 3547–3552, IEEE, 2011.

[120] E. Juárez-Ruiz, R. Cortés-Maldonado, and F. Pérez-Rodríguez, "Relationship between the inverses of a matrix and a submatrix," *Computación y Sistemas*, vol. 20, no. 2, pp. 251–262, 2016.

[121] M. S. Bartlett, "An inverse matrix adjustment arising in discriminant analysis," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 107–111, 1951.

[122] R. D. Nowak, "Nonlinear system identification," *Circuits, Systems and Signal Processing*, vol. 21, no. 1, pp. 109–122, 2002.

[123] N. Cristianini, J. Shawe-Taylor, *et al.*, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge university press, 2000.

[124] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge university press, 2004.

[125] I. El-Naqa, Y. Yang, M. N. Wernick, N. P. Galatsanos, and R. M. Nishikawa, "A support vector machine approach for detection of microcalcifications," *IEEE Transactions on Medical Imaging*, vol. 21, no. 12, pp. 1552–1563, 2002.

[126] D. M. Christopher, R. Prabhakar, and S. Hinrich, "Introduction to information retrieval," *An Introduction To Information Retrieval*, vol. 151, no. 177, p. 5, 2008.

[127] J. Weston and C. Watkins, "Multi-class support vector machines," Tech. Rep. CSD-TR-98-04, Computer Science, Royal Holloway, University of London, 1998.

[128] H. K. Khalil, *Nonlinear systems*. Prentice hall press, 2002.

Own Publications

[129] 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.

[130] Y. Du, F. Liu, J. Qiu, and M. Buss, "Online identification of piecewise affine systems using integral concurrent learning," *IEEE Transactions on Circuits and Systems I: Regular Papers*, pp. 1–13, 2021.

[131] Y. Du, F. Liu, J. Qiu, and M. Buss, "A novel recursive approach for online identification of continuous-time switched nonlinear systems," *International Journal of Robust and Nonlinear Control*, 2021.