Lehrstuhl für Steuerungs- und Regelungstechnik
Technische Universität München

Adaptive Identification and Control of Uncertain Systems with Switching

Stefan Kersting

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

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender: Prof. Dr. Majid Zamani

Prüfer der Dissertation:

1. Prof. Dr.-Ing./Univ. Tokio Martin Buss
2. Prof. Dr.-Ing. Matthias Althoff

Die Dissertation wurde am 12.09.2017 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 22.02.2018 angenommen.

Acknowledgment

This thesis completes my research conducted over the last years at the Chair of Automatic Control Engineering, LSR, at the Technical University of Munich, Germany. I have experienced an excellent and inspiring atmosphere, created by a diverse team of clever, open-minded and helpful people. Hence, I will always keep this time in good memory. I would now like to take the opportunity and acknowledge those people who supported me in achieving my goals.

Foremost, I would like to thank my thesis advisor Prof. Martin Buss for his constant support. The level of trust and scientific freedom he granted me in combination with his confidence in my abilities were a steady source of encouragement for my research. Also his insightful comments and recommendations were valuable contributions to my work and are highly appreciated.

Furthermore I want to thank Prof. Majid Zamani and Prof. Matthias Althoff for their constructive feedback on this work.

I also want to express my gratitude to Prof. Sandra Hirche and Dr.-Ing. Marion Leibold for sharing their broad knowledge on control-theoretic topics with me and always having an open ear for my questions.

I thank the whole team at LSR and ITR for both the productive and the fun times we shared. A special thanks goes to my friends and colleagues in the Shrine team: Sotiris Apostolopoulos, Philine Donner, Stefan Friedrich, Rameez Hayat, Fangzhou Liu, Alexander Pekarovskiy, Markus Schill and Michaela Semmler. Without our regular discussions and the constructive feedback, this work would not have been possible.

This thesis also benefited from the excellent work summarized in various student thesis. I am particularly grateful for the commitment and the contributions of my excellent students Patrick Bareiß, Andreas Kunz, Dennis Ossadnik, Philipp Schmitt, Jannick Pfort, Heiko Stüber and Michael Wagner.

Most importantly, no words can fully express my gratefulness to my family. I would like to thank my parents Elisabeth and Franz-Josef and my brother Christoph for their love, continuous support and valuable advice in all these years. Last but not least, I am forever grateful to my dear wife Joyce for her love and encouragement as well as her understanding for me forgetting the time while working on this thesis. Without her, I would not be, where I am today.

Ludwigshafen, May 2018 Stefan Kersting

Abstract

This thesis is motivated by two dominant challenges in the form of switching behaviors and parameter uncertainty arising in the context of complex control systems in unstructured environments. Traditional linear control concepts, which – due to their simplicity – are still favored by practitioners in industry, quickly exceed their capabilities and are unable to deliver the desired performance. This thesis combines the fields of switching and adaptation to provide new ideas, approaches and algorithms to cope with nonlinear uncertain systems and at the same time preserve an almost linear framework to support acceptance among practitioners.

First, the concepts of recursive identification and adaptive control are extended from linear systems to the class of piecewise affine systems, which experienced a steady increase in popularity over the past decades. In piecewise affine systems, the state-input space is partitioned and each partition is assigned some local, linear dynamics. Hence, by increasing the number of partitions and subsystems, piecewise affine systems can almost arbitrarily well approximate nonlinear systems.

Assuming uncertain or unknown parameters, the task of recursive identification is to retrieve all system parameters based on measurements of input and output signals. The task is particularly challenging if both, the partitions and subsystem dynamics, need to be reconstructed based on measurements that do not contain information about which subsystem they belong to. In this thesis, the problem is first decoupled into the individual tasks of recursively estimating the partitions and subsystem dynamics. For estimating the partitions, this thesis builds upon results in observer design for switched systems. Subsystem dynamics are identified by extending parameter identifiers from linear to piecewise affine systems, which requires overcoming challenges related to switching and persistent excitation. Furthermore, the persistent excitation requirements are relaxed through the use of concurrent learning, that is the concurrent use of instantaneous and recorded data. Recursive identification of piecewise affine systems is ideally suited for monitoring purposes as time-varying parameters may be tracked, which enables predictive maintenance strategies to counteract wear or aging in the system.

For the task of controlling complex nonlinear or switched systems with unknown parameters, the concept of model reference adaptive control is extended from linear to piecewise affine systems. With the introduced direct and indirect model reference adaptive control laws, the practitioner may specify a desired linear reference system. During operation, adaptation ensures that the state of the controlled system asymptotically tracks the state of specified reference systems. Hence, a complex nonlinear system can then be forced to behave in a linear way, which makes the closed-loop system linear and thus easier to handle.

Furthermore, this thesis extends the state of the art in multiple models adaptive control, which is applied to control systems with parameter uncertainties too large to be stabilized by a single robust controller. In multiple models adaptive control, the controller switches between a finite set of candidate controllers, which must be distributed in the uncertainty set in such a way that each possible system configuration is stabilized by at least one candidate controller. The question of how to optimally distribute the candidate controllers is answered in this thesis with a novel coverage control-inspired algorithm.

Zusammenfassung

Diese Arbeit ist motiviert durch die zwei dominanten Herausforderungen in Form von schlatendem Verhalten und Parameterunsicherheiten, die sich bei komplexen regelungstechnischen Systemen in unstrukturierten Umgebungen ergeben. Ansätze der linearen Regelungstechnik, die aufgrund ihrer Einfachheit in der Industrie sehr beliebt sind, überschreiten dabei schnell ihre Einsatzmöglichkeiten und sind außerstande die gewünschten Leistungen zu erzielen. Diese Arbeit kombiniert die Bereiche schaltender und adaptiver Systeme und liefert neue Ideen, Ansätze und Algorithmen um mit schaltenden Systemen und Parameterunsicherheiten umzugehen. Gleichzeitig soll Linearität im weitesten Sinne beibehalten werden um die Akzeptanz sicherzustellen.

Zunächst werden die Konzepte der rekursiven Identifikation und adaptiven Regelung von linearen Systemen auf die Klasse der stückweise affinen Systeme erweitert, welche in den letzten Jahrzenten stetig an Beliebtheit gewonnen haben. Bei der Modellierung mit stückweise affinen Systemen wird der Zustandsraum partitioniert und jeder Partition wird eine lokale, lineare Dynamik zugewiesen. Ein nichtlineares System kann somit durch Vermehrung der Partitionen und Subsysteme beinahe beliebig genau durch ein stückweise affines System approximiert werden.

Unter der Annahme, dass Systemparameter unbekannt sind, ist es das Ziel der rekursiven Identifikation alle Systemparameter basierend auf einem Verlauf an Eingangs- und Ausgangssignalen zu rekonstruieren. Diese Aufgabe ist von besonderer Schwierigkeit wenn sowohl die Partitionen als auch die Subsystemparameter bestimmt werden müssen, die Messwerte aber keine Information darüber enthalten zu welchem Subsystem sie gehören. In dieser Arbeit wird das Problem daher zuerst entkoppelt betrachtet. Basierend auf Beobachtern für schaltende Systeme wird ein Algorithmus entwickelt, der die rekursive Schätzung der Partitionen ermöglicht. Für die Identifikation der Subsysteme werden Parameteridentifizierer von linearen auf stückweise affine Systeme erweitert, was besonders durch die Schaltvorgänge und die Notwendigkeit der beständigen Anregung erschwert wird. Des Weiteren wird die Annahme einer beständigen Anregung durch Concurrent Learning, das heißt die gleichzeitige Verwendung aktueller und aufgenommener Daten, gelockert. Die rekursive Identifikation eignet sich besonders für die Überwachung zeitvarianter Systemgrößen und ermöglicht somit prediktive Wartungsansätze.

Um komplexe, schaltende Systeme mit unbekannten Parametern auch regeln zu können, wird das Konzept der adaptiven Regelung auf stückweise affine Systeme erweitert. Die entwickelten Regelgesetze erlauben dem Anwender ein gewünschtes Systemverhalten in Form eines linearen Referenzsystems vorzugeben. Die Reglerparameter werden dann zur Laufzeit so adaptiert, dass der Zustand des geregelten Systems sich asymptotisch dem des Referenzsystems annähert und folgt. Der geschlossene Regelkreis verhält sich damit linear, was die Handhabung wesentlich vereinfacht.

Zusätzlich wird im Zuge dieser Arbeit der Stand der Technik im Bereich der adaptiven Regelung mit multiplen Modellen erweitert, welche bei Systemen mit sehr großer Parameterunsicherheit eingesetzt wird. Dabei wird zwischen einem Satz möglicher Regler umgeschaltet, welche im Parameterraum so verteilt sein müssen, dass jede mögliche Systemkonfiguration von mindestens einem Regler stabilisiert werden kann. Die Frage der optimalen Verteilung wird mit einem neuen, durch Coverage Control inspirierten, Algorithmus beantwortet.

Contents

Notations

Abbreviations

AHH	Adaptive hinging hyperplane
BPWARX	Piecewise affine basis function autoregressive exogenous (system)
CL	Concurrent learning
CQLF	Common quadratic Lyapunov function
DBSCAN	Density-based spatial clustering of applications with noise
ELC	Extended linear complementary (system)
EM	Expectation maximization
GHH	Generalized hinging hyperplane
GUAS	Global uniform asymptotic stability
LC	Linear complementary (system)
LCF	Left coprime factorization
LMI	Linear matrix inequality
LPV	Linear parameter varying (system)
LTI	Linear time invariant (system)
LTV	Linear time varying (system)
M-RLP	Multicategory-Robust linear programming
M-SVM	Multicategory-Support vector machines
MARS	multivariate adaptive regression splines
MAX FS	Maximum feasible subsystems
MLD	Mixed logical dynamical (system)
MMAC	Multiple models adaptive control
MMPS	Max-min-plus-scaling (system)
MRAC	Model reference adaptive control
PE	Persistently exciting / persistent excitation
PHA	Probabilistic Hybrid Automata
PrARX	Probability weighted autoregressive exogenous (system)
PWARX	Piecewise autoregressive exogenous (system)
PWA	Piecewise affine (system)
PWL	Piecewise linear (system)
RLP	Robust linear programming
RLS	Recursive least squares
SARX	Switched autoregressive exogenous (system)
SVD	Singular value decomposition
SVM	Support vector machines
TLS	Total least squares

Conventions

Scalars, Vectors, and Matrices

Scalars and *vectors* are denoted by lower case letters. *Matrices* are denoted by upper case letters. The elements of a matrix M are refered to as m_{ij} (i^{th} row, j^{th} column).

Subscripts and Superscripts

$\dot{a}$	Time derivative of a
$\hat{a}$	Estimate of a
$\check{a}$	Open-loop estimate of a
$\tilde{a}$	Difference between estimated and true value of a
$\bar{a}$	Optimized switching time a
a^{-1}	Inverse of a
a^{+}	Moore-Penrose pseudoinverse of a
$a^{\top}$	Transpose of a
a^{*}	Optimal value of a
$a^{\star}$	Operating point at a
a_i	Refers to a of the i-th subsystem
a_c	Point a is the center of an ellipsoid
a_e	Equilibrium at a
$\dot{a}^{C}$	Update law for a associated with current data
$\dot{a}^{R}$	Update law for a associated with recorded data
$a_{(i)}$	Refers to the i-th element of a vector a

Number Sets

$\mathbb{N}$	Natural numbers
$\mathbb{R}$	Real numbers
$\mathbb{R}^{+}$	Positive real numbers
$\mathbb{C}$	Complex numbers

Matrix and Vector Norms

For a vector $x \in \mathbb{R}^n$, the following norm definitions exist.

ℓ_0-norm:	number of non-zero entries in x
ℓ_1-norm:	$\|x\|_1 = \sum_{i=1}^{n} \|x_i\|$
ℓ_2-norm:	$\|x\|_2 = \sqrt{\sum_{i=1}^{n} \|x_i\|^2}$
ℓ_∞-norm:	$\|x\|_\infty = \max_i \|x_i\|$

For a matrix $X \in \mathbb{R}^{m \times n}$, the following norm definitions exist.

Frobenius norm: $\|x\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |x_{ij}|^2}$

Symbols

General

g	Nonlinear vector function to describe nonlinear systems
J	Cost or performance signal
$\mathcal{N}$	Normal distribution
P	Positive definite matrix in a quadratic Lyapunov function
s	Complex number in Laplace transformation
t	Time
T	Time interval
u	Input of a dynamic system
v_x	State-measurement noise
v_y	Output-measurement noise
V	Lyapunov function
W	Bound on the derivative of Lyapunov function
x	State of a dynamic system
y	Output of a dynamic system
δ	Small constant
ϵ	Small constant
θ	System parameter vector
τ	Time in integral

State-Space Systems

A	State or system matrix in a state-space representation
B	Input matrix in a state-space representation
b_k	k-th column of input matrix B
C	Output matrix in a state-space representation
D	Feedthrough matrix in a state-space representation
f	Affine input vector of an affine LTI-system
g	Affine feedthrough vector of an affine LTI-system
n	Dimension of the system state x
o	Dimension of the system output y
p	Dimension of the system input u
Δu	Deviation of the input from the operating point
Δx	Deviation of the state from the operating point

ARX Systems

$\mathcal{A}$	Cluster of regressor vectors
d	Dimension of regressor vector
n_a, n_b	Model orders
r	Regressor

$\mathcal{R}$	Polyhedral region in regressor domain
φ	Extended regressor

Switched Systems

A_i	System matrix of the i-th subystem of a switched linear system
B_i	Input matrix of the i-th subystem of a switched linear system
C_i	Output matrix of the i-th subystem of a switched linear system
f_i	Affine input vector ot the i-th subystem of a switched affine system
h	Vector defining a single hyperplane
$\mathcal{H}_i$	Matrix defining the hyperplanes of region Ω_i
$\mathcal{I}$	Set of subsystem indices $\mathcal{I} := \{1, \ldots, s\}$
N	Number of data samples for identification
s	Number of subsystems in a switched system
t_j	Switching time of j-th switch
μ_i	Number of hyperplanes defining Ω_i
σ	Switching signal
τ	Switching time
τ^+	right limit of switching time τ
τ^-	left limit of switching time τ
$\tau_{i,q}^{\text{in}}$	Switching time: q-th activation of subsystem i
$\tau_{i,q}^{\text{out}}$	Switching time: q-th deactivation of subsystem i
χ_i	Indicator function of subsystem i
Ω_i	Polyhedral region to define state-space partition associated with subsystem i
$\preceq_{[i]}$	List of operators $\leq$ and $<$ for the i-th system

Hyperplane Estimation

$\mathcal{A}$	Matrix of data points
$\mathcal{B}$	Vector of data points
$\mathcal{E}_\mathcal{A}, \mathcal{E}_\mathcal{B}$	Slack variables
L	Observer gains
m	Number of switches
s_i	Performance state
$\mathcal{S}_\Delta$	Threshold on performance signal
$\mathcal{X}_\Delta$	Threshold on state mismatch
U_i	Integral over observability Gramian
V	Matrix in singular value decomposition $U\Sigma V^\top$
X_i^*	Optimized initial state
X	Vector of unknown hyperplane parameters
z_i	Auxiliary states
T_δ	Time frame for mode estimation
T_Δ	Time frame for switch detection
ϕ	Observability Gramian

Adaptive Systems

$a(s)$	Denominator in transfer function: $a(s) = \det(sI_n - A)$
A_m	Stable design matrix in parameter identifiers or reference system
B_m	Input matrix of reference system
e	Tracking error
f_m	Affine input vector of reference system
$\mathcal{F}_{u_k}(\omega_{kl})$	Peak in the spectral distribution S_{u_k} at frequency ω_{kl}
$H_f(s)$	Transfer function from affine input f to internal state z
$H_k(s)$	Transfer function from input u_k to internal state z
$\bar{H}_k(s)$	Modified transfer function: $a(s)H_k(s)$
k_f	Control gains for affine feedforward
K_r	Control gains for reference signal feedforward
K_x	Control gains for state-feedback
ℓ_k	k-th column of identity matrix
M_{si}	Matrix in Lyapunov function for MRAC: $M_{si} = (K_{ri}S_i)^{-1}$
$Q_{\tilde{x}}, Q_e, Q_\mathrm{m}$	Positive definite matrices satisfying for instance $A_\mathrm{m}^\top P + PA_\mathrm{m} = -Q_e$
r	Reference signal
$R_z(0)$	Auto covariance of z
s	Laplace variable
S_i	Known design matrix of subsystem i in multivariable MRAC
S_1	Spectral distribution of constant input
S_u	Spectral distribution of the scalar signal u
$U(s)$	Laplace Transform of u
$\tilde{x}$	Prediction error
x_m	State of reference system
$\mathcal{X}, \mathcal{Y}, \mathcal{Z}$	Elements of auxiliary vector χ
z	Internal state that needs to be PE
$Z(s)$	Laplace Transform of z
$\mathit{\Gamma}$	Adaptive learning rate (scalar)
Γ	Matrix of adaptive learning rates or scaling constants
$\delta(\omega)$	Unit function of ω
Δ_{A_m}	Maximum difference between reference system matrices
$\varepsilon_A, \varepsilon_B, \varepsilon_f$	Closed-loop estimation errors in MRAC
$\tilde{\vartheta}$	Vector in MRAC framework formed with control gain errors
$\tilde{\vartheta}^B$	Vector in MRAC framework formed with control gain errors and input matrix
ω	Frequency
ω_{kl}	l-th frequency in the input u_k
χ	Auxiliary vector to show positive definiteness of R_z
Ψ	Matrix built with vector z: $\Psi = z \otimes I_n$

Concurrent Learning

E	Matrix to define an ellipsoid in the parameter space
q	Number of recorded history stack triplets

Q_θ	Symmetric positive definite matrix formed from Ξ
u_{i_j}	j-th recorded input measurement for subsystem i
U_i	History stack of measurements u_{i_j}
v_w	Process noise in fixed point smoothing
w	Extended state for fixed point smoothing
x_{i_j}	j-th recorded state measurement for subsystem i
$\dot{x}_{i_j}$	j-th recorded state derivative for subsystem i
X_i	History stack of measurements x_{i_j}
$\dot{X}_i$	History stack of measurements $\dot{x}_{i_j}$
δ	Error between true derivative and measured derivative
Δ	Error term resulting from δ
ε	Error between estimated and recorded state derivative
ϑ^C	Threshold for current-data based adaptation
ϑ^R	Threshold for recorded-data based adaptation
Θ_j	Subspace in the parameter space associated with j-th triplet
$\nu_{\max}$	Maximal semi-principal axis of an ellipsoid
ξ	Vector formed with history stack measurements
Ξ	Symmetric positive definite matrix formed with history stack measurements
υ	Threshold for populating history stacks
φ	Angle between current- and recorded-data based adaptation
$\bar{\varphi}$	Filtered version of φ
Φ	State-input vector

Multiple Models

$a_{\mathrm{m},k}$	k-th diagonal entry of the design matrix A_m
d	Dimension of parameter vector
M	Number of redistributed models
N	Number of models in MMAC
$\mathcal{R}$	Redistribution subspace
$\mathcal{R}_{\mathrm{imp}}$	Subspace of improved parameter estimates
$\mathcal{V}$	Auxiliary matrix in MMAC
v_k	k-th column of $\mathcal{V}$
$\mathcal{W}$	Weighting matrix in MMAC
w_j	Weight for j-th model in performance-based weighting
x_I	State vector for state-sharing approach
x_{II}	State matrix for state-sharing approach
α	Weight on current error in performance index J of MMAC
β	Weight on past error in performance index J of MMAC
γ	Decaying rate of performance index J of MMAC
δ_r	Redistribution factor to scale $\mathcal{R}$
$\hat{\theta}_{\min}$	Best parameter vector with minimum J
ρ	Probability of obtaining improved model

Coverage Control for Multiple Models

$A_{\mathcal{C}}, B_{\mathcal{C}}, C_{\mathcal{C}}, D_{\mathcal{C}}$	Matrices in state-space realization of controller $\mathcal{C}$
$b_{\mathcal{G}_i,\mathcal{C}_i}$	Generalized stability margin for plant $\mathcal{G}_i$ with controller $\mathcal{C}_i$
$b_{\mathcal{G}_i,\max}$	Maximum generalized stability margin
$\mathcal{C}_i$	Candidate controller for i-th nominal model
$d(q, p_i)$	Metric distance between q and p_i
d_P	Power distance
$\mathsf{d}_G(\mathsf{q}_k, \mathsf{q}_l)$	Distance between q_k and q_k on Graph G
$\mathsf{d}_{G,P}(\mathsf{q}, \mathsf{p}_i)$	Power distance between q_k and p_i
E	Edge set of G
$f(d)$	Inverse control performance
F_i	Auxiliary matrix in robust controller design
$\mathcal{G}(q)$	Dynamical system with parameter vector q
$G(Q, E, d)$	Undirected weighted graph
$\mathcal{H}$	Coverage function
$\mathcal{H}_{\mathcal{V}}$	Coverage function with power diagrams
$\mathsf{H}(\mathsf{P}, \mathsf{W})$	Discrete coverage function
$\check{\mathsf{H}}(\mathsf{p}_i, \mathsf{W}_i)$	Discrete coverage of a single partition
L_i	Auxiliary matrix in robust controller design
M_i, N_i	Matrices of normalized left coprime factorization $\mathcal{G}_i = M_i^{-1} N_i$
$\mathcal{N}(\mathsf{q}_k)$	Neighborhood of q_k
p_i	Parameter vector of i-th nominal model
p_i	Nominal vertex
$\mathcal{P}$	Set of nominal system configurations p_i
P	Set of nominal vertices p_i
q	Uncertain parameter vector
q	Quantized uncertain parameter vector
$\mathcal{Q}$	Uncertainty set
$Q(G)$	Vertex set of quantized vectors q_k
R_i	Auxiliary matrix in robust controller design
S_i	Auxiliary matrix in robust controller design
$\mathcal{V}$	Power diagram (special type of Partitioning)
V	Power diagram for G
$\mathcal{W}$	Partitioning of the uncertainty set in the parameter space
W	Partitioning of graph G
$x_{\mathcal{C}}$	State of the controller
X_i	Solution of Riccati equation in robust controller design
Z_i	Solution of Riccati equation in robust controller design
$\gamma_{\mathcal{G}_i,\min}$	Inverse of maximum generalized stability margin $b_{\mathcal{G}_i,\max} = \gamma_{\mathcal{G}_i,\min}^{-1}$
Γ	Design parameter in inverse control performance
$\delta_\nu(\mathcal{G}_i, \mathcal{G})$	Vinnicombe distance (ν-gap) between $\mathcal{G}_i$ and $\mathcal{G}$
$\eta(q)$	Auxiliary point on the shortest path between p_i and q
$\kappa(\mathcal{G})$	Number of poles of $\mathcal{G}$ in the open right half plane
$\tau(\mathsf{q}_k)$	Variable storing the closes nominal vertex to q_k

$\phi(q)$	Distribution density function
$\nabla_{\mathsf{H}}(\mathsf{p}_i)$	Approximation of continuous gradient $\partial\mathcal{H}/\partial p_i$

1 Introduction

The two dominant challenges of switching behavior and uncertainties, which frequently arise in modern control systems, are the motivation for this thesis.

A dynamical system is said to have switching behavior, if it is characterized by a continuous state and a discrete mode. The discrete mode specifies the evolution of the continuous state. Hence a change in the discrete mode causes the applied vector fields of the continuous state to switch. Systems with continuous-discrete dynamics are referred to as hybrid dynamical systems [45, 102].

Various practical examples for this first challenge of switching behavior can be found. Generally, system dynamics may change as the system reaches physical limits, enters deadzones or exceeds thresholds. In robotic manipulation, for instance, contacts introduce a dominant switching behavior. The dynamics of a robot arm that moves freely through the work space differs greatly from those of a robot arm striking the surface of an object [1, 157]. In that sense the ground reaction forces in humanoid walking robots are another example of hybrid dynamics [105]. Furthermore, mechanical systems with different friction models (stick-slip), backlash in gears or dead zones in cog wheels all constitute demonstrative examples of switching behavior [47]. Another class of systems, often characterized by switching, are biological and chemical systems [4]. Switching is also found in electric systems in the form of switched-capacitor filters, modulators, analog-to-digital converters, diodes, power converters and choppers. Ideal diodes, for instance, are governed by two modes of operation: the conducting mode and the blocking mode. The switching between these two discrete modes is based on the continuous state of the system. When the applied voltage changes its sign, the diode switches its mode. These simple examples illustrate the typical correlation between the continuous state of a system and its discrete mode. Such phenomena complicate the analysis and controller design for dynamical systems, and motivate the research results summarized in this thesis.

Switching may also be introduced my control engineers themselves. One such possibility is to add a discrete decision logic in the control framework which orchestrates the switching between multiple continuous feedback strategies [43, 170, 262]. Such hybrid control strategies result in improved control performance and robustness at the cost of increased complexity. Another popular control engineering framework, which introduces switching, is the class of piecewise linear or affine systems [267]. In this class, complex nonlinear dynamics are linearized around multiple operating points. Together with a suitable partition of the state-input space, the original nonlinear system is approximated by switching between a finite set of linear dynamics.

The second challenge for control engineers are uncertainties in the system. In the ideal case, a white-box model of the considered system is available. That means all mechanisms in the system are precisely known, which enables the application of standard analysis and control-synthesis tools. In practice, however, the majority of systems are either described

by black-box or gray-box models. Gray-box models are characterized by limited prior knowledge about the internal mechanisms. This is, for instance, the case if the underlying differential equations can be derived from first principles and only the parameters in these differential equations are unknown. If all parameters in the differential equations were perfectly known, a white-box model would arise. Hence, gray-box models are characterized by known differential equations with uncertainties in system parameters. Given that the first principles governing the system dynamics are unknown or too complicated to be modeled, the system is referred to as a black-box model. Due to the unknown internal structure, the best option is to parametrize a model capable of approximating the input-output behavior of the black-box model. Overall the less knowledge is available about the internal mechanisms of a dynamical system, the more complicated are its analysis and controller synthesis. Also note that control systems may operate in dynamically changing and unstructured environments, which results in time-varying and unknown parameters. This needs to be accounted for in the control loop.

In summary, uncertainty and switching constitute two critical challenges for control engineers. Despite the described complications due to switching and uncertainty, the save operation of control systems demands for provably stable feedback loops. In this context linear controllers, which are often favored by practitioners in industry due to their simplicity, quickly reach their limits and cannot deliver the desired level of performance or stability.

Various research directions have been explored to cope with these challenges. In the first place is fundamental research on general nonlinear systems [115, 127] as systems with switching are a subclass of nonlinear systems. A stronger focus on switching is adopted in [149]. Note that the complex structure of nonlinear control concepts frequently leads to rejection by practitioners. The system class of piecewise linear systems maintains the linear structure to a large extend and is still capable of modeling systems with strong switching. Due to their linear characteristic, piecewise linear systems over the past decades received a growing interest and acceptance in industry.

Uncertainties in dynamical systems are usually handled either by robust controller design or by adaptive control. In order to determine their preferred use cases, it is advisable to analyze the kind of uncertainty more precisely and discriminate small and large uncertainties. Here, uncertainties are considered small in case there exists a single fixed controller to stabilize all possible configurations. In the field of robust control [167, 226] the task is to design such a single fixed controller capable of stabilizing the considered system despite the uncertainty. If, however, the uncertainty becomes too large, there may not exist a single fixed controller to stabilize all possible configurations. In such cases, the controller must be adaptive in a sense that its parameters are adjusted according to the unknown system parameters. The field of adaptive control has brought forward a multitude of adaption schemes to stabilize systems with unknown parameters and achieve given control performance criteria [6, 114, 175, 240].

While the above introduction sets the general motivation for this work, it should be noted that the focus of this thesis is on selected theoretical aspects. The following section describes the challenges addressed in this thesis in greater detail.

1.1 Challenges

This thesis is concerned with three challenges revolving around uncertainties in dynamical systems. As adaptation has proven to be a suitable approach to cope with uncertainties and time-varying parameters, this thesis investigates three adaptive strategies characterized by switching phenomena:

Part I: Adaptive identification of piecwise affine systems

Part II: Adaptive control of piecwise affine systems

Part III: Similarity-based controller selection

In the first two strategies, i.e. the adaptive identification and control of switched systems, switching is an eminent characteristic of the considered systems. In the third strategy, switching serves as the solution to cope with large uncertainties in linear systems.

Uncertainties in Switched Systems – Identification and Control

The first two challenges are motivated by the fact that linear systems theory fails to treat the increasing complexity introduced by switching behavior. In turn, the hybrid systems framework emerged as a viable tool to cope with these problems [102, 135, 230]. Hybrid systems are generally characterized by a continuous state and a discrete mode. The discrete mode stands for the current mode of operation and thus determines the way in which the continuous state evolves. Logic rules determine the evolution of the discrete mode. The continuous state on the other side is governed by a set of differential equations depending on the discrete mode. Another specialty in hybrid systems are jumps in the continuous state at switching times.

The class of *Piecewise Affine (PWA) Systems* forms a popular framework to model certain hybrid systems without jumps in the continuous state [109, 267]. In PWA systems, the continuous state-input space is partitioned into various regions and each region is assigned some local, linear dynamics. These linear dynamics govern the evolution of the continuous state of the system. The discrete state of the system on the other side is the index of the region that comprises the current state and input of the system. Figure 1.1 visualizes an exemplary state-space partition and shows a trajectory governed by a piecewise linear dynamic. Since PWA systems unite the simple linear structure with applicability to nonlinear and switching systems, they experienced growing interest over the last years. The recent trend relies on a long history of research on PWA systems, which resulted in a series of relevant textbooks [119, 135, 149, 267].

The history of PWA systems begins in the middle of the 20th century. The modeling of saturated systems with a partitioning of the state space by Kalman [124] in the 1950s is among the first works towards a qualitative understanding of piecewise linear systems. In the 1970s, PWA models experienced their first major growth in interest, when the circuit community was in need for efficient simulation and analysis tools [57, 58] for large-scale circuits with piecewise linear elements such as diodes. In the 1980s Sontag provided pioneering insights into discrete-time PWA systems that are still appreciated today [227, 228].

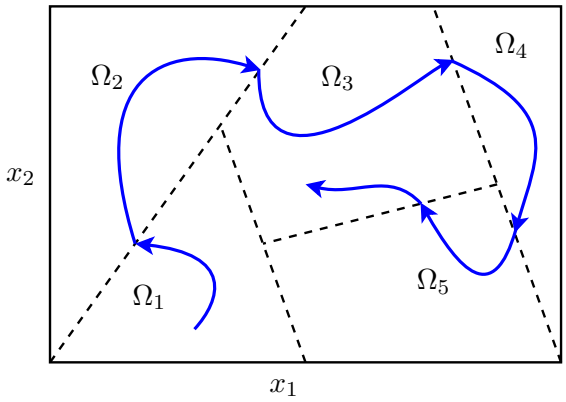


Fig. 1.1: States x_1 and x_2 of a continuous state-space trajectory traversing the partitions Ω_1 to Ω_5 of a PWA system.

Today, PWA systems are particularly popular for two reasons. First, PWA maps are known to have universal approximation properties for certain classes of nonlinear systems [150, 151]. By increasing the number of subsystems in a suitable way, nonlinearities can be approximated almost arbitrarily well. This approximation property also enables a precise analysis of rule-based fuzzy systems [86, 87]. The second reason for the popularity of PWA systems is that they can model switching and hybrid (discrete-continuous) phenomena. The equivalence between PWA systems and other classes of hybrid systems such as mixed logical dynamical (MLD) systems, linear complementarity (LC) systems, extended linear complementarity (ELC) systems or max-min-plus-scaling (MMPS) systems was shown by Heemels *et al.* [109].

Overall, the framework of PWA systems proves useful for modeling and control of many practical systems under the influence of switching. In order to apply PWA systems in practice, the standard tools available for linear systems need to be extended to PWA systems. While this extension is highly motivating, it is not trivially accomplished due to the switching behavior of PWA systems. Switches affect stability and other system properties in various ways and must thus be treated carefully [149]. Active research in recent years has successfully extended various aspects for this framework such as modeling [119, 120, 245], analysis [85, 113, 224, 231, 245] and control [119, 208, 209]. Various open questions remain in fields of controllability [28] and observability [22, 26, 48, 238], recursive identification and adaptive control of PWA systems. The later two open fields constitute the first two challenges considered in this thesis. A more detailed revision of achievements in the respective fields will be given in Chapter 2 and Chapter 7.

The problem in the recursive identification of PWA systems is to estimate both the unknown state-space partitions and the unknown subsystem parameters based on continuous measurements of the inputs and outputs. The benefit of a recursive implementation is furthermore that time-varying system parameters are tracked. This makes the presented identification algorithms particularly useful for monitoring purposes as parameter drifts due to wear or aging are recognized. Such insights enable predictive maintenance solutions and increase the reliability and safety of the controlled system.

Closely related to the recursive identification of PWA systems is their adaptive control. Adaptive control is applicable if the subsystem parameters of the PWA system are un-

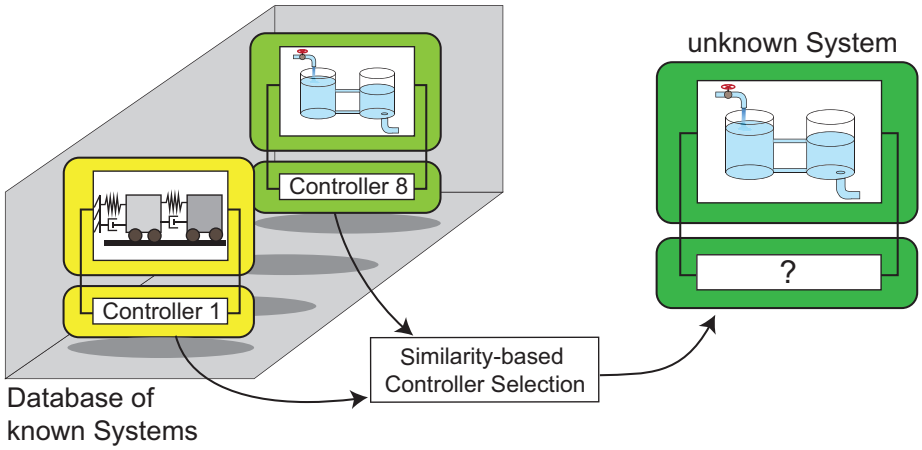


Fig. 1.2: The concept of similarity-based controller selection is applied in Multiple Models Adaptive Control or Supervisory Control.

known but a controller satisfying precise performance specifications is required. In that case, adaptive control algorithms implement time-varying control gains which are updated during operation to match the design specifications of the closed-loop system. Challenges in the context of PWA systems arise due to the switching behavior which must be considered in the stability analysis.

Controlling Uncertain Systems with Switching Controllers

As pointed out above, large parameter uncertainties in a control system demand for some sort of adaptation in the controller. The third challenge considered in this thesis is related to the concept of similarity-based controller selection as it is applied in the popular frameworks of Multiple Models Adaptive Control [2, 3, 107, 112, 177] or Supervisory Control [110, 111, 169, 170, 190]. The central idea is to switch between a set of controllers. Therefore, in this framework, adaptation is not carried out by continuously updating the controller parameters as it is the case in most adaptive controllers. Here, adaptation is rather a discrete decision logic to reason at every time instance which controller is best suited for the unknown system. Hence, switching is the solution to cope with large uncertainties in the controlled system.

Figure 1.2 visualizes the concept of similarity-based controller selection. Initially, the database of known systems on the left is established. For each system in this database, a nominal controller is designed such that it achieves the desired performance for the chosen nominal system with additional uncertainties. During operation, this database of system-controller pairs faces an unknown system for which a suitable controller needs to be found and put into feedback. To decide upon one of the controllers, the input-output behavior of the unknown system with an arbitrary controller is compared to the behavior of all database systems. The controller of the database system, which describes best the measurements, is put in feedback with the unknown system.

Various metrics exist to define distances between dynamical systems and perform the anticipated comparison. In preliminary studies for this thesis [275, 282, 290], it was found that the performance index in Multiple Models Adaptive Control is best suited for the task

of similarity evaluation between the controlled system and the database systems. Other distance measures for dynamical systems such as the ν-gap metric [253], the Binet-Cauchy Kernels [254] and the Martin distance [161] proved less useful for the task of evaluating similarity based on input-output measurements.

Multiple Models Adaptive Control is a highly modular framework which can be divided into further subproblems. Most research is performed on the design of provably stable strategies to orchestrate the switching of database controllers [44, 111, 112, 170]. In those works, the database of system-controller pairs is usually assumed to be known. Few works like [2] are concerned with the design of suitable databases. Hence, the optimal design of system-controller pairs remains a challenging open issue and will be treated in this thesis as the third challenge associated with uncertainties and switching.

1.2 Major Contributions and Thesis Outline

This thesis provides solutions to the above motivated challenges arising for dynamical systems with uncertainties and switching. Before going into detail, the interested reader may refer to Appendix A for helpful background information on switching systems and adaptive control. Chapter 2 begins with an introduction to PWA systems and gives an extensive overview on the state of the art in identifying them. The recognized lack of adaptive identification algorithms for continuous-time PWA systems is filled in Chapter 3 to Chapter 6. The gained insights are readily transferred to the second challenge of adaptive control for PWA systems in Chapter 7. The third challenge is subject in Chapter 8, where an optimal controller distribution for Multiple Models Adaptive Control is proposed. Final conclusions and possible directions for future work will be given in Chapter 9.

Region Estimation in continuous-time PWA Systems (Chapter 3)

The state-of-the-art algorithms for reconstructing the state-space partitions in PWA systems were developed for discrete-time systems. Measurements are assigned to subsystems based on prediction errors. Then support vector machines linearly separate the subsystems and thus reconstruct the switching hyperplanes. Applied to continuous-time PWA systems, this traditional approach yields suboptimal switching hyperplanes, which are shifted away from the actual hyperplanes. In this thesis, a novel algorithm is designed specifically for continuous-time PWA systems. A hybrid observer reconstructs the unknown discrete mode of the system and thereby estimates the switching times. Due to inherent delays in observers, a small optimization problem is solved to reconstruct the switching times more precisely. Finally, recursive least squares is applied to fit the estimated switching hyperplanes to the obtained switching states. Besides delivering more precise estimates of the switching hyperplanes, the proposed algorithm can be implemented in a recursive way and thus allows to track time-varying hyperplanes.

These results were previously presented in [277].

Parameter Identifiers for PWA Subsystem Identification (Chapter 4)

Most identification algorithms for switching systems are heuristic in nature and consider discrete-time systems in input-output form. In Chapter 4, the concept of parameter identifiers borrowed from the adaptive control literature is extended from linear time-invariant systems to piecewise linear and piecewise affine systems with known state-space partitions. The presented algorithms eliminate the lack of provably converging parameter estimation techniques for continuous-time PWA systems.

The material presented in Chapter 4 has been published in [273, 276].

Parameter Identifiers with Concurrent Learning (Chapter 5)

The convergence of parameter estimates in parameter identifiers is tied to a persistent excitation condition, which is undesirable in practice. In order to overcome this deficit, a concurrent learning-based extension is presented in Chapter 5. By incorporating memory into the adaptation, the restrictive assumption of persistent excitation is relaxed to a linear independence of the recorded data. Recording of linearly independent data can be achieved with less excitation leading to greater acceptance in practice. As memory in the adaptation disables tracking of time-varying parameters, additional routines are proposed to manage the history stack data, and replace erroneous or outdated data by new measurements.

The contributions in the directions of concurrent learning have been published in [274, 278]. Furthermore, a journal article summarizing the presented results on parameter identifiers with concurrent learning is available in the International Journal of Control [281].

Multiple Models-based Identification of PWA Systems (Chapter 6)

The assumption of known switching hyperplanes in Chapter 4 and Chapter 5 constitutes the limiting factor. In Chapter 6, the performance index commonly applied in Multiple Models Adaptive Control is utilized to guide the identification of PWA systems with unknown switching hyperplanes. It is shown that the performance index enables sorting a set of models according to their parametric distances to the true parameter configuration. In a swarm-based algorithm, this finding is exploited in a goal-directed redistribution of swarm models such that a subset of the swarm provably converges to the right parameter configurations. In a second approach, parameter identifiers are equipped with a performance based weighting term which enables convergence to the right parameters even for unknown switching hyperplanes, given a good initialization.

Model Reference Adaptive Control for PWA Systems (Chapter 7)

In order to control uncertain PWA systems with known state-space partitions, this thesis proposes two Model Reference Adaptive Control algorithms. By choosing a linear reference model, the closed-loop system behaves linearly and the original switching behavior can be hidden from the user. The two algorithms are direct and indirect in nature. Direct MRAC is best suited if precise reference signal tracking is to be achieved. If the estimation of

subsystem parameters for monitoring purposes is desirable, indirect MRAC is preferably. For both algorithms the asymptotic convergence of all parameter estimates to their nominal values is proven under the assumption of persistent excitation.

The material on Model Reference Adaptive Control appeared in the IEEE Transactions on Automatic Control [280].

Optimal Model Distribution in MMAC (Chapter 8)

Finally, in order to cope with large parameter uncertainties, Chapter 8 presents new ideas in the field of Multiple Models Adaptive Control (MMAC). While most research in MMAC focuses on the derivation of a suitable switching logic to orchestrate the switching between the family of nominal controllers, the question of how to optimally distribute the family of nominal system-controller pairs in the uncertainty set is a long standing problem. More precisely, a set of nominal models needs to be distributed in such a way that their corresponding controllers can stabilize all possible system configurations in a large uncertainty set. The solution proposed in this thesis relies on a coverage-control formulation of the problem, in which each of the controllers covers a subset of the parameter uncertainty. The subsets are determined based on robust control tools. From the resulting coverage functional, two algorithms to optimize the controller distribution are derived. The same concept may be applied also in linear parameter-varying systems or gain scheduling, where similar covering formulations are applicable.

A journal article containing the presented material on optimal model distribution in MMAC is published in the IEEE Transactions on Automatic Control [279].

Part I

Adaptive Identification of Piecewise Affine Systems

2 Introduction to Piecewise Affine Systems and Problem Formulation

This chapter provides an introduction to piecewise affine (PWA) systems and formulates their identification problem. Section 2.1 begins with a formal definition of PWA systems in state-space form, which is the system class considered throughout this thesis. Some remarks about how they arise as the approximation of nonlinear systems are given. Furthermore, the two closely related system classes of switched autoregressive exogenous systems and and piecewise autoregressive exogenous systems are presented. Their identification poses similar challenges as PWA systems in state-space form. Therefore, Section 2.2 formulates the identification problem for all three system classes and summarizes state of the art identification algorithms. An important aspect of identifying piecewise affine systems is the reconstruction of state-space or regressor partitions, which will be subject in Section 2.3. Based on the literature review, we derive limitations in the state of the art in Section 2.4 and formulate open problems to be solved in subsequent chapters.

2.1 Definition of PWA Systems in State-Space Form

The PWA systems considered in this thesis are a special class of hybrid systems, i.e., systems whose dynamics are characterized by a continuous state and a discrete mode. The combination of continuous dynamics and discrete modes enables PWA systems to describe systems with switching behavior. PWA systems are also a special class of switched linear systems in which the switching signal is not exogenous but depends on a partitioning of the state-input space. In the following, PWA systems are defined as switched systems with partitioned state space and state-dependent switching. It is furthermore shown how a PWA approximation of a general nonlinear system is obtained by linearizing around various operating points.

2.1.1 Switched Affine Systems

A switched affine system is defined as an affine system with time-varying parameters taken from the family $\{A_i, B_i, f_i, C_i\}_{i=1}^{s}$ of $s \in \mathbb{N}$ affine dynamics

$$\begin{aligned} \dot{x}(t) &= A_i x(t) + B_i u(t) + f_i \\ y(t) &= C_i x(t) + v_y(t), \end{aligned} \tag{2.1}$$

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^p$ is the control input, $y \in \mathbb{R}^o$ is the measured output. Measurement noise is given by $v_y \in \mathbb{R}^o$. The system matrices and vectors are $A_i \in \mathbb{R}^{n \times n}$, $B_i \in \mathbb{R}^{n \times p}$ and $f_i \in \mathbb{R}^{n \times 1}$ and $C_i \in \mathbb{R}^{o \times n}$. A special case of (2.1) is the full state measurement for which the output matrices are given by identity matrices $C_i = I_n$.

While the state x defines the continuous state of the switched system, the index of the currently active subsystem should be interpreted as the discrete mode of the system. Hence, let $\mathcal{I} = \{1, \ldots, s\}$ be the *index set of admissible subsystems* and define the piecewise constant *switching signal* $\sigma(t) : \mathbb{R}^+ \to \mathcal{I}$ to take values from this index set. Then the *switched affine system* takes the form

$$\begin{aligned} \dot{x}(t) &= A_{\sigma(t)}x(t) + B_{\sigma(t)}u(t) + f_{\sigma(t)} \\ y(t) &= C_{\sigma(t)}x(t) + v_y(t). \end{aligned} \tag{2.2}$$

An alternative representation of switched systems, which is more common in adaptive control, relies on so-called indicator functions. The *indicator functions* $\chi_i(t) : \mathbb{R}^+ \to \{0, 1\}$ defines for each time instance whether the i-th subsystem is active ($\chi_i(t) = 1$) or inactive ($\chi_i(t) = 0$). As only one subsystem can be active at any time instance, the switched system (2.2) may also be expressed as

$$\begin{aligned} \dot{x}(t) &= \sum_{i=1}^{s} \left(A_i x(t) + B_i u(t) + f_i\right) \chi_i(t) \\ y(t) &= \sum_{i=1}^{s} C_i x(t) \chi_i(t) + v_y(t). \end{aligned} \tag{2.3}$$

With all affine input vectors f_i equal to zero, the systems (2.2) or (2.3) are referred to as *switched linear systems*

$$\begin{aligned} \dot{x}(t) &= A_{\sigma(t)}x(t) + B_{\sigma(t)}u(t) \\ y(t) &= C_{\sigma(t)}x(t) + v_y(t). \end{aligned} \tag{2.4}$$

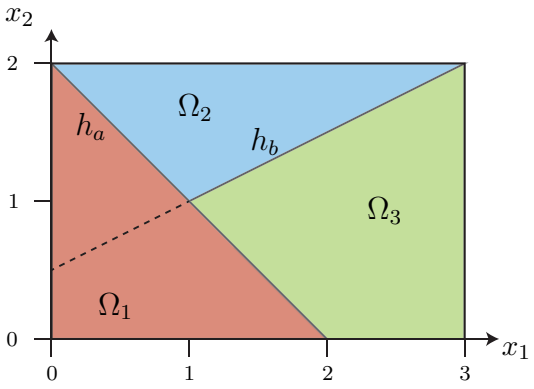
Note that the class of switched systems can be further classified according to the mechanism generating the switching signal σ or the indicator functions χ_i. One distinguishes state-dependent, time-dependent, autonomous and controlled switching. PWA systems fall into the category of *state-dependent switching* and σ or χ_i are obtained through a partitioning of the state space with a different linear subsystem dynamic associated to each partition.

2.1.2 State-Space Partitions

The state-input space $[x^\top, u^\top]^\top \in \mathbb{R}^{n+p}$ is partitioned into s *polyhedral regions* Ω_i. Let the borders of the polyhedral region Ω_i be defined by a set of $\mu_i \in \mathbb{N}$ hyperplanes in the state-input space. All points on a single *hyperplane* h fulfill the condition

$$\sum_{k=1}^{n} \alpha_k x_k + \sum_{k=1}^{p} \beta_k u_k + \gamma = h \begin{bmatrix} x \\ u \\ 1 \end{bmatrix} = 0, \tag{2.5}$$

where $h = \begin{bmatrix} \alpha_1, \ldots, \alpha_n, & \beta_1, \ldots, \beta_p, & \gamma \end{bmatrix} \in \mathbb{R}^{1\times(n+p+1)}$. Replacing the equality sign in (2.5) with inequalities ≤ 0 or < 0 yields the definition of a half-space of the state-input space



$$h_a = \begin{bmatrix} 1 & 1 & -2 \end{bmatrix}, \qquad h_b = \begin{bmatrix} -0.5 & 1 & -0.5 \end{bmatrix}$$

$$\Omega_1 = \left\{ x \in \mathbb{R}^2 \;\middle|\; \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 1 & 1 & -2 \end{bmatrix} \begin{bmatrix} x \\ 1 \end{bmatrix} \begin{bmatrix} \leq \\ \leq \\ \leq \end{bmatrix} 0 \right\}$$

$$\Omega_2 = \left\{ x \in \mathbb{R}^2 \;\middle|\; \begin{bmatrix} 0 & 1 & -2 \\ -1 & -1 & 2 \\ 0.5 & -1 & 0.5 \end{bmatrix} \begin{bmatrix} x \\ 1 \end{bmatrix} \begin{bmatrix} \leq \\ < \\ < \end{bmatrix} 0 \right\}$$

$$\Omega_3 = \left\{ x \in \mathbb{R}^2 \;\middle|\; \begin{bmatrix} 1 & 0 & -3 \\ 0 & 1 & -2 \\ -1 & -1 & 2 \\ -0.5 & 1 & -0.5 \end{bmatrix} \begin{bmatrix} x \\ 1 \end{bmatrix} \begin{bmatrix} \leq \\ \leq \\ < \\ \leq \end{bmatrix} 0 \right\}$$

Fig. 2.1: Exemplary partitioning of the x_1-x_2 space. The three partitions Ω_1-Ω_3 are highlighted in different colors and constructed based on the hyperplanes h_a, h_b and the bounding box.

$\mathbb{R}^{n+p}$ including or not including the hyperplane, respectively. Let $\mathcal{H}_i \in \mathbb{R}^{\mu_i \times (n+p+1)}$ be the combination of $\mu_i \in \mathbb{N}$ such hyperplanes, i.e.,

$$\mathcal{H}_i = \begin{bmatrix} h_{1,i}^\top & \dots & h_{\mu_i,i}^\top \end{bmatrix}^\top.$$

Furthermore, let $\preceq_{[i]}$ be a list of μ_i operators $\leq$ and $<$ to decide which hyperplanes of $\mathcal{H}_i$ belong to Ω_i. Then a compact definition for the polyhedral regions in the state-input space is given by

$$\Omega_i = \left\{ \begin{bmatrix} x \\ u \end{bmatrix} \in \mathbb{R}^{n+p} \;\middle|\; \mathcal{H}_i \begin{bmatrix} x \\ u \\ 1 \end{bmatrix} \preceq_{[i]} 0 \right\}, \tag{2.6}$$

which can thus also be interpreted as the intersection of μ_i half-spaces. 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 regions, i.e. $\Omega_i \cap \Omega_j = \emptyset$, $\forall i \neq j$. An example of such a state-space partition is given in Fig. 2.1.

2.1.3 Switching in Piecewise Affine Systems

From the set of affine subsystems (2.1) and polyhedral regions (2.6), a PWA system is obtained by assigning each of the subsystem dynamics to one of the regions. The region which contains the current state-input vector then determines the active subsystem. Hence, the switching signal in PWA systems follows

$$\sigma(t) = i \quad \Longleftrightarrow \quad \begin{bmatrix} x \\ u \end{bmatrix} \in \Omega_i. \tag{2.7}$$

Equivalently, the indicator functions in PWA systems are expressed as

$$\chi_i(t) = \begin{cases} 1, & \text{if } (x(t), u(t)) \in \Omega_i, \\ 0, & \text{otherwise.} \end{cases} \tag{2.8}$$

2.1.4 Approximating Nonlinear Systems by PWA Models

For the sake of completeness and in order to demonstrate the benefit of PWA systems compared to PWL systems, a brief revision, of how a PWA model (2.2) of a nonlinear system is obtained by linearization around multiple operating points, is appropriate.

Consider the nonlinear system

$$\dot{x}(t) = g\left(x(t), u(t)\right), \tag{2.9}$$

where $g : \mathbb{R}^{n+p} \to \mathbb{R}^n$ is a smooth (continuously differentiable) nonlinear vector function. Neglecting higher order terms in the linearization of g around an *operating point* (x_i^*,u_i^*) yields the affine model

$$\begin{aligned} \dot{x} &= g(x_i^*, u_i^*) + A_i(x - x_i^*) + B_i(u - u_i^*) \\ &= A_i x + B_i u + g(x_i^*, u_i^*) - A x_i^* - B u_i^* \\ &= A_i x + B_i u + f_i, \end{aligned}$$

with $A_i = \frac{\partial g(x,u)}{\partial x}\big|_{(x_i^*,u_i^*)}$, $B_i = \frac{\partial g(x,u)}{\partial u}\big|_{(x_i^*,u_i^*)}$ and $f_i = g(x_i^*, u_i^*) - A_i x_i^* - B_i x_i^*$. A piecewise affine model is obtained by linearizing around various operating points $\{(x_i^*, u_i^*)\}_{i=1,\dots,s}$ and partitioning the state space such that the active subsystem in each region is characterized by a minimal error with respect to the original nonlinear system.

Considering only the deviations $\Delta\dot{x}_i = \dot{x} - g(x_i^*, u_i^*)$, $\Delta x_i = x - x_i^*$ and $\Delta u_i = u - u_i^*$, the same considerations yield the PWL model

$$\Delta\dot{x}_i = A_i \Delta x_i + B_i \Delta u_i.$$

Therefore, in order to obtain PWL models, additional knowledge of the operating points x_i^* and u_i^* as well as the values of $g(x_i^*, u_i^*)$ is required. Hence, PWA models are more favorable in the design and application of adaptive algorithms as less knowledge about the system has to be introduced a priori.

2.1.5 SARX and PWARX Systems

So far we have discussed continuous-time modeling of switching and hybrid systems. A very popular alternative to this are discrete-time models in input-output form. *Switched autoregressive exogenous (SARX)* models belong to the class of input-output models, which map inputs $u \in \mathbb{R}^p$ to outputs $y \in \mathbb{R}^o$. They consist of a collection of ARX models that operate on the same regressor vector. This *regressor vector* constitutes the continuous state of the system, which collects previous input and output data in a single vector given

by

$$r(t) = \left[y(t-1)^\top, \ldots, y(t-n_a)^\top, u(t)^\top, u(t-1)^\top, \ldots, u(t-n_b)^\top\right]^\top, \tag{2.10}$$

where n_a and n_b are the model orders, and therefore $r \in \mathbb{R}^d$ with $d = on_a + p(n_b + 1)$. Let $\varphi(t) = [r(t)^\top, 1]^\top$ be the *extended regressor vector*. Then, the output $y(t)$ of a SARX system is given by the following linear mapping of the extended regressor vector:

$$y(t) = \theta_{\sigma(t)}^\top \varphi(t) + v_y(t), \tag{2.11}$$

where $v_y \in \mathbb{R}^o$ is measurement noise and $\sigma(t) = \{1, \ldots, s\}$ is the switching signal. The switching signal specifies at every sampling time t which of the s ARX models $\theta_i \in \mathbb{R}^{d+1}$ is active and hence applied for the mapping. In SARX systems, $\sigma(t)$ is an external input signal, that is usually assumed to be unknown.

Piecewise autoregressive exogenous (PWARX) models are a subclass of SARX models, in which the switching signal $\sigma(t)$ is state-dependent, i.e. depends on the current regressor vector $r \in \mathbb{R}^d$ in (2.10). Similar to the PWA system in state-space form, the *regressor domain* $\mathcal{R} \subseteq \mathbb{R}^d$ is partitioned into s polyhedral regions $\mathcal{R}_i, i = 1, \ldots, s$. The shape of the regressor domain $\mathcal{R}$ is determined by physical constraints of the system. The switching signal and hence the currently active subsystem is given by the polyhedral region that contains the regressor vector. Mathematically, we express this relationship as

$$\sigma(t) = i \quad \Longleftrightarrow \quad r(t) \in \mathcal{R}_i. \tag{2.12}$$

In order to avoid ambiguity in the switching signal, the set of polyhedral regions $\{\mathcal{R}\}_{i=1}^s$ forms a complete partitioning of the regressor domain, i.e. $\cup_{i=1}^s \mathcal{R}_i = \mathcal{R} \subseteq \mathbb{R}^d$, and no two regions overlap, i.e. $\mathcal{R}_i \cap \mathcal{R}_j = \emptyset, \forall i \neq j$.

An interesting field of research emerged from the multitude of system representations. The field of Realization Theory is concerned with converting a system from one representation to another [203]. Such conversions can be helpful in many aspects. On the one hand side, the conversion of systems might enable the transfer of theoretical properties or tools from one system representation to the other. Also, it can be desirable to identify a SARX model of a system, convert it to its state-space representation in order to apply available analysis tools or control design methods for state-space systems. In [200], the identifiability of linear switched systems was studied based on realization theory.

By defining the state of a discrete-time switched state-space system as the regressor vector (2.10) without the current control input $u(t)$, i.e.

$$x(t) = \left[y(t-1)^\top, \ldots, y(t-n_a)^\top, u(t-1)^\top, \ldots, u(t-n_b)^\top\right]^\top,$$

it is straight forward to convert a PWARX model to its state-space representation. Note, however, that the obtained realization is not guaranteed to be minimal. The minimality of discrete-time switched systems is analyzed in [201].

The input-output realization problem, i.e., realizing a SARX system for an available switched affine state-space system, has received more attention. It is known that observability of a SARX system is a sufficient condition for the existence of a suitable state-space representation [261]. Other necessary conditions for realizing input-output representations

of switched affine systems are given in [192].

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 [194, 211]. 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.

Finally, it is worth mentioning that for every PWARX system, there exists an equivalent PWA state-space system. There are, however, 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.

2.2 Problem Formulation and State of the Art in Identifying PWA Systems

After introducing the three system classes of SARX, PWARX and PWA systems in state-space form, we are now going to formulate their identification problem and present an overview of existing identification algorithms. For SARX models of the form (2.11), the most general problem formulation for their identification reads as follows:

Problem 2.1 (Identification of SARX Systems) Given N sampled pairs of input-output data $(u(t), y(t)), t = 1, \ldots, N$. Estimate the model orders n_a, n_b, the number of subsystems s, and the parameter vectors θ_i of each subsystem $i = 1, \ldots, s$. Also, reconstruct the switching signal $\sigma(t)$ associated with the recorded input-output data for $t > \max(n_a, n_b)$.

While Problem 2.1 is the most general problem formulation, the approaches to this problem found in the literature can be distinguished by which quantities are assumed to be unknown. In most algorithms, for instance, the system orders n_a and n_b are assumed to be known. Other algorithms assume an upper bound on these orders to be known.

If the number of subsystems s was known, one could (in theory easily) adjust the standard prediction error method [152] to facilitate identification of SARX models. Consider therefore a non-negative function like $\ell(x) = x^2$ or $\ell(x) = \|x\|$. The prediction error method tries to minimize the prediction error penalty $\ell(y(t) - \varphi(t)^\top \theta_{\sigma(t)})$ at every time step. Hence, for SARX systems, the corresponding optimization problem reads as follows:

$$\begin{aligned} \min_{\theta_i, \chi_i(t)} & \sum_{t=1}^{N} \left(\sum_{i=1}^{s} \ell(y(t) - \varphi(t)^\top \theta_i) \chi_i(t) \right) \\ \text{s.t.} \quad & \sum_{i=1}^{s} \chi_i(t) = 1, \forall t \\ & \chi_i(t) \in \{0, 1\}, \forall t, i. \end{aligned} \tag{2.13}$$

The challenge in (2.13) lies in the fact that it describes a mixed integer program. The computational complexity of such non-convex combinatorial optimization problems is prohibitively large and hence renders the above approach only solvable for small problems.

Nevertheless, many identification approaches discussed in the literature - and analyzed in more detail in Appendix B - are motivated by (2.13). Such approaches typically rely on (2.13) as a starting point and propose relaxations of the problem that enable efficient solvers and provide solutions close to the desired values.

Other approaches do not assume the number of subsystems s to be known. In that case, s constitutes an additional optimization variable, which allows for two extremes. For $s = 1$ on the one hand side, one model is estimated based on the entire data, which results in an unsatisfactory fit. For $s = N$ on the other side, one would fit a separate model to each data point leading to a perfect fit for the training data. This, however, would lead to over-fitting and result in poor or no generalization ability. Therefore, a trade-off between model complexity (characterized by the number of subsystems s) and model accuracy (characterized by the prediction errors) is needed. Such a trade-off certainly depends on the available input-output data. The optimal trade-off still constitutes an open issue. Therefore, different identification algorithms can be distinguished in terms of how they penalize an increase in subsystem models and hence realize the trade-off.

As PWARX systems are a subclass of SARX systems, their identification problems are closely related. We obtain the following problem formulation for PWARX models, whose continuous state follows the difference equation in (2.11) and whose discrete mode is given by (2.12).

Problem 2.2 (Identification of PWARX Systems) Given N sampled pairs of input-output data $(u(t), y(t)), t = 1, \ldots, N$. Estimate the model orders n_a, n_b, the number of subsystems s, the parameter vectors θ_i, and the polyhedral regions $\mathcal{R}_i$ of each subsystem $i = 1, \ldots, s$.

Comparing Problem 2.1 and Problem 2.2, it can be noticed that one needs to reconstruct the regions $\mathcal{R}_i$ in the PWARX case rather than the switching signal $\sigma(t)$ in the SARX case. It is, however, worthwhile to point out that one can easily reconstruct $\sigma(t)$ from (2.12), once we successfully estimated the regions $\mathcal{R}_i$ of the PWARX system.

As in Problem 2.1, the actual difficulty in Problem 2.2 depends on which quantities are assumed to be known. One simplification, which is assumed by most algorithms for the identification of PWARX systems, are known orders n_a and n_b. Still, Problem 2.2 remains challenging even for known orders. Since PWARX systems are a special case of SARX systems, an intuitive approach to solve Problem 2.2 would be to return to the combinatorial optimization problem in (2.13). Besides the already prohibitively large complexity for small systems, there is an additional flaw in case of PWARX systems. While the switching signal is arbitrary in case of SARX systems, it has to satisfy additional constraints in case of PWARX systems in terms of the polyhedral regions. As it is nontrivial to efficiently express the related constraints in the optimization problem, most algorithms neglect or soften the regional constraints during the parameter estimation phase. In such algorithms, the regressor partitioning is postponed to the final step of the algorithm, which is executed once the subsystem parameter vectors θ_i have been determined. The region estimation for PWARX and state-space systems will be presented in Section 2.3.

Most identification algorithms for PWARX systems do not allow for an automatic estimation of s. In most cases s is either assumed to be known a priori, or single subsystems are iteratively added in order to improve the fit of the obtained PWARX model.

Looking at the PWA systems in state-space form,

$$\begin{aligned} x_\delta(t) &= A_{\sigma(t)}x(t) + B_{\sigma(t)}u(t) + f_{\sigma(t)} \\ y(t) &= C_{\sigma(t)}x(t) + v_y(t), \end{aligned} \tag{2.14}$$

where $x_\delta(t) = x(t+1)$ in the discrete-time setting and $x_\delta(t) = \dot{x}(t)$ in the continuous-time case, the identification problem reads as follows.

Problem 2.3 (Identification of switched systems in state-space form) Given N sampled pairs of input-output data $(u(t), y(t)), t = 1, \ldots, N$. Estimate the model order n, the number of subsystems s, the subsystem parameters A_i, B_i, f_i and C_i of each subsystem $i = 1, \ldots, s$. Furthermore, estimate the regions Ω_i, $i = 1, \ldots, s$ or equivalently reconstruct the switching signal $\sigma(t)$, $t = 1, \ldots, N$.

Once again, the actual difficulty of Problem 2.3 depends on which quantities are available a priori. Unlike in the input-output case, however, the estimation problem still remains a challenge, even if the order n, the number of subsystems s and the switching signal $\sigma(t)$ are known for all $t = 1, \ldots, N$. This is due to the fact that the estimated subsystem parameters A_i, B_i, f_i and C_i in each region are not unique but isomorphic up to a linear state transformation. However, in order to combine the subsystems to a single PWA system, they need to share the same state basis. This transformation constitutes a nontrivial task as shown in [249].

Another central difference between identification algorithms for PWARX and PWA systems in state-space form is related to the dwell time of the system. While none of the algorithms in the PWARX case are restricted to slow switching, it is a common assumption in the state-space case. The dwell time is usually necessitated by the missing information about the state x of the system.

Table 2.1 gives an overview of available algorithms addressing Problem 2.1 to Problem 2.3. This list extends previous surveys on the matter [96, 121, 193] with more recent results. A more detailed discussion of all algorithms is presented in Appendix B for the interested reader. Typically algorithms fall into one of four categories: optimization-based methods, clustering-based methods, recursive methods and algebraic methods. Optimization-based methods seek a relaxation of the mixed-integer optimization problem or find an alternative formulation. Algebraic methods are based on the idea that multiple ARX models can be identified (independent of the switching sequence) as a single, lifted ARX model. Clustering techniques are applied at different stages of the identification process in clustering-based methods. Recursive methods estimate the subsystem parameters in a recursive way. The literature review in Appendix B reveals that there exists no provably converging, recursive identification algorithm for PWA systems in state-space form. With provably converging tools from the adaptive control literature, we fill this gap in the first part of this thesis.

Tab. 2.1: Available algorithms for the identification of different switched system representations at a glance.

	SARX	PWARX	Switched Affine SS
Optimization-based methods	[10, 11, 140, 141, 165, 185–189]	[118, 131–134, 145, 162–164, 183, 184, 210, 236, 237, 246, 263]	[16, 34, 80, 99, 172, 220]
Clustering-based methods	[223]	[23, 29, 36, 88, 97, 116, 137, 173, 235]	[35, 156, 171, 197, 198]
Recursive methods	[9, 12, 81, 82, 104, 250, 257, 258]	[12, 27, 42, 234]	[13–15, 50, 196, 242]
Algebraic methods	[17, 180–182, 250, 251]	-	-

2.3 Region Estimation

Note that the algorithms listed in the previous section (and detailed in the appendix) identify hybrid systems in two stages. First, the affine subsystems are estimated, which might require a first classification of the available data. Afterwards, the partition of the regressor space or state-input space is reconstructed based on available input-output data and the estimated subsystems. While the overview above has shown that there exists a multitude of approaches to the first problem, there exist few variations to the problem of region estimation, which are discussed in this section.

The following discussion focuses on the estimation of the polyhedral regions $\{\mathcal{R}_i\}_{i=1}^{s}$ in the regressor domain $\mathcal{R}$ of a PWARX system. Note, however, that the same arguments apply without loss of generality for partitioning the state-input space of discrete-time PWA state-space systems into polyhedral regions $\{\Omega_i\}_{i=1}^{s}$.

2.3.1 Data Classification

Before estimating the polyhedral regions $\{\mathcal{R}_i\}_{i=1}^{s}$, the output-regressor pairs $(y(k), r(k))$ (which were formed for the available input-output data $(y(t), u(t)), t = 1, \ldots, N$) must be carefully assigned to the estimated subsystems. Given the parameter vectors θ_i of s subsystems, each $r(k)$ is assigned to the subsystem $\hat{\sigma}(k)$ that minimizes the prediction error. Mathematically this is expressed as

$$\hat{\sigma}(k) = \arg \min_{i=1,\ldots,s} \left\| y(k) - \varphi^{\top}(k)\theta_i \right\|. \tag{2.15}$$

Special care must be given in this step to those measurements that are consistent with multiple subsystems. Especially at the intersection between two or more subsystems, measurements cannot be assigned to one of them in a confident way. Hence, such measurements should be labeled undecidable and be neglected during region estimation. If such measurements were assigned to the wrong region, they lead to errors during the estimation of the polyhedral regions.

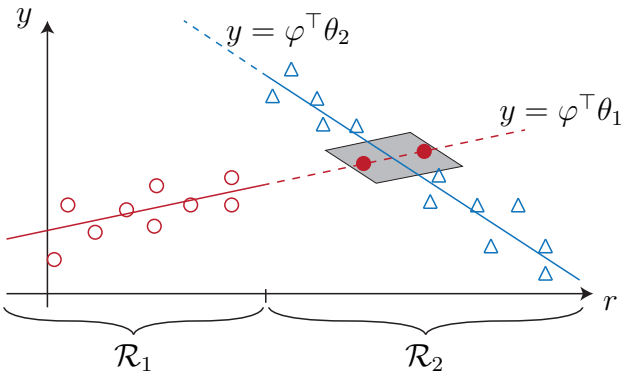


Fig. 2.2: Visualization of undecidable data points that are wrongly assigned to subsystem one due to noisy measurements.

Figure 2.2 visualizes the problem of undecidable measurements for a one dimensional PWARX system with two subsystems. The rule in (2.15) correctly assigns the measurements represented by empty circles and triangles to the regions $\mathcal{R}_1$ and $\mathcal{R}_2$, respectively. The filled circles actually belong to subsystem 2. However, due to noise, the prediction errors for subsystem 1 are lower and these points would thus be wrongly assigned. With this wrong classification, the two clusters (circles and triangles) are not linearly separable anymore which results in poorly estimated regions. To avoid this problem, all measurements inside the gray box, which is given by

$$\|y(k) - \varphi^\top(k)\theta_1\| \leq \epsilon \quad \text{and} \quad \|y(k) - \varphi^\top(k)\theta_2\| \leq \epsilon,$$

should thus be labeled undecidable. Note that some of the algorithms discussed in Appendix B deal with this problem in the subsystem parameter estimation phase already.

Finally, note that additional challenges arise if the PWARX system is described by the same subsystem dynamics in two or more unconnected regions. In such cases, the subset of data points assigned to this subsystem dynamic must be divided further such that each cluster can be described by a polytope. Otherwise, the data clusters are not linearly separable from other subsystems which would result in poorly estimated regressor partitions.

Assigning a mode $\hat{\sigma}(k)$ to each available regression vector $r(k)$, yields a total of s clusters

$$\mathcal{A}_i = \big\{ r(k) \big|\ \hat{\sigma}(k) = i \big\}, \quad i = 1, \ldots, s.$$

In order to estimate the regions of the PWARX system, we now seek a complete polyhedral partition $\{\mathcal{R}\}_{i=1}^{s}$ of the regressor domain $\mathcal{R}$ such that $\mathcal{A}_i \subseteq \mathcal{R}_i$. Recall that the polyhedral regions $\mathcal{R}_i$ were defined in terms of hyperplanes (see (2.6) in Section 2.1). The considered problem is therefore equivalent to the construction of linear classifiers (hyperplanes) that separate s sets of data points. In the following, two approaches to solve this problem are presented. In the first approach, a linear classifier is constructed for every combination of sets $(\mathcal{A}_i, \mathcal{A}_j), i \neq j$. In the second approach, a single linear classifier discriminates among all classes $\mathcal{A}_i$.

2.3.2 Region Estimation with Two-Class Linear Separation

For this approach, two sets $\mathcal{A}_i$ and $\mathcal{A}_j$ are considered at a time with the aim to construct a separating hyperplane to discriminate among the two sets. This is called a *two-class separation* problem. Performing this task for every possible combination (i, j) with $i \neq j$ amounts to solving a total of $s(s-1)/2$ two-class separation problems. A separating hyperplane in the regressor domain $\mathcal{R}$ is given by $w^\top r + \gamma = 0$. Therefore, the linear separation problem for the two sets $\mathcal{A}_i$ and $\mathcal{A}_j$ corresponds to finding $w \in \mathbb{R}^d$ and $\gamma \in \mathbb{R}$ such that

$$\begin{aligned} w^\top r(k) + \gamma > 0 \quad \forall r(k) \in \mathcal{A}_i, \\ w^\top r(k) + \gamma < 0 \quad \forall r(k) \in \mathcal{A}_j. \end{aligned} \tag{2.16}$$

Note that there might not exist a linear hyperplane which separates the two sets without errors. Data sets for which no linear classifier exists are called not linearly separable. Since measurements are noisy, the identification of hybrid systems frequently faces the problem of not linearly separable data sets. One approach to overcome this problem is to minimize the number of misclassified points in (2.16) or equivalently maximize the number of well-separated points. The literature describes the problem of finding (w, γ) such that the number of satisfied equalities in (2.16) is maximized as maximum feasible subsystem (MAX FS) problem. This approach suffers from two drawbacks: the solution of the problem may not be unique and the problem is generally known to be NP-hard (even though some heuristics have been developed and perform well in practice).

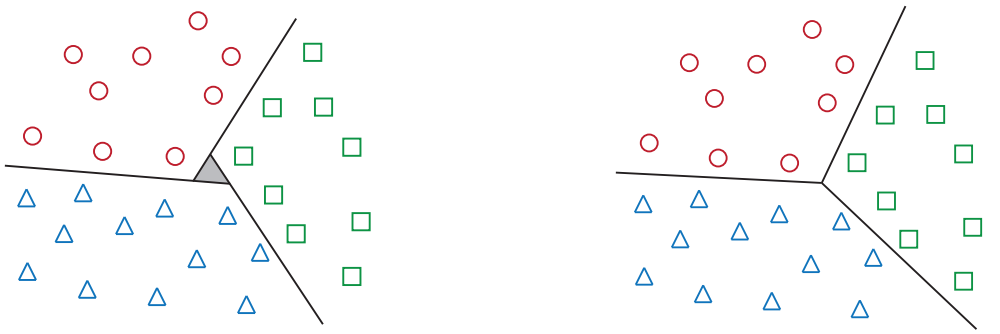
A better approach to find a linear classifier to discriminate between two not linearly separable data sets is to introduce and minimize a cost function that penalizes the error associated with miss-classification. Let us first introduces the auxiliary variable

$$z(k) = \begin{cases} 1 & \text{if } r(k) \in \mathcal{A}_i, \\ -1 & \text{if } r(k) \in \mathcal{A}_j. \end{cases} \tag{2.17}$$

We rewrite the condition (2.16) as $z(k)\left(w^\top r(k) + \gamma\right) > 1$. Furthermore, each regressor $r(k)$ is said to have a miss-classification error $v(k) = \max\{0, 1 - z(k)\left(w^\top r(k) + \gamma\right)\}$. Then, define the following optimization problem with linear inequality constraints:

$$\begin{aligned} \min_{w,\gamma} \; & \sum_k c(k)v(k) \\ \text{s.t.} \quad & z(k)\left(w^\top r(k) + \gamma\right) \geq 1 - v(k) \\ & v(k) \geq 0 \end{aligned} \tag{2.18}$$

where $c(k) > 0$ are miss-classification weights. The solution of (2.18) by means of Robust Linear Programming is presented in [31]. Another approach to solve the separation task is through *Support Vector Machines (SVM)* with linear kernels [60]. The SVM method was developed in statistical learning theory [248] and provides the optimal hyperplane that maximizes the separation margin between the two sets. It involves a quadratic optimization problem with the same constraints as in (2.18). The obtained classifier is a function of



(a) Two-Class Linear Separation

(b) Multi-Class Linear Separation

Fig. 2.3: Comparison of regressor partition obtained with Two-Class and Multi-Class Linear Separation for equal data points associated with the three different classes circles, triangles and squares.

only a subset of the training data. These data points are therefore called support vectors.

With the Two-Class Linear Separation approach $s-1$ separating hyperplanes are obtained for each subsystem. Some of these hyperplanes might be redundant if regions are not adjacent. Therefore, redundant hyperplanes are removed in a final step by standard linear programming. This reduces the number of linear inequalities, which define the i-th region $\mathcal{R}_i$, to $\mu_i \leq s-1$.

The benefit of the Two-Class approach is its computational complexity. While the number of estimated hyperplanes might be excessive in view of the redundant hyperplanes, it is computationally appealing in a sense that not all data is involved simultaneously and the estimation of hyperplanes amounts to solving simple linear or quadratic programs.

2.3.3 Region Estimation with Multi-Class Linear Separation

The use of Multi-Class Linear Separation is motivated by the limitation of Two-Class Linear Separation: there exists no guarantee that the estimated regions form a complete partition of the regressor domain. In such cases, as shown in Fig. 2.3a, there exist holes in the regressor domain which are assigned to none of the subsystems, i.e., there exist regressors $r \in \mathcal{R}$ such that $r \notin \mathcal{R}_i, \forall i$. The resulting model is then not defined on the complete regressor domain, which makes it unfeasible for certain applications. If the obtained PWARX model is applied for simulation, the unclassified regressors can be assigned to the closest region according to the following rule:

$$\hat{\sigma}(r) = \arg\min_i \min_{z \in \mathcal{R}_i} \|z - r\|.$$

For optimal control applications the possible trajectories passing through the holes in the regressor domain must be discarded, which leads to suboptimal solutions. In case an incomplete partition of the regressor domain is not acceptable, the Multi-Class Linear Separation approach should be chosen.

For *Multi-Class Linear Separation*, all sets $\mathcal{A}_i, i = 1, \ldots, s$ are considered simultaneously in the construction of a piecewise linear classifier. A first approach to multi-class classifica-

tion [247] [41] is to construct s linear classifiers that separates the i-th set of points $\mathcal{A}_i$ from the union of all the other sets. This requires the solution of s Two-Class Linear Separation problems. The difference to the above mentioned Two-Class Separation approach is that this approach consider one-against-all separations rather than one-against-one separations. Again, problems arise if each set $\mathcal{A}_i$ is not linearly separable form $\cup_{j\neq i}\mathcal{A}_j$ (the union of the other sets). Then multiply classified or unclassified points may occur in the original data set. By assigning a point to the class corresponding to the classification function that is maximal at that point, this ambiguity can be avoided.

A second approach in multi-class classification is to construct s linear classification functions in such a way that for all data points the function which corresponds to the mode of the current data point is maximal. Extensions for the classical Two-Class Separation methods mentioned above are available. Their names Multicategory SVM (M-SVM) [41] and Multicategory RLP (M-RLP) [32] indicate that they consider problems with more than two categories. Therefore all available data must be considered at the same time which makes these approaches computationally more expensive. Interestingly, the resulting optimization problems to construct the piecewise linear classifiers still rely on a single linear or quadratic program.

Refer to [31, 60, 248] for more details on the Two-Class Linear Separation problem. The Multi-Class Linear Separation problem is discussed in [32, 41, 247]. Furthermore, we refer to [191] for a comprehensive overview on the application of Two-Class and Multi-Class Linear Separation in the reconstruction of PWARX regressor partitions.

2.3.4 Recursive Region Estimation

Table 2.1 and Appendix B show that there exist various approaches to identify PWA systems recursively, e.g. [9, 12–15, 50, 196, 250, 258]. As also pointed out before, these identification problems are usually decoupled from the reconstruction of the switching hyperplanes. Note that the recursive estimation of switching hyperplanes is a nontrivial problem. This is due to the fact that the above mentioned Two-Class and Multi-Class Separation problems are computationally expensive and must hence be solved offline. To the best of our knowledge, there exist no identification algorithms for PWA systems that recursively estimate and track the switching hyperplanes.

In the student project [287], an overview of incremental support vector machines is presented in order to evaluate their application for recursive region estimation. The discussed incremental support vector machines [46, 93, 128, 136, 232] constitute a recursive realization of the linear separation problem. Two algorithms stand out within this selection. The proximal SVM algorithm [93] convinces with low memory complexities and mathematical simplicity. The obtained results are however only of approximate nature. The C&P algorithm [46, 136] is an exact online method for SVM training. Due to the exactness of the solution, updates are more complex than in the proximal approach but still considerably faster than resolving the SVM problem in batch mode for every update.

Overall, the development of recursive classification algorithms still lacks maturity to be readily applied for recursive region estimation in PWA systems. Therefore, Chapter 3 presents an alternative approach to estimate and track time varying switching hyperplanes without solving linear classification problems.

2.4 Summary and Open Problems

2.4.1 Summary

This chapter in combination with Appendix B provided a detailed overview of the state of the art in identifying hybrid systems in the form of switched linear and piecewise affine systems. The problem was formulated for three major system representations: SARX systems, PWARX systems and PWA systems in state-space form. It was furthermore pointed out that with the help of realization theory results from one representation might be transferred to the others.

An extensive list of identification algorithms was presented. Most of the discussed identification algorithms fall into one of four categories. The optimization-based methods seek to solve the underlying non-convex combinatorial optimization problem through a relaxation of the problem or by finding alternative formulations. The algebraic methods share the idea that multiple ARX models can be identified (independent of the switching sequence) as a single, lifted ARX model. In clustering-based methods, clustering techniques are applied at different stages of the identification process. Recursive methods estimate the subsystem parameters in a recursive way but can be special realizations of the other three categories that are commonly executed in batch mode. A central problem is the trade-off between computational complexity and the quality of the suboptimal solutions. Overall, efficient optimization and relaxation tools are currently in favor.

Most algorithms perform the subsystem identification independently from the reconstruction of the regressor or state-space partitioning. The state of the art in the later task relies on linear separation theory. While Two-Class Linear Separation techniques, such as Support Vector Machines, are computationally appealing, they result in incomplete partitions. Multi-Class Linear Separation delivers complete partitions but is computationally more involved.

2.4.2 Open Problems

From the above review of the state of the art in identifying hybrid systems, we conclude the following two open problems, which the subsequent chapters are concerned with.

- *Subsystem Identification*: The majority of identification algorithms focuses on discrete-time systems in input-output form. In practice, however, various control design principles rely on continuous-time state-space systems. Therefore, the identification of continuous-time systems is more desirable as advocated in [95]. Overall, a provably converging algorithm that recursively identifies continuous PWA systems in state-space form is not available in the literature. We fill this gap in Chapter 4 with parameter identifiers for PWA systems and extend their performance through concurrent learning in Chapter 5. Another extension for unknown switching hyperplanes is given in Chapter 6.

- *Region Estimation*: As there exist no identification algorithms for continuous-time PWA systems, there also exists no solution for the estimation of switching hyperplanes in case of continuous-time PWA systems. In Chapter 3, it will be shown that

the results on region estimation (revised above) do not carry over to continuous-time systems. We therefore propose an alternative formulation to the estimation of switching hyperplanes [277]. The proposed algorithm not only yields more precise estimates but at the same time allows tracking of time varying hyperplanes and does not require the removal of redundant hyperplanes.

3 Estimation of Switching Hyperplanes in Continuous-Time PWA Systems

Chapter 2 showed that the identification of piecewise affine (PWA) systems mostly consists of two steps. First, the parameters of the linear subsystems are estimated either simultaneously or iteratively. Once the subsystem dynamics are known, the partitioning of the state-input space is established [193], which was also described in Section 2.3. It was pointed out that measurements of the recorded trajectory are labeled according to the best matching subsystem. Then, differently labeled states are separated with the help of linear classifiers such as Support Vector Machines (SVMs). This chapter first shows that this state-of-the-art approach does not work well for continuous-time PWA systems. This is due to delays and hence errors associated with the data classification. Therefore, we propose an alternative approach to estimate switching hyperplanes in continuous-time PWA systems. This chapter revises and extends our previous results in [277].

The proposed approach builds upon three major components. First, an online switching signal estimator [146] estimates the switching time and state up to a certain level of precision. At the same time, the applied observer records parts of the trajectory and determines the active models before and after the actual switch. The recorded trajectory element thus contains only a single, not very precisely estimated switch. In the second step, the delay in the reconstructed switching signal is removed by solving an optimization problem. The obtained switching state is more precise and guaranteed to lie on a specific switching hyperplane. Repeating the first two steps for each switch leads to point clouds of switching states. Each switching state is thereby associated with two neighboring regions. Finally, Total Least Squares fits hyperplanes to these point clouds and thereby reconstructs the partitions in the state-input space.

The remainder of this chapter is organized as follows. Section 3.1 highlights the limitations of the state of the art in estimating switching hyperplanes for continuous-time PWA systems and formulates the problem considered in this chapter. The three steps of the hyperplane estimation algorithm are presented in Section 3.2, Section 3.3 and Section 3.4. Section 3.5 analyzes the proposed algorithm in numerical simulations. Section 3.6 summarizes the chapter and presents conclusions.

3.1 State of the Art Limitations

In order to motivate the need for an alternative switching hyperplane estimation algorithm, the problem of estimating the switching hyperplanes is first formulated for continuous-time PWA systems. Then, the state-of-the-art algorithm is shown to be not applicable in the continuous-time domain.

Consider the continuous-time PWA state-space system

$$\begin{aligned} \dot{x}(t) &= A_{\sigma(t)}x(t) + B_{\sigma(t)}u(t) + f_{\sigma(t)} \\ y(t) &= C_{\sigma(t)}x(t) + v_y(t), \end{aligned} \tag{3.1}$$

with continuous state $x \in \mathbb{R}^n$, control input $u \in \mathbb{R}^p$, output $y \in \mathbb{R}^o$ and measurement noise $v_y \in \mathbb{R}^o$. The PWA system consists of local, linear models with state matrix $A_i \in \mathbb{R}^{n\times n}$, input matrix $B_i \in \mathbb{R}^{n\times p}$, affine input vector $f_i \in \mathbb{R}^n$ and output matrix $C_i \in \mathbb{R}^{o\times n}$. Note that for the considerations in this chapter, we only assume measurements of the control input u and the noisy output y.

The piecewise constant, right continuous switching signal $\sigma(t) \in \mathcal{I} := \{1, \ldots, s\}$ indicates the currently active subsystem. The current state-input vector and the regions Ω_i determine the active subsystem by

$$\sigma(t) = i \quad \Longleftrightarrow \quad \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} \in \Omega_i, \quad i \in \mathcal{I}.$$

Hyperplanes in the state-input space define the borders of the polyhedral regions Ω_i. All points on a single hyperplane fulfill the condition

$$\sum_{k=1}^{n} \alpha_k x_k + \sum_{k=1}^{p} \beta_k u_k + \gamma = h \begin{bmatrix} x \\ u \\ 1 \end{bmatrix} = 0, \tag{3.2}$$

where $h = \begin{bmatrix} \alpha_1, \ldots, \alpha_n, & \beta_1, \ldots, \beta_p, & \gamma \end{bmatrix} \in \mathbb{R}^{1\times(n+p+1)}$. Replacing the equality sign in (3.2) with inequalities ≤ 0 or < 0 yields the definition of a half-space of the state-input space $\mathbb{R}^{n+p}$ including or not including the hyperplane, respectively. Let $\mathcal{H}_i \in \mathbb{R}^{\mu_i\times(n+p+1)}$ be the combination of $\mu_i \in \mathbb{N}$ such hyperplanes:

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

Furthermore, let $\preceq_{[i]}$ be a list of μ_i operators $\leq$ and $<$ to decide which hyperplanes of $\mathcal{H}_i$ belong to Ω_i. This yields a compact definition for the polyhedral regions in the state-input space:

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

The problem in various PWA system identification algorithms is to estimate or reconstruct the polyhedral regions Ω_i once the subsystem dynamics are known [96]. The corresponding problem discussed in this chapter thus reads as follows.

Problem 3.1 Given the subsystem parameters A_i, B_i, f_i and C_i. Estimate the polyhedral regions Ω_i characterized by $\mathcal{H}_i$, $i \in \mathcal{I}$ based on a stream of input-output data $(u(t), y(t))_{t_0}^{t_\text{end}}$.

The state-of-the-art approach to Problem 3.1 begins with data classification, i.e. a label or subsystem index $\hat{\sigma}$ is assigned to each data point. Standard SVMs can subsequently

separate the labeled state-input vectors, which yields the switching hyperplanes. The available data from PWARX systems consists of extended regressor vectors $\varphi(t) = [r(k)^\top, 1]^\top$ and corresponding outputs $y(k)$. Labels are thus easily generated by predicting the output for each subsystem parameter vector θ_i and finding the subsystem that matches best the measured output:

$$\hat{\sigma}(k) = \arg\min_{i \in \mathcal{I}} \|y(k) - \varphi(k)^\top \theta_i\|. \tag{3.3}$$

The enabling factor for this classification approach in PWARX systems lies in the fact that the model parameters θ_i directly predict the available measurement $y(k)$ based on the available regressor $r(k)$. This fact enables correct labeling even right after the transition from one region to another region in the regressor domain $\mathcal{R}$.

In the continuous-time case, such a direct prediction of the output is not possible as the available submodels in (3.1) only provide state derivatives which can usually not be measured. Furthermore, the partition is carried out in the state-input space for which measurements of the state are unavailable by assumption (in this chapter). Hence, in order to apply the same intuitive approach in continuous-time, one would need to design an observer for each subsystem to estimate the state as well as the measured output. Consider for instance the following observers

$$\begin{aligned} \dot{\hat{x}}_i &= (A_i - L_i C_i)\hat{x}_i + L_i y + B_i u + f_i \\ \hat{y}_i &= C_i \hat{x}_i, \end{aligned} \tag{3.4}$$

where $L_i \in \mathbb{R}^{n \times o}$ are the observer gains which ensure that the estimates $\hat{x}_i$ remain bounded. Applying the same idea as in (3.3), the labels are generated by

$$\hat{\sigma}(y(t), u(t)) = \arg\min_{i \in \mathcal{I}} \|y(t) - \hat{y}_i(t)\|. \tag{3.5}$$

Figure 3.1 depicts the unsatisfactory quality of the labels obtained by (3.5) for a continuous-time PWA system in state-space form switching from subsystem 1 to subsystem 2. A considerable delay can be seen in the estimated switching signal $\hat{\sigma}$. This delay is caused by the limited convergence rate of the observers (3.4). Overall the labels obtained with (3.5) do not coincide well with the true switching signal right after a switch. Therefore, any switching hyperplane reconstructed with such erroneous labels is prone to be defective. Motivated by this limitation in the continuous-time case, we propose a novel approach to identify the polyhedral regions more precisely.

Another limitation of the state of the art in switching hyperplane estimation is the limitation to fixed regions. Fixed regions are required in order to collect a long time series of consistent data-label pairs that can be fed into the SVMs in batch mode. Time-varying switching hyperplanes cause data-label pairs that are inconsistent for long time series. In such cases, linear separation with standard SVMs yields unsatisfactory results. As pointed out in Section 2.3.4, the field of incremental SVMs has not yet reached a level of maturity to provide a solution to this problem. Fortunately, the solution which we propose for Problem 3.1 permits a recursive implementation and therefore allows to track time-varying switching hyperplanes. The following problem formulation describes this more challenging task.

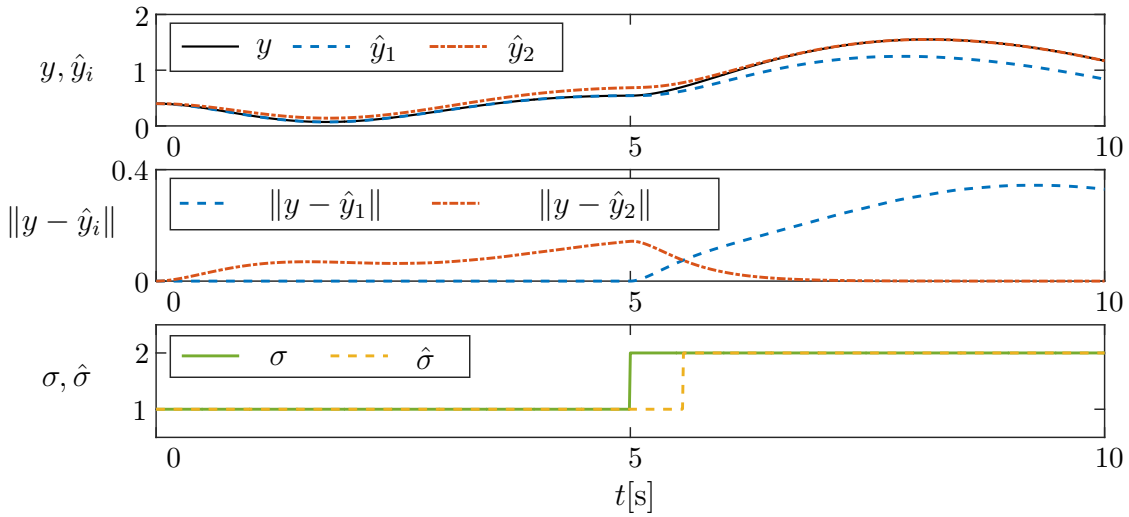


Fig. 3.1: Visualization of the delay in the estimated switching signal due to limited convergence rates of continuous observers.

Problem 3.2 Given the subsystem parameters A_i, B_i, f_i and C_i. Estimate and track the time-varying polyhedral regions Ω_i characterized by $\mathcal{H}_i(t)$, $i \in \mathcal{I}$ based on a stream of input-output data.

The solution to Problem 3.1 and Problem 3.2 presented in this chapter consists of three steps. In the first step, a hybrid observer estimates the switching signal online. Due to the inherently reactive nature of observers to estimation errors, the estimated switching signal obtained in the first step is delayed compared with the true switching signal. However, the applied observer allows to record parts of the output trajectories before and after the switch, which can subsequently help to improve the estimated switching time. In the second step, we define and solve an optimization problem based on the recorded trajectory, which yields more precise estimate of the switching time and state. After collecting several switching states, the third step fits a hyperplane to these states. The following three sections describe this three steps in greater detail.

3.2 Hybrid Switching State Observer

The first step towards estimating the polyhedral regions of PWA systems is the reconstruction of the unknown switching signal $\sigma(t)$ as well as the state $x(t)$ based on measurements of the noisy output $y(t)$. This alone constitutes a challenging task and has attracted a considerable amount of research on observability of hybrid and switched systems (e.g. [48, 202, 221, 238]) and put forth various switching signal estimators and observers for hybrid and switched systems.

In [238], the switching signal is assumed to be known and the challenge is to estimate the states of a switched system with state jumps even if some of the subsystems are not observable. In [90] and [91], the inputs and outputs of the switched system are numerically differentiated to generate residuals. Only the residual which belongs to the active subsystem is zero. Another approach to estimate the switching times of an autonomous, switched

system online is presented in [243] and [244] and based on algebraic tools and distribution theory, which is needed to handle derivatives of non smooth dynamics. The observer for hybrid systems presented in [22] consists, as most hybrid observers, of two parts. The first part estimates the current mode of the system and the second part estimates the continuous states of the system. In [270], average dwell time constraints are derived for which a multiple-model observer achieves exponential state tracking for both synchronous and asynchronous switching of plant and observer. A comparative study of various mode detection algorithms for discrete-time switched affine systems is given in [106]. The observer design proposed in [103] is based on a geometric concept of invariant subspaces. Finally, [239] presents an observer for switched linear systems with state jumps.

For the considered task, the observer for switched linear systems presented in [146] seems to be most suitable, because it allows for an intuitive recording of input and output data before and after each switch. The estimation algorithm in [146] is based on the idea of a minimum distance criterion [24]. The *hybrid observer* [146] estimates switches of the PWA system in two stages. Note that the overall hyperplane estimation consists of three steps, whereas the online switching signal estimator has two stages. The first stage of the observer applies at the beginning or whenever a switch occurs. This stage is called *Estimation-stage* and has a minimum duration of T_δ, in which the current mode $\sigma(t)$ of the PWA system is correctly determined and the state is estimated up to a certain noise level. Once the current mode is correctly estimated, the observer switches to its second stage. The *Detection-stage* aims to detect a switch within a maximum delay T_Δ after it occurs. Once a switch is detected, the observer carries on with the Estimation-stage and so forth.

The Estimation-stage and Detection-stage of the online switching signal estimator are now revised in sufficient detail for implementation. For the rigorous convergence proofs of the algorithm, the reader is referred to [146]. Three assumptions arise for the algorithm to operate in the anticipated way. For a better readability, this chapter only discusses their practical implications. The detailed assumptions are listed in Appendix C. The distinguishability of any two subsystems is guaranteed by Assumption C.1. Assumption C.2 gives constraints for the variables T_δ, T_Δ, $\mathcal{X}_\triangle$ and $\mathcal{S}_\triangle$ and highlights their relationship to the input u and measurement noise v_y. Furthermore, these variables depend on a quantity that expresses the maximum difference/distance between two subsystem parametrizations. A minimum dwell time of length $T_\delta + T_\Delta$ for the PWA system is demanded by Assumption C.3.

3.2.1 Active Mode Estimation

The first stage of the observer is active when the observer starts operating and whenever a switch has been detected. Assume that the j-th switch has occurred at time instance t_j but has only been detected at the *estimated switching time* $\hat{t}_j > t_j$. At the initialization of the observer let $\hat{t}_0 = 0$. While the PWA system (3.1) evolves over the time-interval $[\hat{t}_j, \hat{t}_j + T_\delta]$, the following differential equations with initial conditions $\breve{x}_i(\hat{t}_j) = 0$ and $z_i(\hat{t}_j) = 0$ are

integrated in real-time for all subsystems $i \in \mathcal{I}$:

$$\dot{\check{x}}_i = A_i \check{x}_i + B_i u + f_i \tag{3.6}$$
$$\dot{z}_i = -A_i^\top z_i + C_i^\top (y - C_i \check{x}_i). \tag{3.7}$$

During the same time interval $t \in [\hat{t}_j, \hat{t}_j + T_\delta]$, the control input $u(t)$ and measured $y(t)$ are recorded. Note that the check-symbol ($\check{x}$) characterizes an *open-loop observer* in contrast to the hat-symbol ($\hat{x}$), which represents the standard *Luenberg observer* (see (3.4)).

Based on the states $z_i(\hat{t}_j + T_\delta)$, obtained from (3.7), better *initial conditions* X_i^* for $\check{x}_i$ at time instance $\hat{t}_j$ are calculated as:

$$X_i^* = U_i^{-1}(T_\delta) e^{A_i^\top T_\delta} z_i(\hat{t}_j + T_\delta), \tag{3.8}$$

where

$$U_i(T_\delta) := \int_0^{T_\delta} \phi_i^\top(\tau)\phi_i(\tau) d\tau$$

is the observability Gramian for mode i and

$$\phi_i(t) := C_i e^{A_i t}.$$

With the improved initial conditions $\check{x}_i(\hat{t}_j) = X_i^*$, and $s_i(\hat{t}_j) = 0$, and the stored values of u and y, the following differential equations are once more integrate over the time interval $[\hat{t}_j, \hat{t}_j + T_\delta]$:

$$\dot{\check{x}}_i = A_i \check{x}_i + B_i u + f_i \tag{3.9}$$
$$\dot{s}_i = \|y - C_i \check{x}_i\|^2. \tag{3.10}$$

For convenience, the integration of (3.9) and (3.10) is assumed to happen instantaneously at $\hat{t}_j + T_\delta$. Equation (3.9) represents an open-loop observer without feedback correction from the actual system output. The integral over (3.10) corresponds to the sum of the difference between the measured outputs y and the estimated outputs of the open-loop observer (3.9). For the initial conditions X_i^*, the costs $J_i(X_i^*) := s_i(\hat{t}_j + T_\delta)$ are evaluated at time instance $\hat{t}_j + T_\delta$. The mode with minimum cost is found to be the currently active one:

$$\hat{\sigma}_j := \hat{\sigma}(\hat{t}_j + T_\delta) = \arg\min_{i \in \mathcal{I}} J_i(X_i^*). \tag{3.11}$$

Then, the state of the Luenberg observer is reset to the corresponding best estimate

$$\hat{x}(\hat{t}_j) = X_{\hat{\sigma}_j}^*. \tag{3.12}$$

If Assumption C.1 and Assumption C.2 hold, one can show that the estimated mode $\hat{\sigma}_j$ is correct and that the state estimate $\hat{x}(\hat{t}_j)$ is equal to the true state $x(\hat{t}_j)$ up to certain noise levels.

Theorem 3.3 (Active Mode Estimation) "Under Assumptions C.1 and C.2, if there is no switching between $\hat{t}_j$ and $\hat{t}_j + T_\delta$, then the estimates by (3.11) and (3.12) have the

property that

$$\hat{\sigma}(t) = \sigma(t) \tag{3.13}$$

$$\hat{x}(\hat{t}_j) = x(\hat{t}_j) + U^{-1}_{\hat{\sigma}_j}(T_\delta) \int_{\hat{t}_j}^{\hat{t}_j+T_\delta} \phi^\top_{\hat{\sigma}_j}(\tau - \hat{t}_j) v_y(\tau) d\tau \tag{3.14}$$

for all $t \geq \hat{t}_j + T_\delta$ until the next mode switch occurs." [146, Th. 1]

The hybrid observer then switches to the second stage: Detection.

3.2.2 Mode Switch Detection

The goal in this stage is to detect when the subsystem $\hat{\sigma}_j$ does not apply anymore. For this purpose, the state of the system is estimated with two observers, both of which are parametrized for the $\hat{\sigma}_j$-th subsystem, which was found to be the active one during the Estimation-stage. One of the observers is an open-loop observer ($\check{x}$), while the other one is a standard Luenberg observer ($\hat{x}$).

First, with the recorded data of u and y, the estimation catches up from $\hat{t}_j$ to the current time $\hat{t}_j + T_\delta$ by integrating

$$\dot{\check{x}} = A_{\hat{\sigma}_j}\check{x} + B_{\hat{\sigma}_j}u + f_{\hat{\sigma}_j} \tag{3.15}$$

$$\dot{\hat{x}} = (A_{\hat{\sigma}_j} - L_{\hat{\sigma}_j}C_{\hat{\sigma}_j})\hat{x} + B_{\hat{\sigma}_j}u + f_{\hat{\sigma}_j} + L_{\hat{\sigma}_j}y \tag{3.16}$$

$$\dot{s} = \|y - C_{\hat{\sigma}_j}\check{x}\|^2 \tag{3.17}$$

with initial conditions $\check{x}(\hat{t}_j) = \hat{x}(\hat{t}_j) = X^*_{\hat{\sigma}_j}$ (as in (3.8)) and $s(\hat{t}_j) = 0$. After catching up to $t = \hat{t}_j + T_\delta$, integration of (3.15) - (3.17) proceeds in real-time in time frames of length T_Δ. After each T_Δ-frame, the open-loop observer state and the starting time are reset according to the rule

$$\text{if } t - t_{\text{up}} > T_\Delta, \quad \text{then } \begin{cases} \check{x}(t) & \Leftarrow \hat{x}(t) \\ t_{\text{up}} & \Leftarrow t \end{cases} \tag{3.18}$$

The next switch is detected when one of the following two conditions is fulfilled:

$$\|\check{x}(t) - \hat{x}(t)\| > \mathcal{X}_\triangle, \qquad s(t) - s(t - T_\Delta) > \mathcal{S}_\triangle, \tag{3.19}$$

where the thresholds $\mathcal{X}_\Delta$ and $\mathcal{S}_\Delta$ are given by Assumption C.2 in appendix C. The delay with which a switch is detected is characterized by the following theorem.

Theorem 3.4 (Mode Switch Detection) If Assumptions C.1, C.2 and C.3 hold, the detection algorithm is guaranteed to find an estimate $\hat{t}_{j+1}$ of the next switch at t_{j+1} within the interval $[t_{j+1}, t_{j+1} + T_\Delta]$. [146, Th 2]

Proof: The proof of Theorem 3.4 is obtained by combining Lemma 1 and 2 in [146]. □

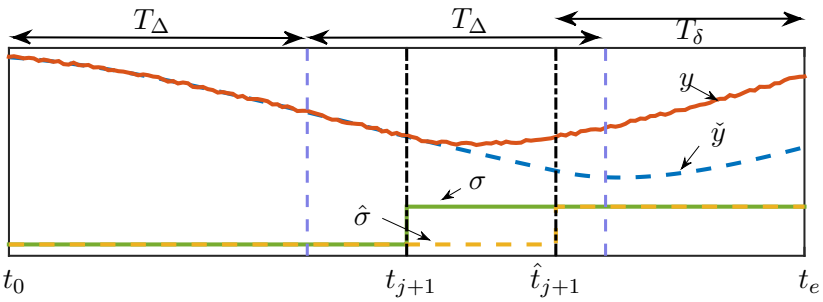


Fig. 3.2: Sketch of the recorded time frame containing the actual switching time t_{j+1} as well as the time instance $\hat{t}_{j+1}$ when the switch was detected.

After detecting a switch, the observer returns to the estimation stage. In the worst case, the estimated switching time $\hat{t}_{j+1}$ is delayed by T_Δ. In order to eliminate this delay by further analysis, we propose to record u and y over each time frame T_Δ as well as the value of $\hat{x}$ at the beginning of each interval. As the memory of control systems is usually limited and the PWA system might not switch its mode for a long time, it suffices to store only the last two recorded intervals. Together with the recorded values during the estimation stage, three connected parts of the system trajectory should be collected: the first element of length T_Δ before the switch, the second element also of length T_Δ containing the estimated switching time $\hat{t}_{j+1}$ and the third element of length T_δ after the estimated switching time. Figure 3.2 sketches the recorded output trajectory and highlights the elements of length T_Δ and T_δ. These recordings are used in the next step of the proposed hyperplane estimation algorithm to estimate the switching state more precisely.

3.3 Switching State Optimization

With the recorded trajectories of the switching state observer, more precise estimates of the true switching times and states can be obtained as the solution of an optimization problem. There exist various approaches for optimal control of switching systems (see [8, 72, 79, 222, 268] and the references therein). Algorithms for optimal switching control usually aim to determine the optimal sequence and time instances in which a given set of system dynamics needs to be activated in order to minimize a given cost function. The optimization problem considered here is easier as the switching sequence is already known and only a single switching time must be optimized.

From the switching state observer, recorded measurements of the inputs u and outputs y of the PWA system are available over a time interval $[t_0, t_e]$. An example for such an interval is given in Fig. 3.2. Furthermore, the switching state observer provides the information that the system switches its mode during this period from $\hat{\sigma}_j$ to $\hat{\sigma}_{j+1}$. Also, the estimated state $\hat{x}(t_0)$ is known from the observer.

Assuming a single switch at time instance τ, the dynamics of the switched system are

$$\dot{\check{x}}(t) = \begin{cases} A_{\hat{\sigma}_j}\check{x}(t) + B_{\hat{\sigma}_j}u(t) + f_{\hat{\sigma}_j} & \text{, if } t \leq \tau \\ A_{\hat{\sigma}_{j+1}}\check{x}(t) + B_{\hat{\sigma}_{j+1}}u(t) + f_{\hat{\sigma}_{j+1}} & \text{, if } t > \tau \end{cases} \tag{3.20}$$

with $\check{x}(t_0) = \hat{x}(t_0)$. Likewise, the output is given by

$$\check{y}(t) = \begin{cases} C_{\hat{\sigma}_j}\check{x}(t) & \text{, if } t \leq \tau \\ C_{\hat{\sigma}_{j+1}}\check{x}(t) & \text{, if } t > \tau \end{cases} \tag{3.21}$$

An integral over the squared error between the estimated output trajectory $\check{y}$ and the recorded output trajectory y serves as the cost function. This cost function only depends on the variable switching time τ and must be minimized to find the *optimal switching time* $\bar{t}_{j+1}$:

$$\bar{t}_{j+1} := \arg \min_{\substack{\tau \\ \text{s.t. } t_0 \leq \tau \leq t_e}} \int_{t_0}^{t_e} \|y(t) - \check{y}(t)\|^2 dt. \tag{3.22}$$

No real-time constraints arise for this constrained optimization problem as the estimation and detection of the hybrid switching state observer does not rely on the solution. Standard optimization techniques such as golden section search and parabolic interpolation are suitable tools to find the solution to the optimization problem [38, 92, 195]. The optimal switching time $\bar{t}_{j+1}$ is used to calculate a more precise estimate of the switching state $\hat{x}(\bar{t}_{j+1})$ and input $u(\bar{t}_{j+1})$, which are used in the subsequent step to estimate the switching hyperplanes.

3.4 Hyperplane Fitting with Total Least Squares

Performing the previous two steps for multiple switches generates sets of labeled switching states. The label uniquely specifies the switching hyperplane that the state belongs to. Assume for instance that a switch from region k to region l was detected in the first step of the algorithm. Hence, it can be concluded that there must exist a switching hyperplane separating Ω_k from Ω_l and the estimated switching state lies on this hyperplane. Collecting multiple switching states for switches from region Ω_k to Ω_l and vice versa yields a set of switching states that lie on the same hyperplane. In the final step, it is left to fit a hyperplane to this set of points.

Recall the definition of the hyperplane (3.2) in terms of the vector h. Given a set of m switching states, fitting a hyperplane to this set can be interpreted as finding the solution to an overdetermined system of equations:

$$\mathcal{A}X \approx \mathcal{B}, \tag{3.23}$$

where the matrix $\mathcal{A} \in \mathbb{R}^{m \times (n+p)}$ and the vector $\mathcal{B} \in \mathbb{R}^m$ consist of given data and $X \in \mathbb{R}^{n+p}$ contains the unknown hyperplane parameters. By setting $\gamma = 1$ in (3.2) it follows that $X = [\alpha_1, \ldots, \alpha_n, \beta_1, \ldots, \beta_p]^\top$. Each row of $\mathcal{A}$ corresponds to one of the obtained switching states (e.g. $[\hat{x}^\top(\bar{t}_j), u^\top(\bar{t}_j)]$) and $\mathcal{B}$ is a column vector with all elements equal to -1.

Typically, there are more data points than unknowns, i.e. $m > n + p$. Furthermore, it must be assumed that both $\mathcal{A}$ and $\mathcal{B}$ contain uncertainties as they are the result of the first two estimation steps in Section 3.2 and Section 3.3. Therefore, an exact solution to the overdetermined system of equations does not exist (hence the approximation symbol $\approx$ in (3.23)). With *Total Least Squares (TLS)* it is possible to find an approximate solution

to the system of equations (3.23) by correcting it with slack variables $\mathcal{E}_\mathcal{A}$ and $\mathcal{E}_\mathcal{B}$:

$$(\mathcal{A} + \mathcal{E}_\mathcal{A})X = \mathcal{B} + \mathcal{E}_\mathcal{B}.$$

This leads to the optimization problem

$$\min_{\substack{X,\mathcal{E}_\mathcal{A},\mathcal{E}_\mathcal{B} \\ \text{s.t. } (\mathcal{A}+\mathcal{E}_\mathcal{A})X=\mathcal{B}+\mathcal{E}_\mathcal{B}}} \left\| [\mathcal{E}_\mathcal{A}\ \mathcal{E}_\mathcal{B}] \right\|_F, \tag{3.24}$$

where $\|\cdot\|_F$ is the Frobenius norm. An overview of the history of Total Least Squares as well as some recent advances in the field is given in [160].

The algorithm proposed in [160, Algorithm 1] to solve the Total Least Squares problem (3.24) begins with the singular value decomposition $[\mathcal{A}\ \ \mathcal{B}] = U\Sigma V^\top$, which yields the block matrix

$$V = \begin{bmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{bmatrix}$$

with $V_{11} \in \mathbb{R}^{(n+p)\times(n+p)}$, $V_{12} \in \mathbb{R}^{(n+p)\times 1}$, $V_{21} \in \mathbb{R}^{1\times(n+p)}$ and $V_{22} \in \mathbb{R}$. A solution to (3.24) exists if V_{22} is nonzero and the optimal parameter vector $\hat{X}$ is then given by

$$\hat{X} = -V_{12}V_{22}^{-1}.$$

Finally, the hyperplane obtained by Total Least Squares and the latest m switching states is $\hat{h} = [\hat{X}^\top, 1]$.

Since any two switches of the PWA system are assumed to be separated by a dwell time, it can usually be assume that the time between two switches suffices to recompute the singular value decomposition. If the dwell time is too short or if the computational cost is too large, one could resort to a recursive implementation of Total Least Squares [62, 84, 123, 207].

3.5 Simulation Studies

The proposed hyperplane estimation algorithm is now tested in a series of simulations. First, a set of time-invariant hyperplanes are estimated. Then, the hyperplane parameters undertake a sudden jump in order to validate the tracking properties of the algorithm. A simplistic example is chosen to illustrate the algorithm in detail. Note that the algorithm was also successfully tested with more complex systems. Increasing complexity, however, might make it impossible to fulfill the required assumptions. Nevertheless, a more complex state-space partition will be considered in Section 6.4.2, where a PWA model of a hybrid two-tank will be estimated.

3.5.1 Estimating Time-Invariant Hyperplanes

Both examples consider the PWA system discussed in [273] and [276] which consists of three subsystems. The subsystems differ in the matrices A_i and f_i:

$$A_1 = \begin{bmatrix} 0 & 1 \\ -2 & -1 \end{bmatrix} \qquad A_2 = \begin{bmatrix} 0 & 1 \\ -2.5 & -1 \end{bmatrix} \qquad A_3 = \begin{bmatrix} 0 & 1 \\ -1.5 & -1 \end{bmatrix}$$
$$f_1 = \begin{bmatrix} 0 & 0.2 \end{bmatrix}^\top \qquad f_2 = \begin{bmatrix} 0 & 0.4 \end{bmatrix}^\top \qquad f_3 = \begin{bmatrix} 0 & -0.3 \end{bmatrix}^\top$$

while $B = [0 \;\; 1.5]^\top$ and $C = [1 \;\; 0]$ for all subsystems. Two hyperplanes partition the state-space. The hyperplane h_{12} separates subsystem 1 and 2, while h_{13} separates subsystem 1 and 3. For the first experiment, the hyperplanes are constant with

$$h_{12,\mathrm{I}} = \begin{bmatrix} -1 & 0 & 0 & 2 \end{bmatrix}$$
$$h_{13,\mathrm{I}} = \begin{bmatrix} 1 & 0 & 0 & 2 \end{bmatrix}.$$

The corresponding regions are

$$\Omega_1 = \left\{ [x^\top, u]^\top \mid -2 < x_1 \leq 2 \right\},$$
$$\Omega_2 = \left\{ [x^\top, u]^\top \mid x_1 > 2 \right\},$$
$$\Omega_3 = \left\{ [x^\top, u]^\top \mid x_1 \leq -2 \right\}.$$

The PWA system is excited by a sinusoidal input signal with varying amplitude:

$$u(t) = (9 + 3\cos(0.015t))\sin(0.1t).$$

White noise with zero mean and $v_{y,\max} = 0.1$ is added to the output $y(t)$. For the online estimation of the switching signal $\hat{\sigma}$ according to Section 3.2, the observer gains L_i are chosen such that the poles of the Luenberg observers $A_i - L_i C_i$ lie at -3 and -3.5. This gives sufficiently fast convergence without amplifying the measurement noise. The remaining design parameters are $T_\delta = 0.8$, $T_\Delta = 0.8$, $\mathcal{S}_\Delta = 0.003$ and $\mathcal{X}_\Delta = 0.04$.

Figure 3.3 visualizes the estimation process for a time period of 250 s. The dash-dotted trajectory represents the true state x of the PWA system, of which only the first state x_1 is available for measurement. The solid, colored trajectory shows the estimated state $\hat{x}$ which is obtained with the hybrid state observer. Besides the estimated state $\hat{x}$, the hybrid state observer also delivers an estimate of the switching signal $\hat{\sigma}(t)$, which is included in the figure through color coding: the red part of the trajectory in the middle is estimated to belong to subsystem 1 ($\hat{\sigma} = 1$); the green part of the trajectory on the right consists of states for which $\hat{\sigma} = 2$; and the blue trajectory on the left is linked to subsystem 3 ($\hat{\sigma} = 3$). Note that the state estimation is precise as long as the estimated switching signal is correct. However, whenever the PWA system switches at $x_1 = \pm 2$, the hybrid observer requires an additional time interval of maximum length $T_\delta + T_\Delta$ to detect the switch and estimate the current mode. During this transient phase, the estimated trajectory deviates from the true trajectory. Such deviations cannot be avoided in hybrid observers with switching signal reconstruction. As will be clear shortly, this deviation does not affect the obtained

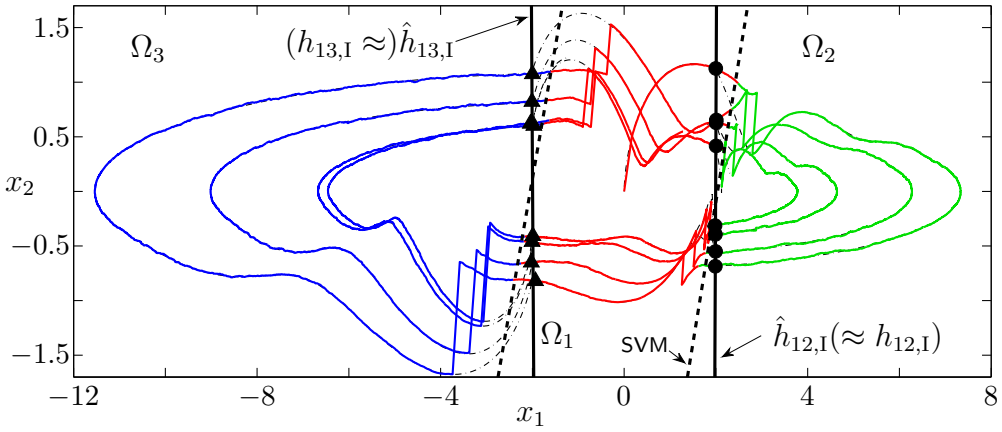


Fig. 3.3: State-space trajectory of the real system (dash-dotted) and the online-switching signal observer (colored). Circles and triangles mark the estimated switches $1 \leftrightarrow 2$ and $1 \leftrightarrow 3$, respectively.

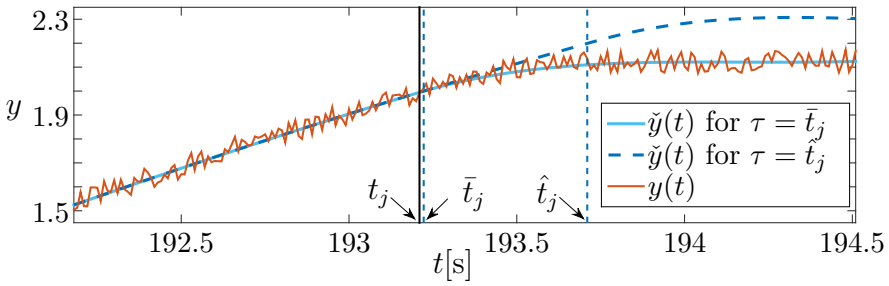


Fig. 3.4: Improving the estimated switching time from $\hat{t}_j$ to $\bar{t}_j$ by minimizing the squared error between measured output $y(t)$ and estimated output $\check{y}(t)$.

hyperplane estimates.

In Section 3.1 above, it was argued that the use of observers leads to an inherent delay in the estimated switching signal. This delay can be seen in Fig. 3.3 as a delayed change in color of the estimated state-space trajectory at the switching hyperplanes. In order to stress the argument that the state of the art is inapplicable for continuous-time PWA systems, a linear SVM is trained with the colored/labeled trajectory elements. The hyperplanes obtained with this state-of-the-art approach (compare Section 2.3) are visualized as dashed lines in Fig. 3.3. Note that the SVM hyperplanes deviate from the true switching hyperplanes by a few degrees, which is clearly unsatisfactory.

As proposed in Section 3.3, solving the simple optimization problem in (3.22) yields a more precise estimate of the switching time. For this simulation, the Matlab function `fminbnd` solved the optimization problem within an average of 443.6 ms on a quad core processor (i5-2500K, 3.30Ghz). Figure 3.4 shows how the optimization step improves the estimated switching time from $\hat{t}_j$ (obtained online) to $\bar{t}_j$. The figure compares the open-loop estimates $\check{y}$ generated according to (3.20) and (3.21) for the initial estimate $\hat{t}_j$ of the switching time and the optimized $\bar{t}_j$. The optimized switching time brings the estimated output $\check{y}$ closer to the noisy measurement y. Furthermore, the optimized estimate $\bar{t}_j$ is now very close to the actual switching time t_j.

The optimization was carried out for every detected switch. The resulting optimal

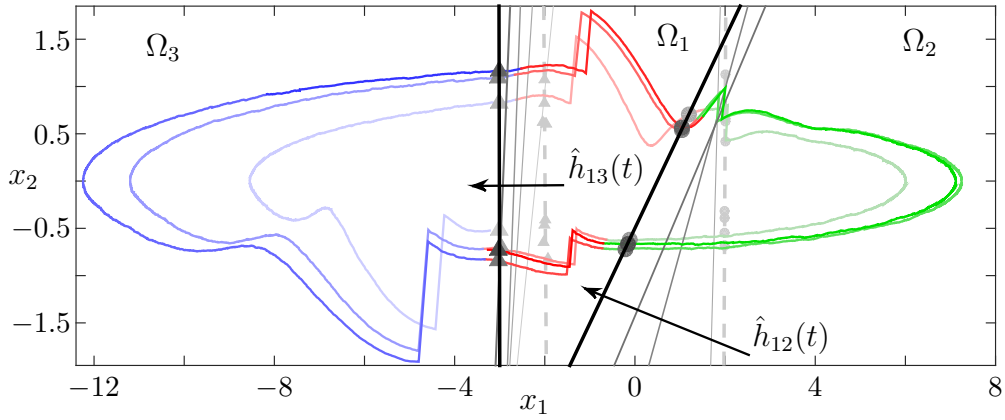


Fig. 3.5: Tracking time-varying hyperplanes with the proposed algorithm.

switching states are highlighted in Fig. 3.3 by circles for switches between subsystem 1 and 2, and by triangles for switches between subsystem 1 and 3. In the final step, presented in Section 3.4, Total Least Squares fits estimated hyperplanes $\hat{h}_{12}$ and $\hat{h}_{13}$ to the optimal switching states. The obtained estimates

$$\hat{h}_{12,\mathrm{I}} = \begin{bmatrix} -1.0024 & 0.0107 & 0.0000 & 2.0000 \end{bmatrix}$$
$$\hat{h}_{13,\mathrm{I}} = \begin{bmatrix} 1.0023 & 0.0149 & 0.0000 & 2.0000 \end{bmatrix}$$

are very close to the true parameters $h_{12,\mathrm{I}}$ and $h_{13,\mathrm{I}}$. Compared with the SVM-approach (dotted lines in Fig. 3.3), the proposed approach (solid lines) delivers a much better approximation of the true hyperplanes.

3.5.2 Tracking Time-Varying Hyperplanes

In order to validate the tracking ability of the proposed algorithm, a change in hyperplane parameters is introduced after the first experiment. At $t = 250\,\mathrm{s}$, the hyperplanes change from $h_{12,\mathrm{I}}$ and $h_{13,\mathrm{I}}$ to $h_{12,\mathrm{II}}$ and $h_{13,\mathrm{II}}$ with

$$h_{12,\mathrm{II}} = \begin{bmatrix} -0.2 & 0.2 & 0 & 0.1 \end{bmatrix}$$
$$h_{13,\mathrm{II}} = \begin{bmatrix} 1 & 0 & 0 & 3 \end{bmatrix}.$$

This changes resemble a rotation of h_{12} and a translation of h_{13} and could be caused by aging or wear in a practical system.

The proposed algorithm gains tracking abilities by calculating the Total Least Squares solution only for the last $m = 6$ switching states per hyperplane. Figure 3.5 shows how the algorithm tracks the changing hyperplanes. The estimated state-space trajectory from $t = 250\,\mathrm{s}$ to $t = 500\,\mathrm{s}$ is given by the colored, solid trajectory. The intensity of the trajectory increases with time. The same applies for the estimated switching states and hyperplanes. The switching states for $t < 250\,\mathrm{s}$ (found in the first experiment) are given by gray circles and triangles, while the corresponding hyperplanes at $t = 250\,\mathrm{s}$ are shown as dashed, gray lines. The intensity of the estimated switching states then shifts from gray to

black as time goes on. The figure also shows the time varying estimates $\hat{h}_{12}(t)$ and $\hat{h}_{13}(t)$ from $t = 250\,\mathrm{s}$ to $t = 500\,\mathrm{s}$. Arrows indicate the movement of the estimates. Intermediate estimates are shown by lines of varying width and intensity. The thinner and lighter a line, the older is the corresponding estimate of the switching hyperplane.

One of the intermediate estimates is

$$\hat{h}_{12,\mathrm{II}}(t = 300) = \begin{bmatrix} -0.0584 & 0.0379 & 0.0000 & 0.1000 \end{bmatrix}$$
$$\hat{h}_{13,\mathrm{II}}(t = 300) = \begin{bmatrix} 1.1903 & -0.1570 & 0.0000 & 3.0000 \end{bmatrix}.$$

The final estimates are again very close to the true parameters $h_{12,\mathrm{II}}$ and $h_{13,\mathrm{II}}$:

$$\hat{h}_{12,\mathrm{II}}(t = 500) = \begin{bmatrix} -0.2020 & 0.2008 & 0.0000 & 0.1000 \end{bmatrix}$$
$$\hat{h}_{13,\mathrm{II}}(t = 500) = \begin{bmatrix} 0.9974 & 0.0045 & 0.0000 & 3.0000 \end{bmatrix}.$$

The simulation results show that tracking of time-varying switching hyperplanes is possible with the proposed algorithm. Abrupt parameter changes, however, require a certain transient phase in oder to converge to the true parameters. The duration of this transient phase is mostly determined by two factors: the frequency with which the PWA systems transitions at a hyperplane; and the number m of past switches that are considered for TLS-based hyperplane fitting.

3.6 Summary

This Chapter is motivated by the observation that existing algorithms for estimating switching hyperplanes in PWA systems deliver shifted estimates if continuous-time state-space systems are considered. The shift is due to observer delays required for switching signal reconstruction. This chapter presents a novel, recursive estimation algorithm for switching hyperplanes in continuous-time PWA systems in state-space form. The proposed algorithm consists of three steps. First, a hybrid state observer estimates the switching signal and the full state of the PWA system. The hybrid state observer alternates between detecting the next mode switch and reconstructing the current mode, once a switch has been detected. The second step in the overall algorithm takes place after every detected switch and consists of solving a simple optimization problem. By solving the optimization problem, the accuracy of the estimated switching times, which are still affected by bounded estimation delays, is improved. Finally, the switching hyperplanes are fitted to the obtained switching states by Total Least Squares. In comparison to the standard SVM approach, the proposed algorithm delivers much more accurate estimates by reliably removing estimation delays. Furthermore, the algorithm is easily implemented in a recursive way, which enables tracking parametric changes of time-varying switching hyperplanes, a feature not available in the SVM approach.

Regarding generalization, the proposed algorithm enables learning of the underlying switching mechanism in hybrid systems much better than previous approaches. This is due to the fact that very precise estimates of both switching times and switching states are produced in addition to a simple labeling of data points carried out in the state of

the art. By learning the underlying switching mechanism in hybrid systems, the tracking performance of observers may be improved by predicting switches rather than reacting to switches. Also, tracking switching boundaries can be an enabling factor for adaptive control of PWA systems, which is so far restricted to fixed and known switching boundaries (e.g. [70]).

Besides the highlighted benefits and possibilities for generalization, the algorithm also faces some limitations. First, the hybrid state observer, which forms an essential part of the algorithm, is currently limited to continuous-time systems in state-space form. It would hence be an interesting step for future work to investigate whether a discrete-time counterpart exists, which would enlarge the class of applicable systems. Next, detecting switches and reconstructing the switching sequence poses dwell-time constraints on the system. If these dwell-time constraints cannot be ensured, the switching hyperplanes cannot be reconstructed reliably. This constitutes a common constraint in identification algorithms for switched state-space systems with partial state measurement. Finally, another challenge in applying the proposed algorithm is finding suitable design parameters. While Appendix C lists explicit constraints, there exists no guideline for finding suitable parameters within the constraints. Defining guidelines to replace the currently heuristic search is therefore another topic for future work.

4 Parameter Identifiers for PWA Subsystem Identification

The state-of-the-art overview in Section 2 and Appendix B revealed that most identification algorithms are concerned with discrete-time PWA systems in input-output form. A provably converging, recursive algorithm for continuous-time PWA systems in state-space form is still lacking. The adaptive control community on the other side has derived various algorithms for recursive parameter estimation in continuous-time state-space systems [6, 114, 175, 240]. Ideas from the adaptive control community, however, find surprisingly little application in the identification of PWA systems. One exception is the method presented in [67], which allows to identify bimodal piecewise linear (PWL) state-space systems based on the gain evolution of an adaptive controller [66]. The main drawback of this approach is the limitation to two subsystems.

Recognizing the potential of applying established estimation algorithms from the adaptive control community to identify PWA systems was the main motivation behind our previous works [273, 276], which extended adaptive *parameter identifiers* [114] from linear systems

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

with unknown system matrices A, B to PWL and PWA systems

$$\dot{x}(t) = A_{\sigma(t)}x(t) + B_{\sigma(t)}u(t) + f_{\sigma(t)},$$

with unknown subsystem matrices A_i, B_i, f_i.

Note that all adaptive algorithms for piecewise linear systems [66, 67, 273, 276] share the assumption of known switching hyperplanes. Together with the assumption of full state measurements, this implies knowledge of the switching signal $\sigma(t)$. The problem of recursive subsystem identification in PWA systems thus reads as follows.

Problem 4.1 Given the state-space partitions Ω_i, find update laws for the estimated parameters $\hat{A}_i$, $\hat{B}_i$ and $\hat{f}_i$, that are based on the measured state x and input u and provably drive the estimates to the true system parameters A_i, B_i and f_i.

This chapter revises the solution to Problem 4.1 proposed in [273, 276]. Additional insights into the proofs are presented and a more detailed numerical testing is performed. The remainder of this chapter is organized as follows. Section 4.1 begins with a revision of parameter identifiers for linear state-space systems. Afterwards, the two extensions to PWL and PWA systems are discussed in Section 4.2. Numerical testing in three scenarios, including a hybrid robot-contact problem, is given in Section 4.3. Conclusions follow in Section 4.4.

4.1 Linear Time Invariant Systems

First, consider the following linear time invariant state-space system

$$\dot{x} = Ax + Bu, \tag{4.1}$$

with the unknown system parameters $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times p}$. For parameter identifiers, one introduces the time varying estimates $\hat{A} \in \mathbb{R}^{n\times n}$ and $\hat{B} \in \mathbb{R}^{n\times p}$. The goal is to design suitable update laws that can be shown to drive these estimates provably to the true parameters. The update laws tune the estimates based on the so called prediction error between the predicted system output and the actual system output. Therefore, the state of the true system (4.1) is predicted based on the current parameter estimates $\hat{A}(t)$ and $\hat{B}(t)$ by

$$\dot{\hat{x}} = A_{\mathrm{m}}\hat{x} + \left(\hat{A} - A_{\mathrm{m}}\right)x + \hat{B}u, \tag{4.2}$$

where $\hat{x} \in \mathbb{R}^n$ is the *predicted state* of the system. The stable matrix $A_{\mathrm{m}} \in \mathbb{R}^{n\times n}$ is chosen by the designer and ensures that the predicted state remains bounded. Since A_{m} is stable, one can always find a symmetric, positive definite matrix $P \in \mathbb{R}^{n\times n}$ for which $A_{\mathrm{m}}^\top P + PA_{\mathrm{m}} = -Q_{\tilde{x}}$, with a positive definite matrix $Q_{\tilde{x}} \in \mathbb{R}^{n\times n}$.

In the application of parameter identifiers, one defines the *prediction error* as $\tilde{x} = \hat{x} - x$ and uses the following standard *update laws* for the parameter estimates

$$\dot{\hat{A}} = -\Gamma_1 P\tilde{x}x^\top, \qquad \dot{\hat{B}} = -\Gamma_2 P\tilde{x}u^\top, \tag{4.3}$$

where $\Gamma_1, \Gamma_2 > 0$ are positive scaling factors, which are frequently referred to as adaptive learning gains. With the preliminaries on adaptive systems (e.g. the signal property of being sufficiently rich) described in Section A.2, the following theorem describes parameter convergence obtained with parameter identifiers.

Theorem 4.2 (Identifiers for linear systems) Let (A, B) be a controllable pair. If the elements $u_k, k = 1, 2, \ldots p$ of u are sufficiently rich of order $n+1$ with distinct frequencies, then the estimates $\hat{A}(t), \hat{B}(t)$ generated by (4.2) and (4.3) converge exponentially fast to the unknown plant parameters A, B, respectively. [114, Th. 5.2.3])

Proof: As will be shown shortly, Theorem 4.2 is a special case of the switched system case in the later Theorem 4.3. We therefore omit the proof at this point and refer to Theorem 4.3 or the standard textbook [114, Theorem 5.2.3] □

4.2 Parameter Identifiers for Switched Systems

Next, the standard parameter identifiers, revised in the previous section, are extended to the PWL and PWA systems introduced in Section 2.1. First, consider a PWL system of the form

$$\dot{x}(t) = A_{\sigma(t)}x(t) + B_{\sigma(t)}u(t), \tag{4.4}$$

with unknown subsystem parameters $A_i \in \mathbb{R}^{n\times n}$ and $B_i \in \mathbb{R}^{n\times p}$. Under the assumption of a known switching signal $\sigma(t)$, it is intuitive to design an individual parameter identifier for each subsystem $i \in \mathcal{I} := \{1, \ldots, s\}$. Despite the intuitive idea, the switching signal needs to be carefully included into the update process in order to maintain stability of the switched system.

Note here that the analysis of switched systems is considerably more challenging than the LTI case above. While a local identifier for each subsystem results in stable and converging estimates for each subsystem alone, this does not hold for the switched system. As pointed out in the introduction to switched systems in Section A.1, a poorly chosen switching sequence can destabilize even a switched system with only stable subsystems. Hence, we first highlight how the state prediction must be modified in order to include the switching signal and maintain stability. Afterwards, the convergence of parameter identifiers for PWL and PWA systems are discussed separately.

4.2.1 State Prediction under Switching

The switching signal is incorporated into the update laws by predicting the state of the system only for the currently active mode ($\sigma(t) = i$):

$$\dot{\hat{x}}_i = A_\mathrm{m}\hat{x}_i + \left(\hat{A}_i - A_\mathrm{m}\right) x + \hat{B}_i u, \quad \text{if } \sigma = i. \tag{4.5}$$

For inactive subsystems the prediction error is unrelated. Hence, once the system switches to $\sigma(t) \neq i$, the estimated state of the i-th subsystem is reset and kept equal to the true state measurement as long as the subsystem remains inactive. That means

$$\hat{x}_i = x, \quad \text{if } \sigma \neq i. \tag{4.6}$$

Figure 4.1 visualizes the state prediction for the i-th subsystem with continuous prediction 4.5 and discrete *state reset* (4.6). This reset is the crucial step in extending parameter identifiers to switched systems because it ensures that switching between different subsystems does not destabilize the overall system.

Another important insight, which can also be seen from the figure, is that the state reset ensures zero prediction errors, i.e. $\tilde{x}_i = x_i - x = 0$, for all inactive subsystems ($i \neq \sigma(t)$). Hence, applying the update laws (4.4) to inactive subsystems causes no change in the estimates $\hat{A}_i$, $\hat{B}_i$.

An intuitive explanation for the state reset idea follows from the fact that the prediction errors $\tilde{x}_i$ are applied for adaptation. One relies on $\tilde{x}_i$ as an indicator for how well the estimates $\hat{A}_i$ and $\hat{B}_i$ of the i-th subsystem match the real system parameters. In order to make a reasonable statement about how close the estimates $\hat{A}_i$, $\hat{B}_i$ are to the true parameters and how they should be adjusted to bring them closer, one should calculate the prediction error $\tilde{x}_i$ only for the active subsystem. The state reset (4.6) achieves that $\tilde{x}_i$ only depends on the mismatch $\hat{A}_i - A_i$ and $\hat{B}_i - B_i$.

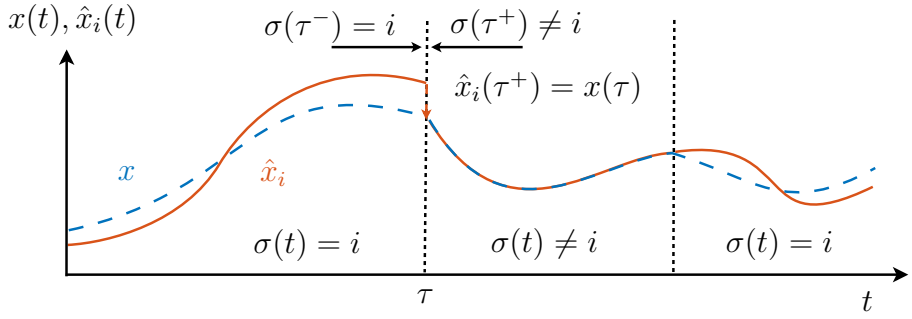


Fig. 4.1: Discrete reset of the state estimation for the i-th subsystem upon deactivation of the i-th subsystem dynamics, i.e. $\hat{x}_i(\tau^+) = x(\tau)$ as $\sigma(\tau^-) = i$ and $\sigma(\tau^+) \neq i$.

4.2.2 Parameter Identifiers for PWL Systems

An intuitive interpretation of the identifiers for switched systems is that a separate identifier is initialized for each subsystem and applied only during the time frames in which the corresponding subsystem is active. With the state reset above, one ensures that the error system is stable even under arbitrary fast switching. Furthermore, as the state reset ensures $\tilde{x}_i = 0, \forall i \neq \sigma(t)$, the update laws from (4.3) can be applied separately to each subsystem, which is written as

$$\dot{\hat{A}}_i = -\Gamma_1 P \tilde{x}_i x^\top, \qquad \dot{\hat{B}}_i = -\Gamma_2 P \tilde{x}_i u^\top, \quad \forall i \in \mathcal{I}. \tag{4.7}$$

Next, note that the parameter identifiers in Theorem 4.4 tie convergence of the estimates to the spectral properties of the input signals. Overall, this requires the LTI system to satisfy certain PE conditions. For the considered extension to switched linear systems, however, PE should not only reflect sufficient excitation in each subsystem but also how frequently each subsystem is visited over time. Only if each subsystem is PE and visited in a persistent manner, it is possible to estimate all system parameters. The first formal but restrictive notion for PE in switched linear systems is given in [199]. In this work, we simply require the input signals to be sufficiently rich and such that they repeatedly drive the PWL system through all regions of the partitioned state space in order to ensure PE of the switched system.

With these insights, parameter identifiers can be applied to identify PWL systems as specified by the following theorem.

Theorem 4.3 (Parameter identifiers for PWL systems) Consider the PWL system (4.4) with controllable pairs (A_i, B_i). Let the state of the system be predicted by (4.5) and (4.6) for all subsystems $i \in \mathcal{I}$. If the stabilizing input signals in u are sufficiently rich of order $n+1$ with distinct frequencies, and such that they cause repeated activation of all subsystems obeying a certain dwell time T_0, then the update laws (4.7) cause the estimates $\hat{A}_i$, $\hat{B}_i$ to converge to the real system matrices A_i, B_i and cause the prediction errors $\tilde{x}_i = \hat{x}_i - x$ to converge to zero.

Proof: (Preliminary versions of this proof are given in [273] and [281]) The proof is separated into two parts. The first part shows stability of all errors in the switched system.

Multiple Lyapunov functions guarantee stability even for arbitrary fast switching. The second part proves convergence to the true parameters with the help of PE requirements. Convergence is achieved under a loose minimal dwell-time condition to ensure sufficient excitation of the system.

Part I - Stability: We formulate a quadratic candidate Lyapunov function for each subsystem in terms of the parameter errors $\tilde{A}_i = \hat{A}_i - A_i$ and $\tilde{B}_i = \hat{B}_i - B_i$, which are therefore combined in a single column vector, the *parameter error vector*

$$\tilde{\theta}_i = \begin{bmatrix} \mathrm{vec}(\tilde{A}_i) \\ \mathrm{vec}(\tilde{B}_i) \end{bmatrix} \in \mathbb{R}^{n(n+p)}. \tag{4.8}$$

Here, the vec-operator concatenates the columns of a matrix in a single vector. Making use of the Kronecker product $\otimes$, one can introduce the matrices

$$\Gamma := \begin{bmatrix} \mathit{\Gamma}_1 I_{nn} & 0 \\ 0 & \mathit{\Gamma}_2 I_{np} \end{bmatrix}, \qquad \Psi := \begin{bmatrix} x \\ u \end{bmatrix} \otimes I_n, \tag{4.9}$$

with identity matrices I_{nn} and I_{np} of dimensions $nn \times nn$ and $np \times np$, respectively. This allows expressing most equations in a more compact form, which in turn simplifies the later analysis. The derivative of the prediction error for the active subsystem is given in terms of $\tilde{\theta}_i$ by

$$\begin{aligned} \dot{\tilde{x}}_i &= \dot{\hat{x}}_i - \dot{x} = A_\mathrm{m}\tilde{x}_i + \tilde{A}_i x + \tilde{B}_i u \\ &= A_\mathrm{m}\tilde{x}_i + \Psi^\top \tilde{\theta}_i. \end{aligned} \tag{4.10}$$

Also, with (4.8) and (4.9), the update laws (4.7) take the form

$$\dot{\tilde{\theta}}_i = \dot{\hat{\theta}}_i = -\Gamma \Psi P \tilde{x}_i. \tag{4.11}$$

For each subsystem i, consider a quadratic candidate Lyapunov function

$$V_i = \begin{bmatrix} \tilde{x}_i^\top & \tilde{\theta}_i^\top \end{bmatrix} \begin{bmatrix} P & 0 \\ 0 & \Gamma^{-1} \end{bmatrix} \begin{bmatrix} \tilde{x}_i \\ \tilde{\theta}_i \end{bmatrix}. \tag{4.12}$$

Its time derivative along (4.10) and (4.11) yields

$$\begin{aligned} \dot{V}_i &= \tilde{x}_i^\top A_\mathrm{m}^\top P \tilde{x}_i + \tilde{x}_i^\top P A_\mathrm{m} \tilde{x}_i + \tilde{\theta}_i^\top \Psi P \tilde{x}_i + \tilde{x}_i^\top P \Psi^\top \tilde{\theta}_i - \tilde{x}_i^\top P^\top \Psi^\top \Gamma^\top \Gamma^{-1} \tilde{\theta}_i - \tilde{\theta}_i^\top \Gamma^{-1} \Gamma \Psi P \tilde{x}_i \\ &= \tilde{x}_i^\top (A_\mathrm{m}^\top P + P A_\mathrm{m}) \tilde{x}_i = -\tilde{x}_i^\top Q_{\tilde{x}} \tilde{x}_i, \qquad \text{if } \sigma = i. \end{aligned} \tag{4.13}$$

For the inactive subsystems ($\sigma \neq i$) on the other side, it follows that $\tilde{x}_i = 0$ due to the state reset (4.6). Therefore, the Lyapunov function of the inactive subsystems remains constant, i.e.

$$\dot{V}_i = 0, \qquad \text{if } \sigma \neq i \tag{4.14}$$

While the negative semi-definiteness of V_i shown in (4.13) and (4.14) yields stability in the sense of Lyapunov for all subsystems individually, it does not (on its own) guarantee stability of the overall switched system. In order to prove stability for the obtained hybrid

system, it has to be shown that the candidate Lyapunov functions are Lyapunov-like functions along all possible trajectories and switching sequences [40, 149]. As discussed in Section A.1, this requires the values of a Lyapunov function V_i at each activation of the corresponding subsystem i to form a decreasing sequence.

A precise notation for the switching instances is required to show that the candidate Lyapunov functions are Lyapunov-like. Express the q-th activation of subsystem i by $\tau_{i,q}^{\text{in}}$ (we have $\sigma^-(\tau_{i,q}^{\text{in}}) \neq i$ and $\sigma^+(\tau_{i,q}^{\text{in}}) = i$). Similarly, the instance $\tau_{i,q}^{\text{out}}$ refers to the q-th exit from subsystem i ($\sigma^-(\tau_{i,q}^{\text{out}}) = i$ and $\sigma^+(\tau_{i,q}^{\text{out}}) \neq i$). With this notation, the following requirement arises [40]:

$$V_i(\tau_{i,q}^{\text{in}}) \geq V_i(\tau_{i,q+1}^{\text{in}}), \quad \forall q \geq 1. \tag{4.15}$$

Due to the negative semi-definiteness in (4.13), it follows that

$$V_i(\tau_{i,q}^{\text{in}}) \geq V_i(\tau_{i,q}^{\text{out}}). \tag{4.16}$$

Furthermore, the state reset associated with (4.6) yields $\tilde{x}_i(\tau) = 0$ for the inactive periods $\tau \in (\tau_{i,q}^{\text{out}}, \tau_{i,q+1}^{\text{in}}]$. With $\tilde{x}_i = 0$, the adaptation pauses and the parameter errors remain constant, i.e. $\tilde{\theta}_i(\tau_{i,q}^{\text{out}}) = \tilde{\theta}_i(\tau_{i,q+1}^{\text{in}})$. We therefore have

$$V_i(\tau_{i,q}^{\text{out}}) = \left[\tilde{x}_i^\top P \tilde{x}_i + \tilde{\theta}_i^\top \Gamma^{-1} \tilde{\theta}_i\right]_{\tau_{i,q}^{\text{out}}} \geq \left[\tilde{\theta}_i^\top \Gamma^{-1} \tilde{\theta}_i\right]_{\tau_{i,q+1}^{\text{in}}} = V_i(\tau_{i,q+1}^{\text{in}}), \tag{4.17}$$

which shows that (4.15) is satisfied. In other words: Switching between different subsystems does not cause divergence of the parameter estimates. It is left to show that all errors converge to zero. This step is carried out for each subsystem individually, hence the subsystem index i is neglected for the remainder of the proof.

Part II - Convergence: The proof of convergence for the individual linear subsystems was previously described in [114, p. 306] and relies on Lemma A.5 and Lemma A.6 in Section A.2. In addition to the previously discussed stability of the switched system, the convergence of all errors to zeros demands for a certain dwell time in order to sufficiently excite each subsystem such that $\dot{V}_i < 0$, $\forall \tilde{\theta}_i \neq 0$ while $\sigma = i$. Let the dwell time be equal to the constant T_0 in the definition of persistence of excitation (see Definition A.3 in Section A.2). The shorter this dwell time is, the lower is the level of excitation and the slower do the parameter errors $\tilde{\theta}_i$ vanish.

First, the error dynamics (4.10) and (4.11) are rewritten in the form of Lemma A.6 given in Section A.2. One obtains

$$\begin{bmatrix} \dot{\tilde{x}} \\ \dot{\tilde{\theta}} \end{bmatrix} = \begin{bmatrix} A_{\text{m}} & \Psi^\top(t) \\ -\Gamma \Psi(t) P & 0 \end{bmatrix} \begin{bmatrix} \tilde{x} \\ \tilde{\theta} \end{bmatrix}, \tag{4.18}$$

with $\Psi = z \otimes I_n$. For convergence of $\tilde{x}$ and $\tilde{\theta}$, Lemma A.6 requires the vector

$$z = \begin{bmatrix} x \\ u \end{bmatrix} \in \mathbb{R}^{n+p}$$

to be PE, which is ensured by Lemma A.7 under the given assumptions. Consequently,

with Lemma A.6, the equilibrium $\tilde{x}_e = 0$, $\tilde{\theta}_e = 0$ of (4.18) is exponentially stable in the large, which implies that the sequence in (4.15) is strictly decreasing to zero. Including the assumption that each subsystem i is visited in a persistent manner, this means that all prediction errors $\tilde{x}_i$ as well as all parameter errors $\tilde{\theta}_i$ convergence to zero, i.e., $\hat{A}_i \to A_i$ and $\hat{B}_i \to B_i$, $\forall i$ as $t \to \infty$. □

4.2.3 Parameter Identifiers for PWA Systems

Finally, let us analyze under which assumptions the discussed parameter identifiers are applicable to PWA systems. As discussed before, the application of PWA systems in parameter identifiers is beneficial compared to PWL systems in a sense that they require less previous knowledge about the system. More specifically, considering PWA systems allows to work directly with the measured states rather than with the deviation from some operating points Δx_i. Hence, no knowledge of operating points is needed when working with PWA systems.

Consider the PWA system

$$\dot{x}(t) = A_{\sigma(t)}x(t) + B_{\sigma(t)}u(t) + f_{\sigma(t)}, \tag{4.19}$$

with unknown subsystem parameters $A_i \in \mathbb{R}^{n\times n}$, $B_i \in \mathbb{R}^{n\times p}$ and $f_i \in \mathbb{R}^n$. Based on the previous assumption of a known switching signal $\sigma(t)$, one introduces estimates $\hat{A}_i$, $\hat{B}_i$ and $\hat{f}_i$ for each subsystem $i \in \mathcal{I}$ and designs update laws that provably drive these estimates to the true parameters. The update laws are again based on the prediction error $\tilde{x}_i = \hat{x}_i - x$. For affine subsystems the state of the currently active subsystem is predicted by

$$\dot{\hat{x}}_i = A_{\mathrm{m}}\hat{x}_i + \left(\hat{A}_i - A_{\mathrm{m}}\right)x + \hat{B}_i u + \hat{f}_i, \qquad \text{if } \sigma = i. \tag{4.20}$$

As in the PWL case above, let the predicted states $\hat{x}_i$ of all inactive subsystems ($i \neq \sigma(t)$) be reset and kept equal to the measured state x (see Fig. 4.1). That means

$$\hat{x}_i = x, \qquad \text{if } \sigma \neq i, \tag{4.21}$$

which renders the prediction errors of all inactive subsystems ($i \neq \sigma(t)$) zero.

Next, note that the affine vector f_i can be interpreted as an additional element of the input matrix B_i associated with a constant input 1, which yields

$$\dot{x}(t) = A_i x(t) + \begin{bmatrix} B_i & f_i \end{bmatrix} \begin{bmatrix} u(t) \\ 1 \end{bmatrix}. \tag{4.22}$$

By applying the same update laws as in the PWL case (4.7), and with some reformulation, one obtains the following update laws for the parameter estimates of an affine subsystem

$$\dot{\hat{A}}_i = -\Gamma_1 P\tilde{x}_i x_i^\top, \qquad \dot{\hat{B}}_i = -\Gamma_2 P\tilde{x}_i u_i^\top, \qquad \dot{\hat{f}}_i = -\Gamma_3 P\tilde{x}_i, \tag{4.23}$$

where $\Gamma_1,\ \Gamma_2,\ \Gamma_3 > 0$ are the scalar learning rates. Recall, however, the critical assumption in Theorem 4.3 that all input signals must be sufficiently rich of order $n+1$. The constant

element 1 in (4.22) does not satisfy this assumption, which raises the question whether one can still prove convergence in the case of affine subsystems? The answer to this question lies in Lemma A.7, which ensures PE of the internal signal vector $[x^\top, u^\top]^\top$ in the linear case. For the affine case, the constant 1 appears in this vector and it needs to be shown that $[x^\top, u^\top, 1]^\top$ is PE. Hence, the following lemma is needed for the extension to PWA systems.

Lemma 4.4 (PE of internal signals – affine system) Consider an affine subsystem in (4.19) with controllable pair (A, B) and invertible A. If the input signals in u are sufficiently rich of order $n+1$ with distinct frequencies, then the vector $z = [x^\top, u^\top, 1]^\top$ is persistently exciting.

Proof: The technical proof to Lemma 4.4 is given in Appendix D for improved readability.□

Lemma 4.4 extends the state of the art in a sense that violating the common assumption of sufficiently rich input signals can be compensated by restricting the system to invertible system matrices. This insight enables the application of parameter identifiers for PWA systems, which is formalized by the following theorem.

Theorem 4.5 (Parameter identifiers for PWA systems) Consider the PWA system (4.19) with controllable pairs (A_i, B_i) and A_i invertible. Let the state of the system be predicted by (4.20) and (4.21) for all subsystems $i \in \mathcal{I}$. If the stabilizing input signals in u are sufficiently rich of order $n+1$ with distinct frequencies, and such that they cause repeated activation of all subsystems obeying a certain dwell time T_0, then the update laws (4.23) cause the estimates $\hat{A}_i,\ \hat{B}_i,\ \hat{f}_i$ to converge to the real system matrices $A_i,\ B_i,\ f_i$ and cause the prediction errors $\tilde{x}_i = \hat{x}_i - x$ to converge to zero.

Proof: (Preliminary versions of this proof are given in [276] and [281]) The proof of Theorem 4.5 follows the same steps as the one of Theorem 4.3. First, with a slight abuse of notation, stability of the switched error system is proven by multiple Lyapunov functions. Afterwards, convergence to zero is concluded under some PE condition, which is guaranteed by Lemma 4.4.

Let the parameter errors of the i-th subsystem be $\tilde{A}_i = \hat{A}_i - A_i$, $\tilde{B}_i = \hat{B}_i - B_i$ and $\tilde{f}_i = \hat{f}_i - f_i$ and let

$$\tilde{\theta}_i = \begin{bmatrix} \mathrm{vec}(\tilde{A}_i) \\ \mathrm{vec}(\tilde{B}_i) \\ \tilde{f}_i \end{bmatrix} \in \mathbb{R}^{n(n+p+1)}. \tag{4.24}$$

With a slight abuse of notation (compared to the proof of Theorem 4.3), introduce the matrices

$$\Gamma := \begin{bmatrix} \mathit{\Gamma}_1 I_{nn} & 0 & 0 \\ 0 & \mathit{\Gamma}_2 I_{np} & 0 \\ 0 & 0 & \mathit{\Gamma}_3 I_n \end{bmatrix}, \qquad \Psi := \begin{bmatrix} x \\ u \\ 1 \end{bmatrix} \otimes I_n, \tag{4.25}$$

where $\otimes$ represents the Kronecker product, and I_n, I_{nn} and I_{np} are identity matrices of dimensions $n \times n$, $nn \times nn$ and $np \times np$, respectively. Equivalent to the PWL case (see

(4.10) and (4.11)) one obtains the following, compact representations:

$$\dot{\tilde{x}}_i = A_\mathrm{m}\tilde{x}_i + \Psi^\top\tilde{\theta}_i, \qquad \text{for } \sigma = i, \tag{4.26}$$

and

$$\dot{\tilde{\theta}}_i = -\Gamma\Psi P\tilde{x}_i. \tag{4.27}$$

Consider the same Lyapunov function

$$V_i = \begin{bmatrix}\tilde{x}_i^\top & \tilde{\theta}_i^\top\end{bmatrix}\begin{bmatrix}P & 0\\ 0 & \Gamma^{-1}\end{bmatrix}\begin{bmatrix}\tilde{x}_i\\ \tilde{\theta}_i\end{bmatrix}, \tag{4.28}$$

and obtain its derivative along (4.26) and (4.27) as $\dot{V}_i = -\tilde{x}_i^\top Q_{\tilde{x}}\tilde{x}_i$. Due to the negative semi-definiteness of $\dot{V}_i$ and the state reset (4.21), stability of the switched systems is concluded with the help of multiple Lyapunov functions [40] (compare (4.15)-(4.17)).

The proof of convergence, is essentially a combination of Lemma A.6 in Section A.2 and Lemma 4.4 proposed above. Note that convergence does not critically depend on the switching signal σ and can therefore be analyzed for every subsystem individually. Hence, we neglect the system index i for the rest of the proof and only refer to the currently active subsystem.

With (4.26) and (4.27), the system dynamics of the active subsystem are in the form of Lemma A.6 given by

$$\begin{bmatrix}\dot{\tilde{x}}\\ \dot{\tilde{\theta}}\end{bmatrix} = \begin{bmatrix}A_\mathrm{m} & \Psi^\top(t)\\ -\Gamma\Psi(t)P & 0\end{bmatrix}\begin{bmatrix}\tilde{x}\\ \tilde{\theta}\end{bmatrix}, \tag{4.29}$$

where $\Psi = z \otimes I_n$ and

$$z = \begin{bmatrix}x\\ u\\ 1\end{bmatrix} \in \mathbb{R}^{n+p+1}$$

According to Lemma A.6, z needs to be PE, which is guaranteed by Lemma 4.4. Consequently, the equilibrium $\tilde{x} = 0$, $\tilde{\theta} = 0$ of (4.29) is exponentially stable in the large. Including the assumption that each subsystem i is visited in a persistent manner, this implies that all Lyapunov functions V_i converge to zero. Therefore, also the prediction errors $\tilde{x}_i$ as well as all parameter errors $\tilde{\theta}_i$ convergence to zero, i.e., $\hat{A}_i \to A_i$, $\hat{B}_i \to B_i$ and $\hat{f}_i \to f_i$, $\forall i$ as $t \to \infty$. □

With Theorem 4.3 and Theorem 4.5, this chapter introduced extensions of parameter identifiers from linear state-space systems to PWL and PWA systems. Before proceeding with numerical analysis of the identification performance, some comments about the theorems, their proofs and the implications are in order.

Note that both Theorems 4.3 and 4.5 assumed a known switching signal σ. Besides the assumption that the switching signal activates all subsystems in a persistent manner, there is no restriction on how the switching signal is generated. Therefore, the proposed identifiers can be applied to switched systems under all kinds of switching, such as state-dependent, time-dependent, autonomous and controlled switching. The main motivation

in this thesis for PWA systems thus only constitutes a special case in which the switching hyperplanes of the PWA system are assumed to be known and in combination with the accessible state measurements enable reconstructing the switching signal at run-time.

The assumption of known switching hyperplanes constitutes at the same time one of the mayor limitations of the proposed identifiers. While it may be possible to define a larger number of fictitious partitions and identify redundant models, it is of course more elegant to estimate the number of state-space partitions and their location from measured data. Chapter 6 will discuss two algorithms capable of doing so. One of them is closely related to the identifiers discussed here.

Theorem 4.3 and Theorem 4.5 both assume a certain dwell time of the switched system. This dwell-time assumption is however not restrictive. As the Lyapunov analysis in both cases showed, it turns out that arbitrary fast switching does not cause divergence of the estimated parameters. This property is essentially due to the state reset at each switch which bounds the prediction error. The dwell time is thus mainly concerned with the convergence rate. If little time is spent in a subsystem, the corresponding estimates receive only minor adaptation, resulting in slow convergence.

4.3 Simulation Studies

The presented parameter identifiers are numerically tested for exemplary PWL and PWA systems in Section 4.3.1 and Section 4.3.2, respectively. Finally, a hybrid robot-contact problem is considered in Section 4.3.3

4.3.1 Estimation in PWL Systems

Consider the PWL system with subsystem parameters

$$A_1 = \begin{bmatrix} 0 & 1 \\ -2.6 & -7.8 \end{bmatrix}, \qquad A_2 = \begin{bmatrix} 0 & 1 \\ -2.9 & -8.6 \end{bmatrix}, \qquad A_3 = \begin{bmatrix} 0 & 1 \\ -2.3 & -7.0 \end{bmatrix},$$
$$B_1 = \begin{bmatrix} 0 \\ 1.0 \end{bmatrix}, \qquad B_2 = \begin{bmatrix} 0 \\ 0.9 \end{bmatrix}, \qquad B_3 = \begin{bmatrix} 0 \\ 1.1 \end{bmatrix},$$

and let the state space be partitioned by three manifolds of the form $x_2 = \alpha_i(x_1 - \gamma)$ with $\alpha_1 = 0.1$, $\alpha_2 = 1.3$, $\alpha_3 = -0.4$ and $\gamma = 0.4$. In the hyperplane framework applied in Section 2.1, these manifolds correspond to vectors

$$h_1 = \begin{bmatrix} 0.1 & -1 & 0 & -0.04 \end{bmatrix}, \quad h_2 = \begin{bmatrix} 1.3 & -1 & 0 & -0.52 \end{bmatrix}, \quad h_3 = \begin{bmatrix} -0.4 & -1 & 0 & 0.16 \end{bmatrix}.$$

The state-space partitions are visualized in Fig. 4.2 together with the state-space trajectory obtained for the input signal

$$u(t) = 2\sin(0.05t) + 2\sin(3t) + \sin(5t).$$

In this choice of reference signals the low frequency of 0.05 rad/s causes repeated activation of all subsystems and the higher frequencies are applied to satisfy the requirement of

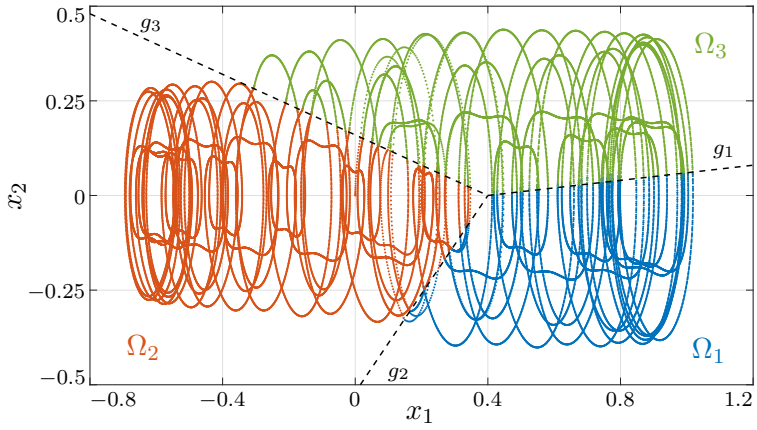


Fig. 4.2: Trajectory through partitioned state space for identifying subsystems of PWL system.

sufficiently rich inputs.

Let all estimates be initialized with zero. Then, with design parameters $A_\mathrm{m} = -I_2$, $\Gamma_1 = 10$ and $\Gamma_2 = 1$, the proposed parameter identifiers yield the temporal evolution of estimated parameters shown in Fig. 4.3. Time frames in which a subsystem is active are represented by solid lines, whereas inactive periods are represented by dotted lines. As suggested by Theorem 4.3, all estimates converge to their nominal values.

Analyzing the plots in more detail, it can be seen that the PWL system frequently switches between subsystem 1 and 3. This is also why it seems like these two subsystems are simultaneously updated in Fig. 4.3. Despite the frequent switching, stability of all estimates is maintained throughout the simulation. Moreover, despite very short dwell times for subsystem 1 and 3 all estimates converge to their nominal parameters, which suggests that the dwell time assumption imposed by Theorem 4.3 is indeed not restrictive.

4.3.2 Estimation in PWA Systems

Next the proposed parameter identifiers are tested with the same second order PWA system with three regions as in [276]. The subsystem dynamics are characterized by

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

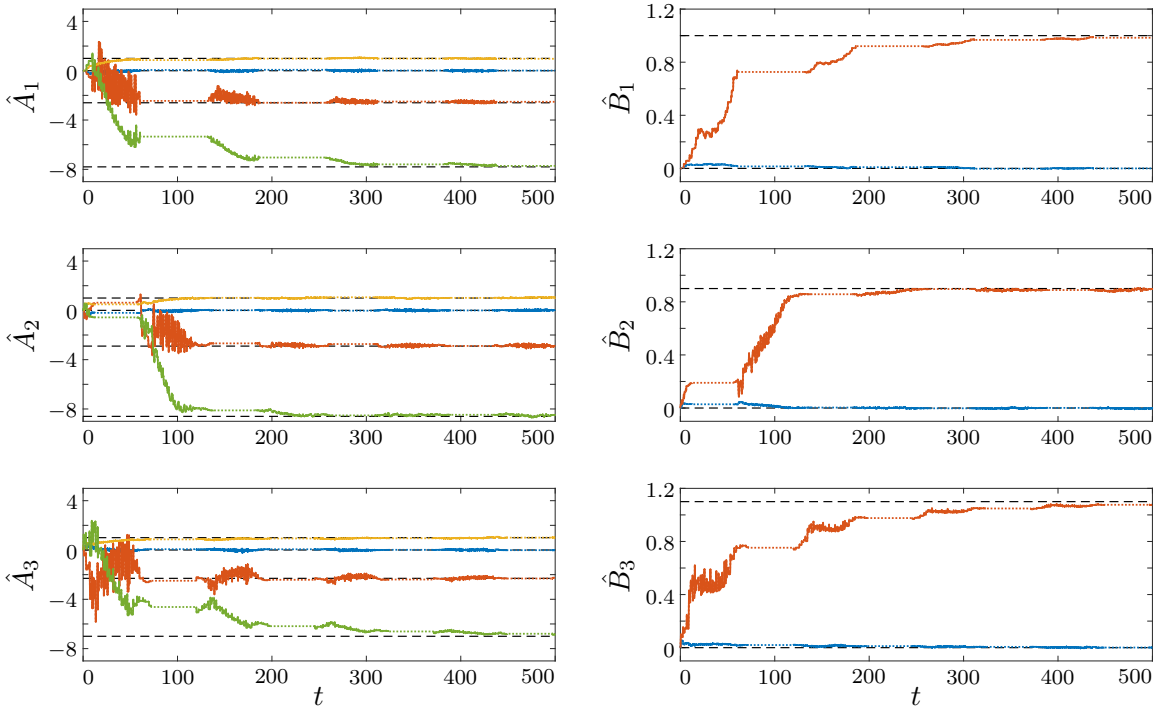


Fig. 4.3: Convergence of parameter estimates $\hat{A}_i, \hat{B}_i$ (colored lines) to their nominal values (black dashed lines). Periods of active adaptation are visualized by solid lines and inactive periods by dotted lines.

$B_i = [0,\ 1.5]^\top\ \forall i$, and the polyhedral regions are

$$\Omega_1 = \left\{ x \in \mathbb{R}^2 \,\middle|\, -2 \leq x_1 \leq 2 \right\}, \qquad \mathcal{H}_1 = \begin{bmatrix} 1 & 0 & 0 & -2 \\ -1 & 0 & 0 & -2 \end{bmatrix}, \qquad \preceq_{[1]} = \begin{bmatrix} \leq \\ \leq \end{bmatrix},$$
$$\Omega_2 = \left\{ x \in \mathbb{R}^2 \,\middle|\, x_1 > 2 \right\}, \qquad \mathcal{H}_2 = \begin{bmatrix} -1 & 0 & 0 & 2 \end{bmatrix}, \qquad \preceq_{[2]} = \begin{bmatrix} < \end{bmatrix},$$
$$\Omega_3 = \left\{ x \in \mathbb{R}^2 \,\middle|\, x_1 < -2 \right\}, \qquad \mathcal{H}_3 = \begin{bmatrix} 1 & 0 & 0 & 2 \end{bmatrix}, \qquad \preceq_{[3]} = \begin{bmatrix} < \end{bmatrix}.$$

Let the system be excited with the input signal

$$u(t) = \bar{u}(t) + \sum_{l=1}^{3} \sin(\omega_l t),$$

where $\omega_1 = 0.5\,\mathrm{rad/s}$, $\omega_2 = 1\,\mathrm{rad/s}$ and $\omega_3 = 3\,\mathrm{rad/s}$. The piecewise constant signal $\bar{u}(t)$ switches every 200 s between the constant offsets $\bar{u}_1 = 0$, $\bar{u}_2 = 6$ and $\bar{u}_3 = -6$ in order to drive the system into the three corresponding regions.

Again, all estimates are initialized with zeros. The design parameters are chosen as $A_\mathrm{m} = -10 I_2$, $\Gamma_1 = \Gamma_2 = \Gamma_3 = 1$ for which the estimation process is visualized in Fig. 4.4 for all three subsystems. As can be seen from the figure, all estimates converge towards their nominal values.

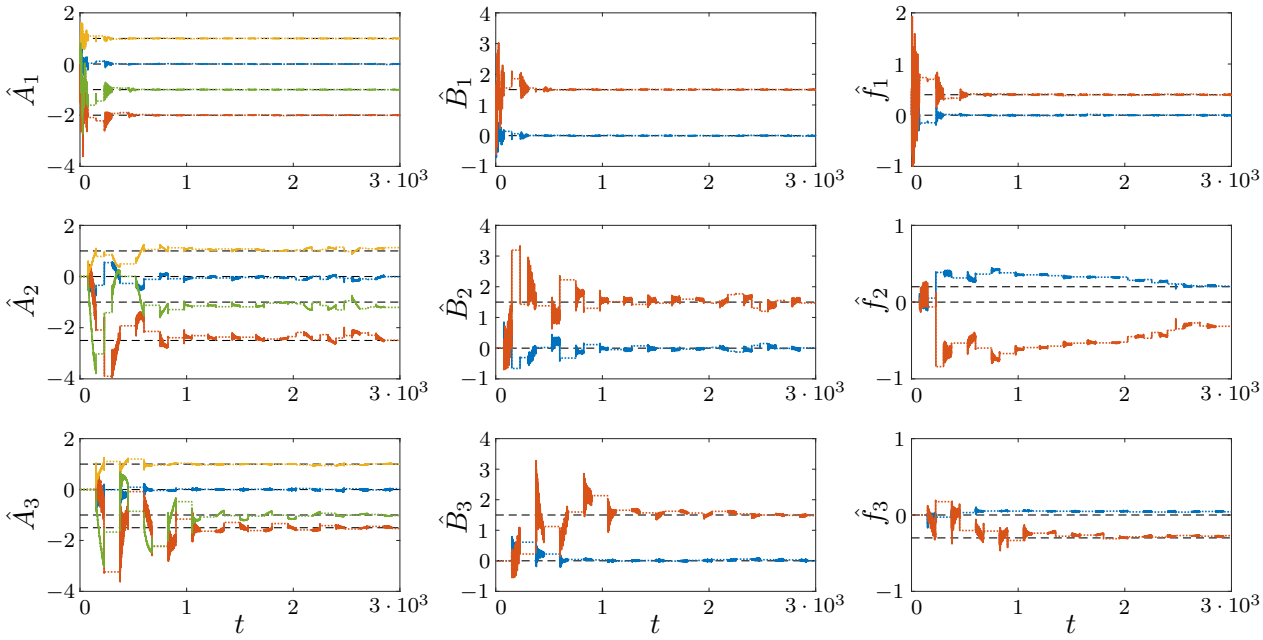


Fig. 4.4: Convergence of parameter estimates $\hat{A}_i, \hat{B}_i, \hat{f}_i$ (colored lines) to their nominal values (black dashed lines). Periods of active adaptation are visualized by solid lines and inactive periods by dotted lines.

Comparing the speed of convergence achieved for the three subsystems, it turns out that subsystem 1 converges considerably faster than the other two subsystems. Note, that it is in general difficult to determine or even to tune the actual convergence speed in adaptive systems due to their highly nonlinear structure. Additional simulation studies, however, reveal that the offset in the input signal is responsible for the reduced convergence rate. To see this, consider the second subsystem alone, with the affine dynamics given by A_2, B_2, f_2. Let the affine system be excited by the same input signal as before with different constant offsets $\bar{u}$ ranging from 0 to 10. For the same design parameters as before and $A_\mathrm{m} = -10I_2$, Fig. 4.5 shows the normed parameter error $\|\tilde{f}_2\|$, whose convergence rate clearly decreases with increasing offset. Unfortunately the offset is needed to drive the state of the PWA system through all state-space partitions and may thus not be removed from the input signal. In Chapter 5, the proposed parameter identifiers are extended by concurrent learning, which yields improved convergence rates.

4.3.3 Robot-Contact Problem

Finally, the proposed parameter identifiers are numerically tested in the simplified hybrid robot-contact problem shown in Fig. 4.6. The figure shows a robot arm and a lever, which both rotate around the same axis. The shared workspace can cause collisions and contacts between the two objects. With the PWA systems framework applied in this thesis, the robot-contact system is expressed as a hybrid system with two continuous subsystems representing the modes contact and free movement, and a reset map to reset the state at each collisions.

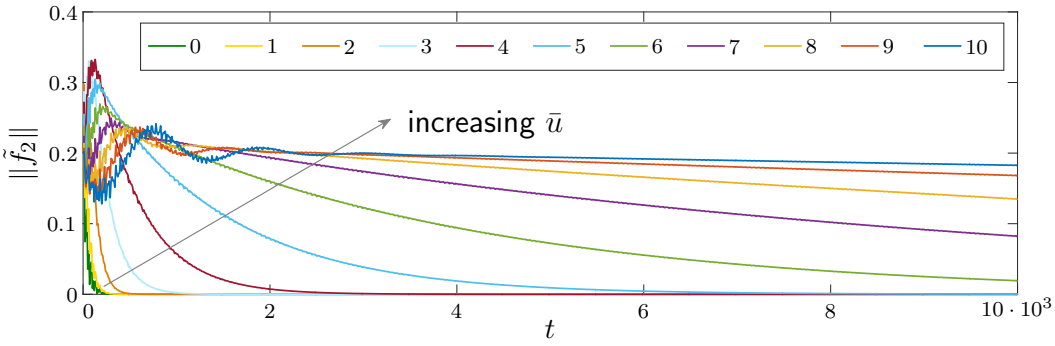


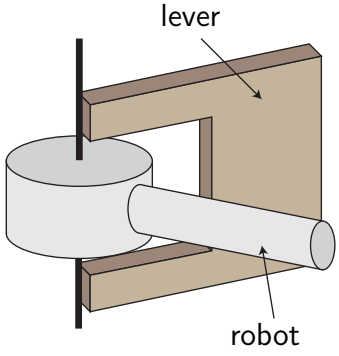
Fig. 4.5: Parameter convergence of $\tilde{f}_2$ for input signals with increasing offset values $\bar{u}$.

The system dynamics are expressed in terms of the robots deflection angle φ_r and its angular velocity $\dot{\varphi}_r$ as well as the deflection angle φ_l and angular velocity $\dot{\varphi}_l$ of the lever. Assume that both objects are attached to the environment via torsional springs (not shown in the figure) which push the objects to their respective rest positions $\varphi_{0,r} = 0$ and $\varphi_{0,l} = -10\,^\circ = -0.1745\,\text{rad}$. Hence, the torsional springs cause restoring torques of the form $\tau_{s,r} = -k_r\varphi_r$ and $\tau_{s,l} = -k_l(\varphi_l - \varphi_{0,l})$, where k_r and k_l are the spring constants of the robot and lever, respectively. Let the robot arm and the lever be damped by friction, which is expressed by torques of the form $\tau_{d,r} = -c_r\dot{\varphi}_r$ and $\tau_{d,l} = -c_l\dot{\varphi}_l$, where c_r and c_l are the friction coefficients of robot and lever. Finally, the input torque $\tau_u = u$ can be applied directly to the robot arm. Let the robot and lever inertia be I_r and I_l, respectively, and note that all system parameters are summarized by the table in Fig. 4.6.

Consider first the case in which there is no contact between robot and lever. The dynamics of such a free movement are expressed by the following state-space system

$$\dot{x} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{k_r}{I_r} & -\frac{c_r}{I_r} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{k_l}{I_l} & -\frac{c_l}{I_l} \end{bmatrix}}_{A_1} x + \underbrace{\begin{bmatrix} 0 \\ \frac{1}{I_r} \\ 0 \\ 0 \end{bmatrix}}_{B_1} u + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{\varphi_{0,l}k_l}{I_l} \end{bmatrix}}_{f_1}. \tag{4.30}$$

with state vector $x = [\varphi_r,\ \dot{\varphi}_r,\ \varphi_l,\ \dot{\varphi}_l]^\top$. Restricting the deflection angles to $\varphi_{r/l} \in [-\pi, \pi]$,



	Robot	Lever
k	$1\,\text{Nm/rad}$	$3\,\text{Nm/rad}$
c	$2\,\text{Nms/rad}$	$1.5\,\text{Nms/rad}$
I	$1\,\text{kgm}^2$	$1.5\,\text{kgm}^2$

Fig. 4.6: Simplified robot contact problem

the free movement according to (4.30) applies for $\varphi_r < \varphi_l$.

An impact between robot and lever occurs for $\varphi_r = \varphi_l$ and $\dot{\varphi}_r > \dot{\varphi}_l$. Modeling of collisions is a complex task and we consider here a simplified case. That is, depending on the closing speed $\Delta v = \dot{\varphi}_r - \dot{\varphi}_l$, we distinguish between an elastic collision and a perfectly inelastic collision. For $\Delta v \geq 0.3\,\mathrm{rad/s}$, the collision is elastic, which leads to the reset maps

$$\dot{\varphi}_r = \frac{1}{I_r + I_l}\left(2I_l\dot{\varphi}_l^- + (I_r - I_l)\dot{\varphi}_r^-\right), \qquad \dot{\varphi}_l = \frac{1}{I_r + I_l}\left(2I_l\dot{\varphi}_r^- + (I_l - I_r)\dot{\varphi}_l^-\right). \tag{4.31}$$

For $\Delta v < 0.3\,\mathrm{rad/s}$, let the impact be perfectly inelastic, such that both the robot and lever obtain the same angular velocity given by

$$\dot{\varphi}_r = \dot{\varphi}_l = \frac{I_r\dot{\varphi}_r^- + I_l\dot{\varphi}_l^-}{I_r + I_l}. \tag{4.32}$$

After a perfectly inelastic collision, the system switches to contact mode as long as $\varphi_r = \varphi_l$, $\dot{\varphi}_r = \dot{\varphi}_l$, and as long as the input torque τ_u satisfies

$$\frac{-k_r\varphi_r - c_r\varphi_r + \tau_u}{I_r} \geq \frac{-k_l(\varphi_l - \varphi_{0,l}) - c_l\varphi_l}{I_l}. \tag{4.33}$$

In contact mode, the angles and angular velocities of lever and robot remain equal. Hence the system dynamics reduce to a two-dimensional state-space model of the form

$$\dot{x} = \underbrace{\begin{bmatrix} 0 & 1 \\ -\frac{k_r+k_l}{I_r+I_r} & -\frac{c_r+c_l}{I_r+I_r} \end{bmatrix}}_{A_2} x + \underbrace{\begin{bmatrix} 0 \\ \frac{1}{I_r+I_r} \end{bmatrix}}_{B_2} u + \underbrace{\begin{bmatrix} 0 \\ \frac{\varphi_{0,l}k_l}{I_r+I_r} \end{bmatrix}}_{f_2}. \tag{4.34}$$

When the contact condition (4.33) is not fulfilled anymore, the system switches back to its free space dynamics (4.30).

In order to identify both subsystems, the input signal was chosen as

$$u(t) = 3(\sin(0.05t))^2 + \sin(0.1t) + \sin(t) + \sin(2t), +2\sin(3t) + \sin(4t) + \sin(7t),$$

which is sufficiently rich for the identification task. Furthermore, the input causes various transitions between free movement and contact phase with elastic collisions. Figure 4.7 shows a 50 s time span in which the robot transits from its free movement to contact with the lever. State resets associated with collisions are highlighted by dotted segments in the plot of angular velocities. It is important to note here that the collisions also cause some excitation in the lever states while $\sigma = 1$.

The estimated parameters are shown in Fig. 4.8 for design parameters $A_\mathrm{m} = -P = I$ and $\Gamma_1 = \Gamma_2 = \Gamma_3 = 10$. As can be seen from the figure, both subsystem parameters are estimated correctly. In case of the free space movement ($\hat{\theta}_1$) this is a quite surprising result. In Theorem 4.5, parameter convergence was tied to controllable subsystems, a property that the dynamics in (4.30) clearly violate. In other words, the controlled input torque can not excite the lever states φ_l and $\dot{\varphi}_l$ in case the system remains in free movement without collisions. Without excitation, however, the parameters describing the lever dynamics

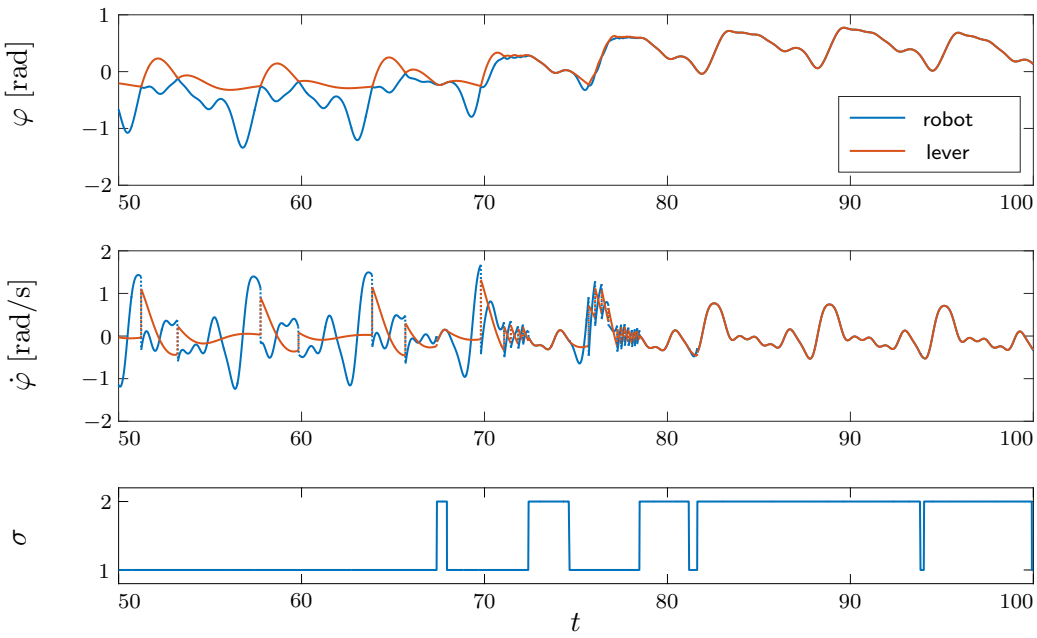


Fig. 4.7: Visualization of the transition from free movement to contact, which is characterized by multiple collisions. While both elastic and inelastic collisions cause state resets, only the final inelastic collision changes the mode of the system.

cannot be estimated from state measurements. In the present simulation study, the state reset associated with collisions introduces another source of excitation not covered by Theorem 4.5. That means the lever is excited through elastic collisions in such a way that the PWA system state satisfies the required PE condition.

Overall, these findings suggest some conservativeness in Theorem 4.5 and future work could investigate in greater detail the impact of state resets and switching on the identification.

4.4 Summary

This chapter presented the extension of parameter identifiers from linear systems to PWL and PWA systems as well as switched linear and switched affine systems in general. In parameter identifiers, the state of the system is predicted based on dynamic estimates of the system parameters. The difference between predicted and measured states drives the adaptation of the estimated parameters. Stability of the update law is ensured by Lyapunov analysis, while convergence is guaranteed under persistent excitation. An intuitive interpretation of the discussed extension to switched systems is that a separate identifier is initialized for each subsystem. Adaptation takes only place during time frames in which the corresponding subsystem is active. The set of identifiers form a hybrid system with continuous adaptation (of the active model) and discrete mode switches. In order to ensure stability of this hybrid system with the multiple-Laypunov-functions concept, the predicted state of inactive subsystems needs to be reset and kept equal to the actual state

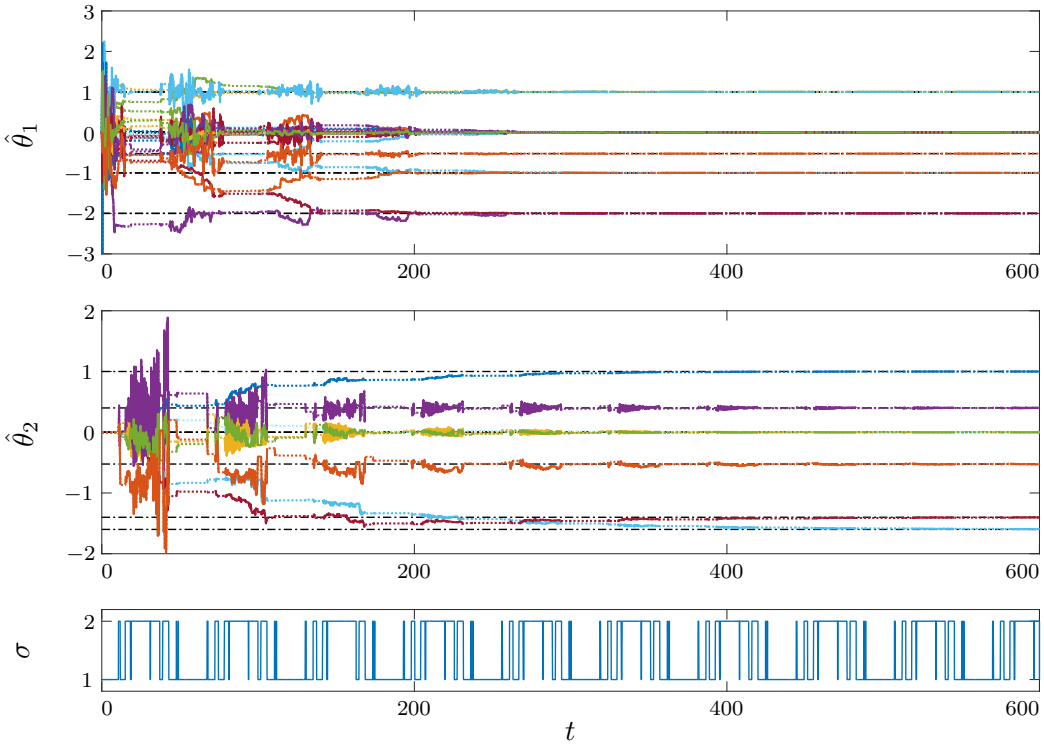


Fig. 4.8: Estimated parameters for the two subsystems in the robot-contact problem (free movement θ_1 and contact θ_2). The currently active and inactive subsystem are visualized by solid and dotted lines, respectively.The true values are shown by dash-dotted lines.

measurements. This state reset ensures stability of the hybrid system even in the presence of arbitrary fast switching. Another challenge, which was overcome in this chapter, is related to affine subsystems. The affine input vectors violate the common assumption of sufficiently rich input signals, which ensure persistent excitation for parameter convergence. It was shown that persistent excitation and hence parameter convergence can be maintained by restricting application to systems with invertible system matrices. The numerical simulations with exemplary piecewise linear and piecewise affine systems and a hybrid robot-contact problem confirmed the theoretically derived parameter convergence.

The class of systems considered in this chapter, i.e. multi-input switched affine and switched linear systems in state-space form with full state measurement and arbitrarily generated switching signal, is quite general, which makes the derived results applicable to a variety of practical systems. There are however a few restrictions. First, the identified system must be controllable in order ensure persistent excitation by means of sufficiently rich input signals. The numerical evaluation of the robot-contact problem has shown that this restriction may be too conservative as additional excitation of the system may be induced by the discrete mode switches. Second, in case of affine switched systems, the system matrices need to be invertible, in order to differentiate the affine input vector from the control input matrix. Third, the presented algorithms cannot be used to identify systems with partial state measurements. As the probably biggest limitation of the presented algo-

rithms is the assumption of a known switching signal, we present a modification towards handling unknown switching signals in Chapter 6.

The conducted simulation studies revealed three minor shortcomings. First, all input signals must be sufficiently rich of order $n + 1$ for all time in order to ensure persistent excitation and parameter convergence. While this is a common shortcoming of adaptive systems, it does limit the applicability. Second, the adaptation of a subsystem only takes place while the subsystem is active, which leads to the third shortcoming in form of rather slow convergence. These shortcomings can be overcome by introducing memory in the form of concurrent learning into the update laws. This will be presented in more detail in the following chapter.

5 Parameter Identifiers with Concurrent Learning

The parameter identifiers discussed in the previous chapter can estimate the subsystem parameters of PWA systems under the assumption of known state-space partitions. This chapter introduces a concurrent learning-based extension of the proposed parameter identifiers to overcome their three highlighted shortcomings. These are the requirement of persistent excitation throughout the identification period, which is both hard to achieve and monitor in practical systems, the fact that only active subsystems are updated, and the rather slow convergence rates.

Chowdhary *et al.* introduced the concept of concurrent learning to adaptive control in order to circumvent the limitations of persistent excitation in adaptive systems [52, 53, 56]. While traditional update laws in adaptive control rely only on current measurements of state and reference input, the central idea in *concurrent learning* is to use current measurements concurrently with past measurements. For this purpose, selected data points are stored in so called history stacks. By doing so, the information contained in past measurements can be repeatedly applied for adaptation, which reduces the required information content in current measurements. The concept of concurrent learning has been applied successfully in various fields such as aircraft control [55], optimization [126] and network synchronization [129]. However, the most promising field for concurrent learning are switched systems, where the use of memory enables a continuous adaptation of active and inactive subsystems [63, 274].

Despite the growing interest in concurrent learning, some of its properties still lack detailed analysis. Especially the effect of erroneous data in the memory has received little attention. It is noted in [56] that tracking errors remain uniformly ultimately bounded and that estimated parameters converge to a compact ball around their ideal values in case of erroneous history stack data. Unfortunately, the size of the compact ball is related to the errors in history stack data. Hence, the achievable convergence degrades for large errors in the history stack. Therefore, the second contribution in this chapter is an algorithm for detecting and removing erroneous history stack elements. With the proposed algorithm the precision of concurrent learning can be increased and tracking of time-varying system parameters is restored.

The extension of parameter identifiers with concurrent learning for PWA systems is subject of Section 5.1. Afterwards, Section 5.2 presents how to record and manage history stack data, with focus on the removal of erroneous data. Then, the proposed algorithms are demonstrated in numerical simulations in Section 5.4. Concluding remarks in Section 5.5 complete the chapter. Parts of this chapter are published in the journal article [281].

5.1 Concurrent Learning Adaptive Identification

Consider the PWA system

$$\dot{x}(t) = A_{\sigma(t)}x(t) + B_{\sigma(t)}u(t) + f_{\sigma(t)}, \tag{5.1}$$

whose unknown parameters A_i, B_i and f_i, $i \in \mathcal{I}$ can be estimated with the parameter identifiers

$$\dot{\hat{A}}_i^{\mathrm{C}}(t) = -\Gamma_1 P\tilde{x}_i(t)x(t)^\top, \qquad \dot{\hat{B}}_i^{\mathrm{C}}(t) = -\Gamma_2 P\tilde{x}_i(t)u(t)^\top, \qquad \dot{\hat{f}}_i^{\mathrm{C}}(t) = -\Gamma_3 P\tilde{x}_i(t), \tag{5.2}$$

introduced in the previous chapter. Recall that $\tilde{x}_i = \hat{x}_i - x$ is the prediction error and $\hat{x}_i$ is the state predicted according to (4.20) based on the estimates $\hat{A}_i$, $\hat{B}_i$ and $\hat{f}_i$. Note that the superscript $^{\mathrm{C}}$ in (5.2) indicates the dependency on current measurements alone (e.g. $\tilde{x}_i(t)$, $x(t)$ and $u(t)$). As shown in the proof of Theorem 4.5, the prediction errors $\tilde{x}_i$ play a central role in the update laws (5.2) as they contain information about the current parameter error, i.e., compare (4.26) and $\dot{\tilde{x}}_i = A_{\mathrm{m}}\tilde{x}_i + \tilde{A}_i x + \tilde{B}_i u + \tilde{f}_i$.

The key idea in concurrent learning is to find a similar expression which describes the parameter error based on recorded system states rather than current prediction errors. For this purpose, consider the expression

$$\varepsilon = \tilde{A}x + \tilde{B}u + \tilde{f},$$

which translates to

$$\varepsilon = \hat{A}x + \hat{B}u + \hat{f} - \dot{x}.$$

Hence given measurements of x, u and $\dot{x}$, it is possible to define the error ε, which constitutes the desired expression, and which is thus later going to drive the adaptation of the estimated parameters.

Update laws in concurrent learning-based adaptive control and identification make concurrent use of current and recorded data. Hence, the corresponding differential equations consist of two parts. The first part is based on current measurements and usually takes the form of traditional update laws. For the concurrent learning-based parameter identifiers proposed in this chapter, (5.2) serves as the current data-based part of the update laws. The second part is based on selected, past measurements that are stored in memory, the so-called history stacks. The arguments above motivate collecting a set of *triplets* $\{x_j, u_j, \dot{x}_j\}_{j=1}^q$ *of recorded data* for adaptation in concurrent learning.

For each subsystem i, a separate set of recorded data needs to be obtained. Therefore, denote with ($x_{i_j} \in \mathbb{R}^n$, $u_{i_j} \in \mathbb{R}^p$, $\dot{x}_{i_j} \in \mathbb{R}^n$) the j-th triplet that is recorded while subsystem i is active. A total of $q \in \mathbb{N}$ such triplets is recorded for each subsystem i and the elements of the triplets are stored in three *history stacks* $X_i := [x_{i_1}, x_{i_2}, \ldots, x_{i_q}]$, $U_i := [u_{i_1}, u_{i_2}, \ldots, u_{i_q}]$ and $\dot{X}_i := [\dot{x}_{i_1}, \dot{x}_{i_2}, \ldots, \dot{x}_{i_q}]$ per subsystem $i \in \mathcal{I}$. Figure 5.1 visualizes the recording of history stack data in PWA systems. For now, let the number of triplets q be fixed and the stack elements x_{i_j}, u_{i_j} and $\dot{x}_{i_j}$ be constant. In Section 5.2, this assumption will be discarded without loss of generality. Furthermore, it will be shown how to record new data for the history stacks and when to replace existing data with new data.

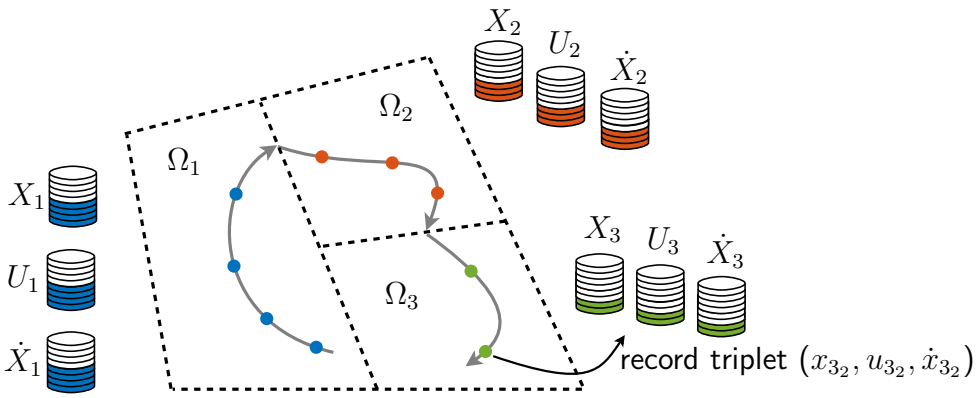


Fig. 5.1: Sampled triplets $(x(t), u(t), \dot{x}(t))$ along the state-space trajectory are stored in the history stacks $X_i, U_i, \dot{X}_i$ associated with the subsystems $i = 1, 2, 3$.

In practice, the state derivatives $\dot{x}_{i_j}$ are often not available for measurement. In such cases, concurrent learning relies on techniques such as fixed point smoothing [225, Chap. 9], [117] in order to obtain an accurate estimate. Section 5.2.1 provides a brief summary of fixed point smoothing. Two assumptions on the history stacks arise for concurrent learning to work well.

Assumption 5.1 (Linear independence) For each subsystem i, the history stacks X_i and U_i contain elements, such that there exist $n + p + 1$ linearly independent vectors $[x_{i_j}^\top, u_{i_j}^\top, 1]^\top$.

Assumption 5.1 ensures that the recorded data contains enough information for the identification task. It also implies the lower bound $n + p + 1 \leq q$ for the number of history stack elements. An upper bound is only given by practical considerations in form of limited, available memory. The existence of linearly independent history stack elements for Assumption 5.1 is guaranteed under PE. Note however that PE is by no means necessary for the collection of the history stack elements.

Assumption 5.2 (Correct derivatives) The elements of each history stack $\dot{X}_i$ fulfill $A_i x_{i_j} + B_i u_{i_j} + f_i = \dot{x}_{i_j}, \forall j \in \{1, \ldots, q\}$.

Assumption 5.2 requires the recorded derivative $\dot{x}_{i_j}$ to match the real derivative $\dot{x}$ at state x_{i_j} and input u_{i_j}. As will be shown in Theorem 5.3, one can relax Assumption 5.2 such that estimates of $\dot{x}$ obtained with fixed point smoothing are sufficiently accurate.

For the j-th triplet of subsystem i, define ε_{i_j} as the *error between estimated and recorded time derivative of the state*:

$$\varepsilon_{i_j}(t) := \hat{A}_i(t) x_{i_j} + \hat{B}_i(t) u_{i_j} + \hat{f}_i(t) - \dot{x}_{i_j}. \tag{5.3}$$

Note that the *history stack elements* x_{i_j}, u_{i_j} and $\dot{x}_{i_j}$ in (5.3) are constant and thus only the time varying estimates $\hat{A}_i$, $\hat{B}_i$ and $\hat{f}_i$ cause changes in ε_{i_j}.

In the ideal case, when Assumption 5.2 is satisfied and thus $\dot{x}_{i_j}$ is correctly measured, the error ε_{i_j} vanishes for the true system parameters:

$$\varepsilon_{i_j}(A_i, B_i, f_i) = A_i x_{i_j} + B_i u_{i_j} + f_i - \dot{x}_{i_j} = 0.$$

In the suboptimal case, when Assumption 5.2 is violated, the measured derivative differs from the true derivative. This is modelled by setting the measured derivative $\dot{x}_{i_j}$ equal to the true derivative $\dot{x}(x_{i_j}, u_{i_j})$ minus an *error term* $\delta_{i_j} \in \mathbb{R}^n$, i.e.,

$$\varepsilon_{i_j}(A_i, B_i, f_i) = A_i x_{i_j} + B_i u_{i_j} + f_i - \underbrace{\Big(\dot{x}(x_{i_j}, u_{i_j}) - \delta_{i_j}\Big)}_{\dot{x}_{i_j}} = \delta_{i_j}.$$

Next, expressing $\dot{x}_{i_j}$ in (5.3) in terms of the true system dynamics $\dot{x}(x_{i_j}, u_{i_j}) = A_i x_{i_j} + B_i u_{i_j} + f_i$ and the error δ_{i_j} yields

$$\begin{aligned}\varepsilon_{i_j} &= (\hat{A}_i - A_i)x_{i_j} + (\hat{B}_i - B_i)u_{i_j} + (\hat{f}_i - f_i) + \delta_{i_j} \\ &= \tilde{A}_i x_{i_j} + \tilde{B}_i u_{i_j} + \tilde{f}_i + \delta_{i_j}. \end{aligned} \tag{5.4}$$

While (5.3) specifies how to calculate ε_{i_j} online, the expression in (5.4) – which cannot be calculated due to the unknown parameter errors – is beneficial in the later convergence analysis. The errors ε_{i_j} form a key element in the recorded data-based part of the update laws (hence superscript $^{\mathrm{R}}$), which is given by

$$\dot{\hat{A}}_i^{\mathrm{R}} = -\Gamma_1 \sum_{j=1}^{q} \varepsilon_{i_j} x_{i_j}^\top, \qquad \dot{\hat{B}}_i^{\mathrm{R}} = -\Gamma_2 \sum_{j=1}^{q} \varepsilon_{i_j} u_{i_j}^\top, \qquad \dot{\hat{f}}_i^{\mathrm{R}} = -\Gamma_3 \sum_{j=1}^{q} \varepsilon_{i_j}. \tag{5.5}$$

The choice of update laws (5.5) is motivated by an exponential parameter convergence, which can be justified through Lyapunov arguments (which will be presented shortly).

As (5.5) does not depend on current measurements of u or x, these update laws can continuously update the parameter estimates independent of the switching signal σ. This way, one can achieve convergence even for currently inactive subsystems. Combining the *current data-based update laws* (superscript $^{\mathrm{C}}$) in (5.2) and the *recorded data-based update laws* (superscript $^{\mathrm{R}}$) in (5.5) yields the following theorem which specifies *concurrent learning adaptive identification of PWA systems*.

Theorem 5.3 (Parameter identifiers with concurrent learning) Consider the PWA system (5.1) with a known switching signal σ and let the subsystem states be predicted by (4.20) and (4.6). Then the update laws

$$\dot{\hat{A}}_i = \dot{\hat{A}}_i^{\mathrm{C}} + \dot{\hat{A}}_i^{\mathrm{R}}, \qquad \dot{\hat{B}}_i = \dot{\hat{B}}_i^{\mathrm{C}} + \dot{\hat{B}}_i^{\mathrm{R}}, \qquad \dot{\hat{f}}_i = \dot{\hat{f}}_i^{\mathrm{C}} + \dot{\hat{f}}_i^{\mathrm{R}} \tag{5.6}$$

cause the estimates $\hat{A}_i$, $\hat{B}_i$ and $\hat{f}_i$ to

- either converge exponentially to the true parameters A_i, B_i and f_i if Assumptions 5.1 and 5.2 are satisfied,
- or converge to a compact ball around the true parameters in case only Assumption 5.1 holds.

Proof: The proof of Theorem 5.3 is based on multiple Lyapunov functions, one per subsystem. The derivative of all Lyapunov functions is shown to be negative definite regardless of σ. The parameter estimates of subsystem i therefore converge towards the true parameters whether the i-th subsystem is active ($\sigma = i$) or inactive ($\sigma \neq i$). It is therefore possible to analyze the stability of each subsystem individually, which is done in the following by neglecting the index i (i.e. $A \triangleq A_i$ or $x_j \triangleq x_{i_j}$).

As for the parameter identifiers in the previous chapter, the parameter error vector $\tilde{\theta} = \mathrm{vec}([\tilde{A}, \tilde{B}, \tilde{f}])$ enables a compact representation of the concurrent learning-based updated laws. First, rewrite (5.4) as

$$\varepsilon_j = \left(\begin{bmatrix} x_j \\ u_j \\ 1 \end{bmatrix} \otimes I_n \right)^\top \tilde{\theta} + \delta_j. \tag{5.7}$$

Then, recall the definitions of Γ and Ψ in (4.25) as well as the compact representation for $\dot{\tilde{x}}$ in (4.26):

$$\dot{\tilde{x}} = A_\mathrm{m}\tilde{x} + \Psi^\top \tilde{\theta}. \tag{5.8}$$

Additionally, define

$$\Xi := \left(\sum_{j=1}^{q} \begin{bmatrix} x_j \\ u_j \\ 1 \end{bmatrix} \begin{bmatrix} x_j^\top & u_j^\top & 1 \end{bmatrix} \right) \otimes I_n. \tag{5.9}$$

With a few transformations involving (5.7) and (5.9), the update laws in (5.6) take the form

$$\dot{\tilde{\theta}} = \dot{\hat{\theta}} = -\Gamma\Psi P\tilde{x} - \Gamma\Xi\tilde{\theta} - \Gamma \underbrace{\left(\sum_{j=1}^{q} \begin{bmatrix} x_j \\ u_j \\ 1 \end{bmatrix} \otimes \delta_j \right)}_{\Delta}. \tag{5.10}$$

For stability analysis, consider the same quadratic candidate Lyapunov function (4.28) as before. The time derivative of V along (5.8) and (5.10) yields:

$$\begin{aligned} \dot{V} &= \tilde{x}^\top A_\mathrm{m}^\top P\tilde{x} + \tilde{x}^\top P A_\mathrm{m}\tilde{x} + \tilde{\theta}^\top \Psi P\tilde{x} + \tilde{x}^\top P\Psi^\top \tilde{\theta} \\ &\quad - \left(\tilde{x}^\top P^\top \Psi^\top \Gamma^\top + \tilde{\theta}^\top \Xi^\top \Gamma^\top + \Delta^\top \Gamma^\top \right) \Gamma^{-1} \tilde{\theta} \\ &\quad - \tilde{\theta}^\top \Gamma^{-1} \left(\Gamma\Psi P\tilde{x} + \Gamma\Xi\tilde{\theta} + \Gamma\Delta \right) \\ &= \tilde{x}^\top (A_\mathrm{m}^\top P + P A_\mathrm{m})\tilde{x} - \tilde{\theta}^\top (\Xi^\top + \Xi)\tilde{\theta} - 2\Delta^\top \tilde{\theta} \\ &= -\tilde{x}^\top Q_{\tilde{x}} \tilde{x} - \tilde{\theta}^\top Q_\theta \tilde{\theta} - 2\Delta^\top \tilde{\theta}, \end{aligned} \tag{5.11}$$

where $Q_{\tilde{x}} := -(A_\mathrm{m}^\top P + P A_\mathrm{m})$ and $Q_\theta := \Xi^\top + \Xi$. Recall that $Q_{\tilde{x}}$ was chosen to be positive definite in the design of the current data-based update laws (5.2). Also, because of Assumption 5.1, the matrix Ξ and thus Q_θ are positive definite. Note that the desirable

property $-\tilde{\theta}^\top Q_\theta \tilde{\theta} < 0, \forall \tilde{\theta} \neq 0$ results from the recorded data-based update laws (5.5) and thus constitutes the central motivation behind the choice of (5.5). Assuming the ideal case in (5.11), i.e. $\delta_j = 0, \forall j \in \{1, \ldots, q\}$, then the Δ-term vanishes and the derivative of the candidate Lyapunov function is indeed negative definite. Hence, all estimation errors $\tilde{x}$ and parameter errors $\tilde{\theta}$ converge exponentially to zero.

For the suboptimal case, however, in which $\delta_j \neq 0$, the derivative (5.11) consists of quadratic terms in $\tilde{x}$ and $\tilde{\theta}$ but is also linear in $\tilde{\theta}$. This observation permits a division of the error space $[\tilde{x}^\top,\ \tilde{\theta}^\top]^\top \in \mathbb{R}^{n(n+p+2)}$ in two sets as shown exemplarily in Fig. 5.2. For some small values of $\tilde{\theta}$ the linear term outweighs the quadratic terms. This results in a first closed set for which $\dot{V} > 0$. The first set is surrounded by a second set in which the quadratic terms outweigh the linear term and $\dot{V} < 0$. The outer set with $\dot{V} < 0$ guarantees boundedness of $\tilde{x}$ and $\tilde{\theta}$.

We can approximate the compact set to which the parameter errors $\tilde{\theta}$ converge by analyzing the shape of the boundary between the two sets. On this boundary $\dot{V} = 0$, or equivalently

$$\begin{bmatrix}\tilde{x}^\top & \tilde{\theta}^\top\end{bmatrix} \begin{bmatrix} Q_{\tilde{x}} & 0 \\ 0 & Q_\theta \end{bmatrix} \begin{bmatrix}\tilde{x} \\ \tilde{\theta}\end{bmatrix} + \begin{bmatrix}0 & 2\Delta^\top\end{bmatrix} \begin{bmatrix}\tilde{x} \\ \tilde{\theta}\end{bmatrix} = 0. \tag{5.12}$$

It is easy to verify that both the origin and the state with $\tilde{x} = 0$ and $\tilde{\theta} = -2(Q_\theta^{-1})^\top \Delta$ reside on this boundary. Furthermore, (5.12) resembles an ellipsoid in the error space centered at $\tilde{x}_c = 0$ and $\tilde{\theta}_c = -(Q_\theta^{-1})^\top \Delta$. An alternative formulation of this ellipsoid is

$$\begin{bmatrix}\tilde{x}^\top - \tilde{x}_c^\top & \tilde{\theta}^\top - \tilde{\theta}_c^\top\end{bmatrix} \underbrace{\frac{1}{\Delta^\top Q_\theta^{-1} \Delta} \begin{bmatrix} Q_{\tilde{x}} & 0 \\ 0 & Q_\theta \end{bmatrix}}_{=:E} \begin{bmatrix}\tilde{x} - \tilde{x}_c \\ \tilde{\theta} - \tilde{\theta}_c\end{bmatrix} = 1.$$

The eigenvectors of the positive definite matrix E define the principal axes of the ellipsoid. The length of the semi-axes is given by the square root of the inverse eigenvalues of E. Figure 5.2 visualizes the ellipsoid for a two-dimensional parameter vector $\theta = [\theta_1, \theta_2]^\top$.

In a last step, exploiting the triangle inequality on the centroid of the ellipsoid and its maximal semi-principal axis (characterized by the minimal eigenvalue) gives an approximation of the ball around the true parameters to which the estimates converge in the presence of errors δ_j:

$$\|\tilde{\theta}^*_{\max}\| \leq \|\Delta^\top Q_\theta^{-1}\| + \left(\lambda_{\min}\left(\frac{Q_\theta}{\Delta^\top Q_\theta^{-1}\Delta}\right)\right)^{-\frac{1}{2}},$$

where $\|\cdot\|$ is the Euclidean norm and $\lambda_{\min}(\cdot)$ denotes the minimum eigenvalue. Compare with Fig. 5.2 for a visual interpretation of this approximation. □

While the discussion of concurrent learning-based parameter identifiers thus far involved only PWA systems, it needs to be stressed at this point that the results also apply for PWL systems. In PWL systems, there is no affine input vector f. Consequently, deleting the terms related to f yields the special case of PWL systems. Importantly, this removes the constant 1 in Assumption 5.1 and in the definitions of Ψ and Ξ.

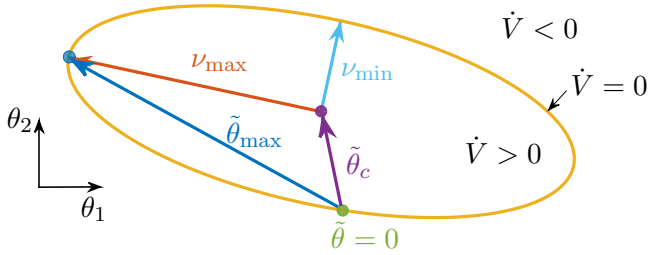


Fig. 5.2: Visualization of the ellipsoidal set for which $\dot{V} \geq 0$ and approximation of the maximum parameter error $\|\tilde{\theta}_{\max}\|$ in terms of the centroid $\tilde{\theta}_c$ and the maximal semi-principal axis $\nu_{\max}$ of the ellipsoid ($\|\nu_{\max}\| = 1/\sqrt{\lambda_{\min}}$).

From an adaptive control or identification point of view, the above proof is remarkable in a sense that convergence is guaranteed without the usual PE requirement. The linear independence of the history stack elements (required by Assumption 5.1) replaces PE. This is beneficial as linear independence is considerably easier to monitor than PE. Furthermore, excitation over a limited period of time suffices to record linear independent data. Overall the success of parameter identifiers with concurrent learning heavily relies on the history stack data, which will be discussed in greater detail now.

5.2 History Stack Management in Concurrent Learning

In the discussion so far, the stack elements were assumed to be constant. Looking back at (5.11), however, reveals that parameter convergence is maintained as long as Q_θ remains positive definite. In other words, as long as Q_θ is ensured to be positive definite, one can arbitrarily replace history stack elements. From a control theoretic perspective, this resembles switching between different stable dynamics that share the common Lyapunov function V. In the following steps, this insight is exploit in order to alter history stack data and further improve the performance of concurrent learning-based adaptive estimation. First, we give a brief introduction to fixed point smoothing, which is used to estimate the state derivatives stored in history stacks.

5.2.1 Fixed Point Smoothing

Fixed point smoothing is a discrete-time filtering technique. It can be used to estimate a partially measurable state at time instance j based on measurements before and after j. Jazwinski gave the following motivation for fixed point smoothing [117, p. 143]: "In a post mortem (after the fact) analysis, it is possible to wait for more observations to accumulate. In that case, the estimate can be improved by smoothing."

In the setting of concurrent learning, the extended state vector that is to be estimated at time instance j by fixed point smoothing is $w = [x^\top, \dot{x}^\top]^\top$. It is assumed that only x is available for measurement. The corresponding discrete-time state-space system with

sampling time dt is approximated by

$$w(k+1) = \underbrace{\exp\begin{bmatrix} 0_n & I_n \\ 0_n & 0_n \end{bmatrix}_{dt}}_{=:F_k} w(k) + v_w(k), \qquad x(k) = \underbrace{\begin{bmatrix} I_n & 0_n \end{bmatrix}}_{=:H_k} w(k) + v_x(k), \tag{5.13}$$

where $0_n \in \mathbb{R}^{n\times n}$ is a zero matrix. The process noise and measurement noise are characterized by $v_w \sim \mathcal{N}(0, Q)$ and $v_x \sim \mathcal{N}(0, R)$, respectively. Fixed point smoothing, as presented in [225, Chap. 9], constitutes an extension of the Kalman filter (KF). Smoothing is carried out in two steps. First a modified forward KF is applied on measurements taken up to the time instance j at which the extended state w is to be smoothed. That means the following KF equations are applied for $k = 0, 1, \ldots, j-1$:

$$\begin{aligned} L_k &= F_k P_k^- H_k^T \left(H_k P_k^- H_k^T + R_k\right)^{-1}, \\ P_{k+1}^- &= F_k P_k^- \left(F_k - L_k H_k\right)^T + Q_k, \\ \hat{w}_{k+1}^- &= F_k \hat{w}_k^- + L_k \left(x_k - H_k \hat{w}_k^-\right), \end{aligned}$$

where the index specifies the time instance. Once j is reached ($k = j$), the second step of fixed point smoothing begins. A backwards working KF is initialized with estimates from the forward KF, i.e., $\Sigma_j = P_j, \Pi_j = P_j, \hat{w}_{j|j} = \hat{w}_j^-$, and executed for $k = j, j+1, \ldots l$. The backwards KF equations are:

$$\begin{aligned} L_k &= F_k P_k H_k^T \left(H_k P_k H_k^T + R_k\right)^{-1}, \\ \Lambda_k &= \Sigma_k H_k^T \left(H_k P_k H_k^T + R_k\right)^{-1}, \\ \hat{w}_{j|k+1} &= \hat{w}_{j|k} + \Lambda_k \left(x_k - H_k \hat{w}_k^-\right), \\ \hat{w}_{k+1}^- &= F_k \hat{w}_k^- + L_k \left(x_k - H_k \hat{w}_k^-\right), \\ P_{k+1} &= F_k P_k \left(F_k - L_k H_k\right)^T + Q_k, \\ \Pi_{k+1} &= \Pi_k - \Sigma_k H_k^T \Lambda_k^T, \\ \Sigma_{k+1} &= \Sigma_k \left(F_k - L_k H_k\right)^T. \end{aligned}$$

Every measurement after time instance j thus improves the accuracy of the estimate $\hat{w}_{j|l}$ of w based on measurements up to time instance l. For more information about fixed point smoothing the reader is encouraged to refer to [225].

For the presented setting of concurrent learning, the forward KF continuously estimates w and hence also $\dot{x}$. Whenever suitable data for the history stack is measured (e.g. $x(k), u(k)$ enable faster convergence), a new backward KF is initialized in order to smooth the desired state estimate based on the next $l - k$ measurements. This results in a more precise estimate $\hat{\dot{x}}(k)$ available at time instant l.

5.2.2 Populating the History Stacks

Since initially all history stacks are empty and Assumption 5.1 is violated, consider first the process of populating the history stacks. The history stacks need to be filled with at

least $n+p+1$ linearly independent elements. Also, a new data point should be sufficiently different from the previously recorded data point. Let $\Phi(t) := [x^\top(t),\ u^\top(t)]^\top$ be the state-input vector at time instance t and let Φ_{prev} denote the last recorded state-input vector. In concurrent learning, it is common to refer to a new measurement $\Phi(t)$ as sufficiently different if the following inequality

$$\frac{\|\Phi(t) - \Phi_{\text{prev}}\|^2}{\|\Phi(t)\|} \geq \upsilon \tag{5.14}$$

is satisfied, where the threshold $\upsilon \in \mathbb{R}^+$ is a design parameter. Until the history stack fulfills Assumption 5.1, a new measurement is added to the history stack if x and u are linearly independent of all previously recorded elements and if they fulfill (5.14).

5.2.3 Maximizing the Rate of Convergence

In order to confine computational complexity, the number of recorded data elements q per history stack is limited. Once a history stack is filled with q data points, it has to be reasoned whether replacing old data with new data is beneficial or not.

With the derivative of the Lyapunov function in (5.11), we are able to characterize the rate of convergence. Note that in the ideal case ($\delta_{i_j} = 0$), the derivative of the Lyapunov function of an inactive subsystems ($\tilde{x} = 0$) is bounded by

$$\dot{V} = -\tilde{\theta}^\top Q_\theta \tilde{\theta} \leq -\lambda_{\min}(Q_\theta)\tilde{\theta}^\top\tilde{\theta}. \tag{5.15}$$

It is thus intuitive to record stack elements x_j and u_j such that the minimum eigenvalue of Q_θ is maximized, as this maximizes the rate of convergence. Especially for full history stacks, this approach allows to reason whether or not it is beneficial to replace an existing stack element with a new measurement. While the same idea is also applied in concurrent learning adaptive identification [274], it was Chowdhary *et al.* who originally introduced it to concurrent learning adaptive control [54].

Note that the maximization of the convergence rate also affects the centroid $\tilde{\theta}_c$, which characterizes the remaining parameter error. Approximating the squared norm of $\tilde{\theta}_c$ as

$$\|\tilde{\theta}_c\|^2 = \Delta^\top Q_\theta^{-1}(Q_\theta^{-1})^\top\Delta \leq \lambda^2_{\max}(Q_\theta^{-1})\|\Delta\|^2 = \frac{1}{\lambda^2_{\min}(Q_\theta)}\|\Delta\|^2,$$

reveals that improving the convergence rate by maximizing $\lambda_{\min}(Q_\theta)$ provides a second benefit. That is, with increased $\lambda_{\min}(Q_\theta)$, the error-term Δ – caused by errors δ_j in the estimation of $\dot{x}(x_j, u_j)$ – has a weakened effect on the parameter errors.

5.2.4 History Stack Purging

Theorem 5.3 showed that wrongly recorded or estimated stack elements hinder convergence to the true parameters. This constitutes a major limitations in concurrent learning but has received only limited attention in previous works on concurrent learning, excluding [63]. In turn, concurrent learning is expected to work well in case system parameters remain

the same for all times. However, if system parameters change over time (e.g. due to aging or wear), then history stack elements become outdated and estimates converge to (or are stuck at) wrong values. In order to overcome the problem of outdated measurements, the history stacks need to be cleared from time to time. In the literature, this process is referred to as purging.

The *purging algorithm* proposed in [63] replaces entire history stacks at once. The general idea is to keep one set of fixed history stacks $X_{\text{use}}, U_{\text{use}}, \dot{X}_{\text{use}}$ per subsystem in use. At the same time, a set of new history stacks $X_{\text{new}}, U_{\text{new}}, \dot{X}_{\text{new}}$ is filled in the background according to the population and managing strategies discussed above. It then has to be reasoned at what time purging is performed, i.e. when to replace the used stacks by new stacks. This decision is again based on the performance characteristics $\lambda_{\min}(Q_{\theta,\text{use}})$ and $\lambda_{\min}(Q_{\theta,\text{new}})$ obtained for the two history stacks. Let the aging of the used stacks be modeled by an exponential decay of $\lambda_{\min}(Q_{\theta,\text{use}})$ and perform purging when the new stacks yield better performance than the decaying stacks. The corresponding condition is formally stated as

$$\lambda_{\min}(Q_{\theta,\text{new}}) < c_1 e^{-c_2(t-\bar{t})}\lambda_{\min}(Q_{\theta,\text{use}}) + c_3, \tag{5.16}$$

where $c_1, c_2, c_3 > 0$ are design parameters and $\bar{t}$ is the time at which the last purging was performed.

Note that history stack purging according to (5.16) enables the removal of outdated measurements and thus restores the tracking ability of concurrent learning-based algorithms. As it replaces whole history stacks at once, however, such algorithms are unable to remove individual erroneous stack elements. Hence, if the recording is prone to outliers, which is certainly the case in fixed point smoothing and switched systems (as we consider them here), each new history stack is still likely to contain outliers.

One research direction in concurrent learning is therefore to combine the purging idea with dynamically generated stack measurements based on the current parameter estimates [125]. Under certain dwell-time assumptions such algorithms produce more precise parameter estimates after each purging which in turn yield improved stack measurements and so on. In the following, a different approach is presented that allows to identify individual erroneous stack elements and replace them in order to obtain better parameter estimates and restore tracking.

5.3 Removal of Erroneous History Stack Elements

This subsection describes the first algorithm to detect and remove erroneous stack elements online, which we previously presented in [278, 281]. The algorithm is based on the finding presented in the proof of Theorem 5.3, i.e., the estimated parameters converge to an ellipsoidal set in the parameter space (depicted in Fig. 5.2) in the presence of erroneous history stack elements.

For the case of erroneous stack elements, the prediction error will not converge to zero but remain $\tilde{x} \neq 0$ or greater than some threshold in the presence of noise. Erroneous stack elements can then be detected by comparing the individual contributions in the update

laws (5.6). To do this, define the contribution of the current measurements as

$$\dot{\hat{\theta}}^C(x(t), u(t)) = -\Gamma \Psi P \tilde{x} \tag{5.17}$$

and the contribution of the j-th history stack triplet as

$$\dot{\hat{\theta}}_j^R(x_j, u_j, \dot{x}_j) = -\Gamma \Xi_j \hat{\theta} + \Gamma \xi_j, \quad j \in \{1, \dots, q\}, \tag{5.18}$$

where

$$\Xi_j := \left(\begin{bmatrix} x_j \\ u_j \\ 1 \end{bmatrix} \begin{bmatrix} x_j^\top & u_j^\top & 1 \end{bmatrix} \right) \otimes I_n, \qquad \xi_j := \begin{bmatrix} x_j \\ u_j \\ 1 \end{bmatrix} \otimes \dot{x}_j.$$

The directional information in the update vectors (5.17) and (5.18) is of particular interest. Recall that the update law (5.17) is only based on current data and according to Theorem 4.5 drives the parameter estimates to the true parameters under the assumption of persistent excitation. Therefore, $\dot{\hat{\theta}}^C$ serves as a reference point.

Note that the PE requirement should not be understood as a limitation at this point. Once the history stack is known to contain erroneous elements, one must in any case record new measurements. In order to fulfill Assumption 5.1 excitation of the system is beneficial. During the collection of new measurements, it can be reasoned which elements are erroneous and need to be replaced. While the existence of measurements which satisfying Assumption 5.1 is guaranteed under PE, it is not a necessary condition. Future work could analyze how the following ideas perform without PE. As was shown in Lemma 4.4, PE can be ensured by choosing input signals sufficiently rich of order $n + 1$ with distinct frequencies.

In order to analyze to which extent the recorded data-based update laws (5.18) coincide with the reference vector $\dot{\hat{\theta}}^C$, define for each stack element j the angle φ_j between $\dot{\hat{\theta}}^C$ and $\dot{\hat{\theta}}_j^R$:

$$\varphi_j(t) = \sphericalangle \left(\dot{\hat{\theta}}^C(t),\ \dot{\hat{\theta}}_j^R(t) \right). \tag{5.19}$$

Note that statements, which are based on the angles φ_j, are only meaningful if the norms of the corresponding vectors are sufficiently large. Therefore, two thresholds ϑ^C and ϑ^R for $\dot{\hat{\theta}}^C$ and $\dot{\hat{\theta}}^R$ are introduced to formalize this conditions. If $\|\dot{\hat{\theta}}^C\| < \vartheta^C$, the current data-based update vector fails to qualify as a reliable reference point. This is for instance the case if the parameter estimates are in the vicinity of the true parameters. For $\|\dot{\hat{\theta}}_j^R\| < \vartheta^R$, the contribution of the j-th history stack element is very small. In this case, the j-th history stack triplet $(x_j, u_j, \dot{x}_j)$ cannot be falsified by the current estimates $\hat{\theta}$. In summary, falling below these two thresholds corresponds to good estimates. The angle φ_j is therefore set to zero whenever one of the two thresholds is not exceeded:

$$\varphi_j(t) \Leftarrow 0, \quad \text{if } \left(\|\dot{\hat{\theta}}^C\| < \vartheta^C \text{ or } \|\dot{\hat{\theta}}_j^R\| < \vartheta^R \right). \tag{5.20}$$

The parameter estimates $\hat{\theta}$ and angles φ_j are time-varying for sinusoidal excitation and enter into a limit cycle. Due to the known excitation u, the frequency of this limit cycle

can be approximated. Since we are interested in the update directions on average, we consider the low-pass filtered angles $\bar{\varphi}_j$, where the cut-off frequency of the filter is lower than the lowest excitation frequency in u. The following algorithm to detect and remove erroneous history stack elements is based on the filtered angles $\bar{\varphi}_j$.

Algorithm 5.4 (Detection of erroneous stack elements) Consider the concurrent learning-based update law in Theorem 5.3 with errors δ_j. Monitor the filtered angles $\bar{\varphi}_j$ according to (5.19) and (5.20). If the parameter estimates enter a limit cycle (i.e. $\tilde{x} \neq 0$ and $\dot{\bar{\varphi}}_j \approx 0$), replace the k-th history stack triplet, where

$$k = \arg\max_j \left\{ \bar{\varphi}_j \middle| \bar{\varphi}_j \geq \frac{\pi}{2} \right\}, \tag{5.21}$$

by a new measurement with $\bar{\varphi}_{\text{new}} < \bar{\varphi}_k$ in order to reduce the errors in the history stack.

Algorithm 5.4 is further justified by the following considerations. First, note that each triplet in the history stack spans a subspace in the parameter space. The subspace Θ_j associated with the j-th triplet contains all parameter configurations that cannot be falsified by the measurements $(x_j, u_j, \dot{x}_j)$:

$$\Theta_j = \left\{ \hat{\theta} \;\middle|\; \hat{A}(\hat{\theta})x_j + \hat{B}(\hat{\theta})u_j + \hat{f}(\hat{\theta}) = \dot{x}_j \right\}. \tag{5.22}$$

The parameter configurations $\hat{\theta} \in \Theta_j$ form the solution of an under-determined system of linear equations, from which the dimension of Θ_j can be derived to be $\dim(\Theta_j) = \dim(\theta) - n = n(n+p)$.

In the error-less case ($\delta_j = 0, \forall j$) and under Assumption 5.1, all subspaces intersect at a single point $\theta = \cup_{j=1}^{q} \Theta_j$, for which $\tilde{\theta} = 0$. In case $\delta_j \neq 0$, however, the subspaces likely do not intersect at a single point. Figure 5.3 displays four exemplary subspaces. The three green lines represent the subspaces of error-less history stack triplets. The red line corresponds to an erroneous history stack element and is therefore shifted away from the intersection point. Due to the shifted subspace Θ_4, the estimates do not converge to the true parameter configuration θ. Instead the estimates enter a limit cycle in the ellipsoidal set as shown in the proof of Theorem 5.3.

Next, note that the recorded data-based update laws (5.18) correspond to an orthogonal projection onto the subspaces Θ_j. For correct history stack triplets, this projection points into the ellipsoid. Also the current data-based update (5.17) generally points towards the true parameter configuration. Convergence is prevented if at least one erroneous stack element exists and points out of the ellipsoid. Therefore, the averaged angle of at least one erroneous history stack element is greater than $\pi/2$. This insight affirms Algorithm 5.4.

In case of multiple erroneous history stack elements, Algorithm 5.4 removes them sequentially. The geometric considerations above only allow to falsify one erroneous history stack element at a time. Consider for instance the case in which a second erroneous subspace is located below the ellipsoid in Fig. 5.3.

Since PE constitutes a critical assumption in adaptive systems, we complete the theoretical part of this chapter by briefly repeating the role of PE in parameter identifiers

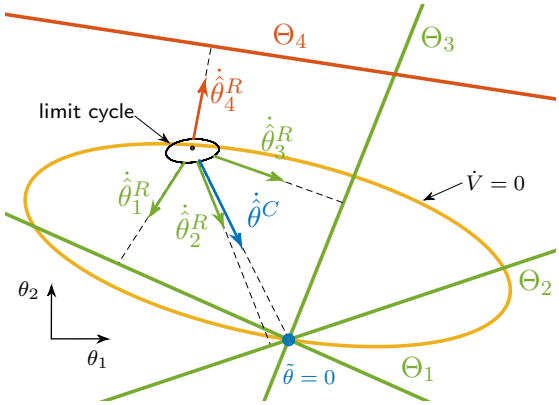


Fig. 5.3: Exemplary limit cycle in a two dimensional parameter space. The History stack has 4 elements for which only $\delta_4 \neq 0$. The update vectors $\dot{\hat{\theta}}_j^R$ are an orthogonal projection onto Θ_j.

with and without concurrent learning. For the parameter identifiers without concurrent learning discussed in Chapter 4, PE is a necessary condition for parameter convergence. The concurrent learning-based parameter identifiers in Theorem 5.3 replace PE by linear independence of the history stack elements. While PE in this context guarantees the existence of linearly independent measurements, PE is by no means necessary in order to obtain them. In case of an erroneous history stack element, we propose in Algorithm 5.4 to resume PE over a limited period of time in order to detect and replace the erroneous element. Once the erroneous element is replaced, convergence can proceed without PE.

5.4 Simulation Studies

In this section, the proposed concurrent learning-based parameter identifiers for PWA systems and Algorithm 5.4 for the detection of erroneous history stack elements in concurrent learning are numerically tested. First, the advantage of concurrent learning over traditional adaptation is demonstrated for a piecewise linear bimodal system. Then, we analyze the estimation and tracking of time-varying parameters with concurrent learning-based parameter identifiers. Afterwards, we compare the Algorithm 5.4 proposed in Section 5.3 with standard Recursive Least Squares (RLS) and the state-of-the-art purging approach discussed in Section 5.2.4. Finally, the concurrent learning-based parameter identifiers are applied for the adaptive identification of a hybrid two-tank system.

5.4.1 Benefit of Concurrent Learning

First, consider the bimodal piecewise linear system with $n = 1$, $p = 1$, and subsystem parameters

$$\theta_1 = \begin{bmatrix} -2 & 1 \end{bmatrix}^\top \qquad \Omega_1 = \{x \in \mathbb{R} | x \geq 0\}$$
$$\theta_2 = \begin{bmatrix} -5 & 2 \end{bmatrix}^\top \qquad \Omega_2 = \{x \in \mathbb{R} | x < 0\}.$$

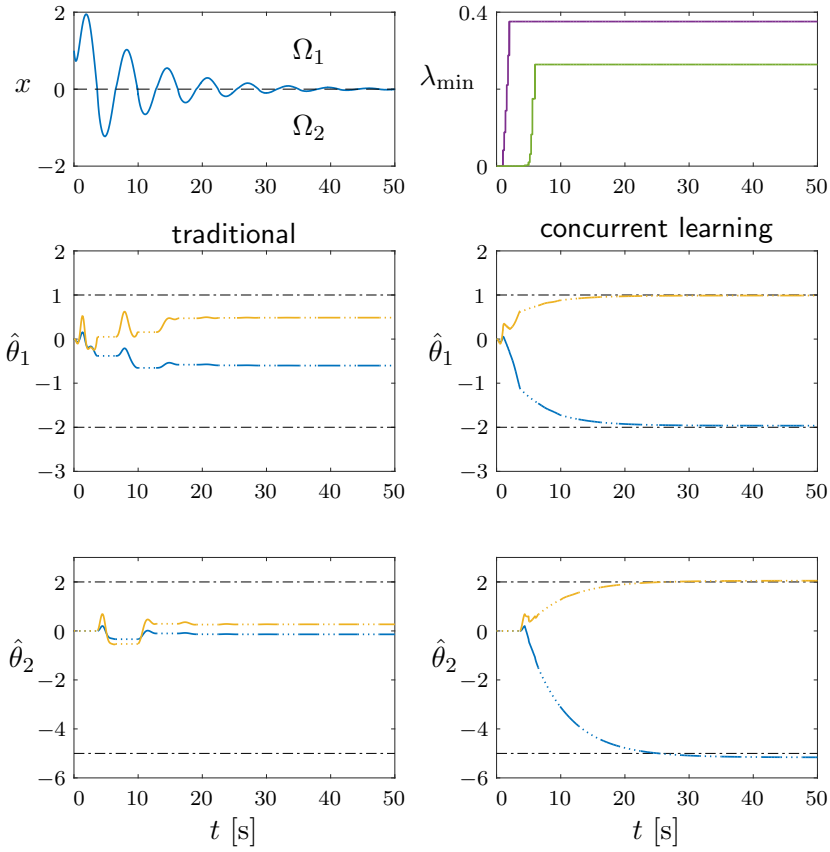


Fig. 5.4: Comparison of parameter identifiers without (left column) and with (right column) concurrent learning. The top left shows the exponentially decaying excitation of the state in both cases. The top right shows the minimum Eigenvalues associated with the two history stacks for the concurrent learning-based estimation.

The initial state is $x = 1$ and the control input is the exponentially decaying sinusoidal signal $u(t) = 5\mathrm{e}^{-0.1t}\sin(t)$. Figure 5.4 shows the resulting evolution of the system state in the top left plot. The figure also shows the estimated parameters $\hat{\theta}_1$ and $\hat{\theta}_2$ obtained with the traditional updated laws (4.7) in the left column and concurrent learning-based update laws (5.6) in the right column. The plot at the top of the right column additionally visualizes the minimum eigenvalues $\lambda_{\min}(Q_\theta)$ in the concurrent learning framework for the two subsystems. Simulation results were obtained with design parameters $-A_{\mathrm{m}} = P = \Gamma_1 = \Gamma_2 = 1$ and state derivatives estimated by fixed point smoothing according to Section 5.2.1.

The exponentially decaying state does not yield a persistent excitation of the system. Hence, the estimates do not converge under the traditional update laws. For concurrent learning, the initial excitation is sufficient to fill the history stacks with linear independent data, which can be seen from the increase in $\lambda_{\min}$. Based on the recorded data, exponential convergence is achieved for both active (solid lines) and inactive (dotted lines) subsystems. This comparison demonstrates the benefit of concurrent learning in terms of reduced PE requirements and increased convergence rates.

5.4.2 Parameter Estimation and Tracking

Next, consider a PWA system with three subsystems and dimensions $n = 2$, $p = 1$. In order to test the tracking abilities of the proposed algorithms, the subsystems experience a sudden change in its parameters after 700 s. On the first time interval $t \in [0, 700)$s, the unknown parameter vectors are

$$\begin{aligned}\theta_1' &= \begin{bmatrix}0 & -2 & 1 & -3 & 0 & 1 & 0 & 0\end{bmatrix}^\top \\ \theta_2' &= \begin{bmatrix}0 & -2 & 1 & -4 & 0 & 1 & 0 & 0\end{bmatrix}^\top \\ \theta_3' &= \begin{bmatrix}0 & -2 & 1 & -2 & 0 & 1 & 0 & 0\end{bmatrix}^\top\end{aligned}$$

and during the second interval $t \in [700, 1400)$s, the parameters are

$$\begin{aligned}\theta_1'' &= \begin{bmatrix}0 & -3 & 1 & -1 & 0 & 1.3 & 0 & 0.3\end{bmatrix}^\top \\ \theta_2'' &= \begin{bmatrix}0 & -2 & 1 & -2 & 0 & 1.5 & 0 & 0.4\end{bmatrix}^\top \\ \theta_3'' &= \begin{bmatrix}0 & -3 & 1 & -2 & 0 & 1.7 & 0 & -0.2\end{bmatrix}^\top.\end{aligned}$$

The known state-space partitions remain constant throughout the simulation and are given by

$$\begin{aligned}\Omega_1 &= \left\{x \in \mathbb{R}^2 |\ -3 \leq x_1 \leq 3\right\} \\ \Omega_2 &= \left\{x \in \mathbb{R}^2 |\ x_1 < -3\right\} \\ \Omega_3 &= \left\{x \in \mathbb{R}^2 |\ x_1 > 3\right\}.\end{aligned}$$

The system is excited by the input signal $u(t) = \sin(0.8t) + \sin(t) + \sin(3t) + \bar{u}(t)$, where $\bar{u}(t)$ is a ramp-like offset shown in Fig. 5.5, which drives the system into all three regions. Therefore, the input signal u is sufficiently rich of order 6, which enables traditional updating according to Theorem 4.5 as well as recording linearly independent data needed to satisfy the assumptions in Theorem 5.3. Gaussian white noise with zero mean and 10^{-3} variance is added to the state measurement x.

The design parameters are chosen as $P = I_2$ and $A_\mathrm{m} = -10I_2$, and $\Gamma_1 = \Gamma_2 = \Gamma_3 = 1$ for balanced adaptation of A_i, B_i and f_i. All parameter estimates $\hat{\theta}_i$ are initially zero and the history stacks of size $q = 4$ are initially empty. Selected estimates of the state derivative for the history stacks are obtained with fixed point smoothing according to Section 5.2.1 with $dt = 0.005\,\mathrm{s}$, $R = 10^{-4}I_2$ and $Q = \mathrm{diag}(0, 0, 0.005, 0.005)$. The design parameters for the detection of erroneous history stack elements are $\vartheta^C = 0.09$, $\vartheta^R(t) = 0.1 \max_j(\|\dot{\hat{\theta}}_j^R(t)\|)$.

Figure 5.6 shows the evolution of estimates $\hat{\theta}_i$ for all subsystems $i = 1, 2, 3$ on the entire time interval $[0, 1400]$s. The figure shows that once a subsystem is activated for the first time, its history stacks are quickly filled with measurements and the parameter estimates converge to the true parameters. Note that the transition between two regions is characterized by multiple switches due to the sinusoidal excitation. These switches likely cause the smoothing process to result in erroneous history stack elements. Especially the estimates $\hat{\theta}_3$ converge to wrong parameters between 400 s and 450 s. The proposed detection

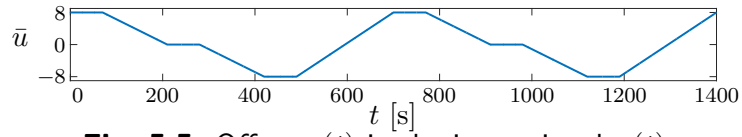


Fig. 5.5: Offset $\bar{u}(t)$ in the input signal $u(t)$.

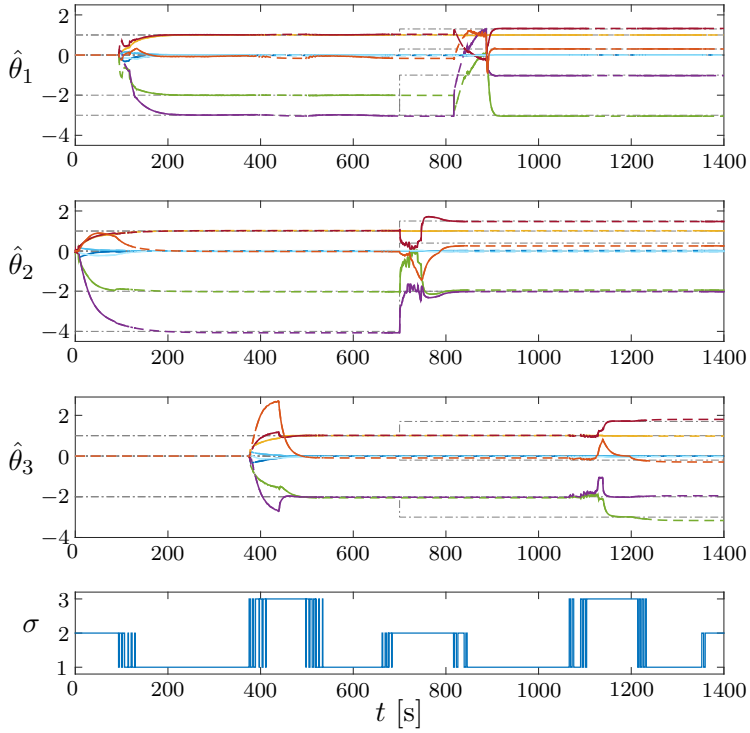


Fig. 5.6: Identification and tracking of PWA subsystem parameters with concurrent learning-based parameter identifiers and removal of erroneous history stack elements.

algorithm however detects and removes these erroneous elements once the system remains in Ω_3 for a sufficiently long time.

The evolution of $\hat{\theta}_2$ after the parameter change at 700 s shows that the system enters into a limit cycle due to the parameter mismatch. This can be seen by oscillating estimates, which naturally triggers the proposed detection algorithm. In turn, outdated history stack elements taken before the change are successfully replaced. As can also be seen from the figure, the removal of erroneous history stack elements only takes place for the active subsystem, which is why $\hat{\theta}_3$ reacts to the parameter change with a delay of 400 s. Approximately 100 s after a subsystem is activated, the corresponding history stacks do not contain outdated measurements taken with θ_i' any more and the estimates converge to the true parameters θ_i''.

Figure 5.7 shows the normed parameter errors $\|\tilde{\theta}_i\|$ for all subsystems. At around 400 s and 900 s it can be seen that the sequential removal of erroneous history stack elements according to Algorithm 5.4 might lead to a temporary increase in the parameter errors. As shown in the proof of Theorem 5.3, however, all errors remain bounded.

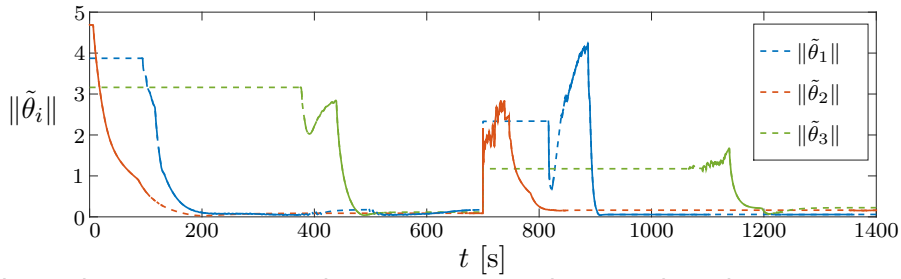


Fig. 5.7: Normed parameter errors during concurrent learning-based parameter estimation with removal of erroneous history stack elements. The lines are dashed for inactive and solid for active subsystems.

5.4.3 Comparison with Cyclic Purging and Recursive Least Squares

Reducing the parameter error $\tilde{\theta}$ in the presence of errors δ_j constitutes a major advantage of the proposed algorithm over other approaches. In the following simulation of an affine LTI system, we demonstrate this by comparing Algorithm 5.4 to the cyclic purging approach proposed in [63] and revised in Section 5.2.4 as well as standard RLS [152, Chap. 11]. For the purging approach, let the aging of data be modeled by the exponential decay (5.16) with design parameters $c_1 = c_3 = 0$ and $c_2 = 0.01$. The standard RLS algorithm is implemented on the history stack measurements x_j, u_j and $\dot{x}_j$ with a forgetting factor $\gamma = 0.95$.

A stable, randomly chosen state-space system of higher order ($n = 5, p = 2$) is considered, which means that $\theta \in \mathbb{R}^{40}$. The system is excited with the two input signals $u_1(t) = \sin(\frac{5}{3}t) + \cos(1.1t) + \sin(1.7t)$ and $u_2(t) = \sin(\frac{4}{3}t) + \cos(t) + \sin(\frac{7}{3}t)$, each sufficiently rich of order 6 with distinct frequencies. The history stacks are initialized with $q = 10$ measurements, where the errors of each stack element are drawn from a normal distribution $\delta_j \sim \mathcal{N}(0, 0.3)$. For 2000 s, the stack elements with the greatest errors – detected by Algorithm 5.4 – are replaced by new measurements. Note that the errors δ_j of new elements follow the same normal distribution as the initial errors. The chosen design parameters are the same as in the previous simulation in Section 5.4.2.

Figure 5.8 shows the estimation process in terms of the normed parameter errors for all three approaches. The normed parameter error for cyclic purging is greater than the normed error obtained with the proposed error detection approach, because cyclic purging exchanges the entire history stack at once. In that case, it is likely that the new history stack contains elements with similarly large errors, leading to no improvement. Even though RLS provides lower estimation errors than concurrent learning with cyclic purging, it is still affected by the errors δ_j. In contrast, Algorithm 5.4 selectively replaces the most erroneous history stack element. After about 900 s, the algorithm has found history stack elements with sufficiently small errors which cause the update terms to fall below the thresholds ϑ^C and ϑ^R in (5.20). This terminates the proposed detection algorithm and the history stack is left unchanged. Consequently, the estimates converge to the smallest neighborhood of the true parameters, as can be seen from the smallest normed parameter error.

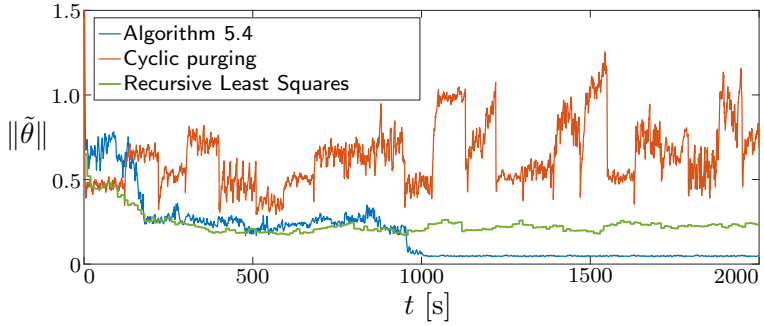


Fig. 5.8: Comparison of the proposed detection mechanism for erroneous stack elements (blue) with a cyclic replacement of the entire history stack (red) and Recursive Least Squares (green).

5.4.4 Adaptive Identification of a Hybrid Two-Tank System

Finally, parameter identifiers with concurrent learning and removal of erroneous history stack elements are applied to estimate the parameters of a PWA model for the hybrid two-tank system shown in Fig. 5.9, which is among the exemplary systems in the HYSDEL toolbox [245].

The system consists of two tanks, each of height 4 m. The cross-sectional areas of the tanks are $C_{t1} = 1.54\,\mathrm{m}^2$ and $C_{t2} = 0.79\,\mathrm{m}^2$. Two pipes connect the tanks: one at the bottom of the tanks and one at height $h_c = 2\,\mathrm{m}$. The cross-sectional area of the upper pipe is $C_{p1} = 0.0314\,\mathrm{m}^2$, which is wider than the one of the lower pipe $C_{p2} = 0.0079\,\mathrm{m}^2$. The only outflow of the system is through the pipe with cross-sectional area $C_{\mathrm{out}} = 0.0154\,\mathrm{m}^2$ at the bottom of the right tank.

The control input is the inflow to tank 1, i.e., $u = q_{\mathrm{in}}$. The levels in the tanks h_1 and h_2 form the states of the system and are governed by the outflow from the second tank

$$q_{\mathrm{out}} = C_{\mathrm{out}}\sqrt{2gh_2}$$

and flows q_1 and q_2 between the tanks, which are given by

$$q_1 = C_{p1}\mathrm{sgn}(h_1 - h_2)\sqrt{g\,\big|(1+\mathrm{sgn}(h_1-h_c))(h_1-h_c) - (1+\mathrm{sgn}(h_2-h_c))(h_2-h_c)\big|},$$
$$q_2 = C_{p2}\mathrm{sgn}(h_1 - h_2)\sqrt{2g\,|h_1 - h_2|},$$

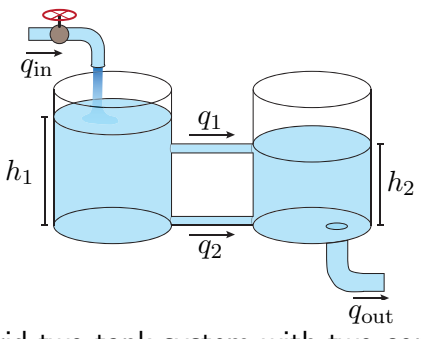


Fig. 5.9: Hybrid two-tank system with two connecting pipes.

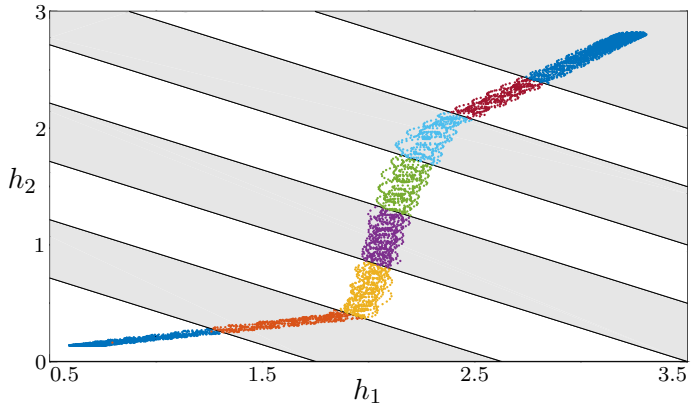


Fig. 5.10: State-space trajectory for excitation with triangle-like offset and two sinusoidal signals. The state-space partitions are given by gray and white stripes as well as different colors of the belonging state measurements.

Hence, the dynamics of the system are

$$\dot{h}_1 = \frac{1}{C_{t1}}(u - q_1 - q_2), \qquad \dot{h}_2 = \frac{1}{C_{t2}}(q_1 + q_2 - q_{\text{out}}).$$

Note that the system dynamics are on the one hand side non-linear due to the Bernoulli equation and on the other hand hybrid due to the sudden change in dynamics as the levels h_1 and h_2 rises above or fall below the height h_c of the connecting pipe. Hence, this example is the first showcase of the proposed parameter identifiers to simultaneously approximate non-linear and hybrid effects by a PWA model of the system.

The maximal inflow $u_{\max} = 0.127\,\text{m}^3/\text{s}$, at which the tanks will overflow, limits the controlled inflow u. The system is excited by two sinusoidal signals with amplitudes $0.3u_{\max}$ and with frequencies $0.4\,\text{rad/s}$ and $1\,\text{rad/s}$. Furthermore, a triangle-like offset that alternates between $0.2u_{\max}$ and $0.9u_{\max}$ is added to u and Gaussian noise $\mathcal{N}(0, 0.003)$ is added to the measured tank levels.

For the a priori partitioning of the state-input space we incorporate the previous knowledge that the system does not switch with respect to the inflow u. Hence, only the two dimensional state space is partitioned. The state-space partitions together with the state-space trajectory obtained with the excitation signal are shown in Fig. 5.10.

The sampling time for fixed point smoothing is set to $dt = 0.05\,\text{s}$ and the noise variance matrices are $R = \text{diag}(0.03, 0.03)$ and $Q = \text{diag}(0.0002, 0.0002, 0.15, 0.15)$. Other concurrent learning design parameters are $A_{\text{m}} = -I_n$, $P = I_n$, $\Gamma = I_{n(n+p+1)}$, $q = 18$ and $\nu = 0.03$. The adaptive identification is carried out for $4000\,\text{s}$ and the parameter estimates in each subsystem converged approximately $500\,\text{s}$ after the corresponding history stacks satisfied Assumption 5.1.

The comparison between the estimated PWA model and the hybrid two tank is shown in Fig. 5.11. The excitation for this experiment differs from the signal which was used to estimate the model parameters. As can be seen from the figure, the estimated PWA model yields a very good approximation of the non-linear, switching dynamics of the hybrid two-tank system.

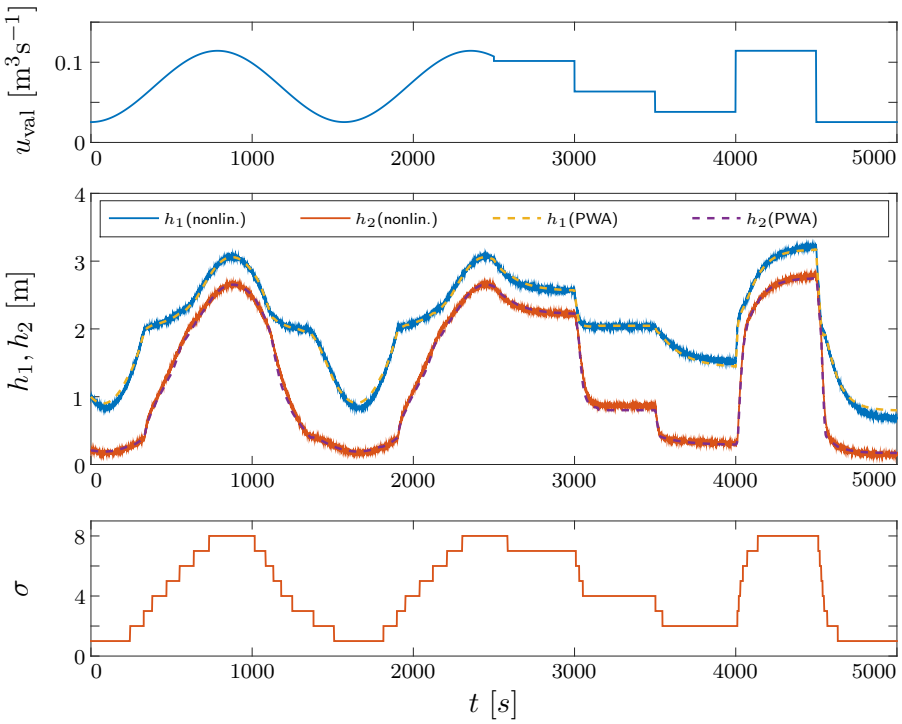


Fig. 5.11: Comparison of the hybrid two-tank system with the obtained PWA model for the validation input u_{val}.

5.5 Summary

In this chapter, the adaptive estimation in PWA systems with parameter identifiers is enhanced through the use of concurrent learning. More precisely, memory-based terms are added to the traditional update laws.

In parameter identifiers, the adaptation process is driven by state prediction errors, where the state of the system is predicted by an observer structure. Concurrent learning uses prediction errors of the state derivatives. For this purpose, historical data of state derivatives and corresponding states and control inputs is recorded in a history stack. Each recorded state derivative is predicted based on its recorded state and control input and the current parameter estimates. The prediction errors are incorporated into the update laws in such a way that the Lyapunov function is shown to be negative definite independent of the switching sequence given that a sufficient amount of linearly independent data has been recorded.

The eminent benefit of using concurrent learning is related to the excitation needed in the system to correctly estimate all parameters. Traditional update laws are solely based on current measurements and hence demand for a persistent excitation throughout the estimation process. This is both hard to ensure and to monitor. With concurrent learning, a more simple requirement on linearly independent recorded data replaces the previous persistent excitation assumptions. In the switched systems setting, the introduction of recorded data furthermore allows to continuously update even currently inactive parameter estimates. As a consequence, parameter identifiers with concurrent learning achieve

improved convergence rates in the form of exponential convergence.

One limitation of concurrent learning is the assumption of full state measurements and state derivatives. Currently there exists no generalization to systems with partial state measurements. In case the state derivative is not available, the use of fixed point smoothing is proposed to obtain accurate estimates of a finite number of state derivatives. It is shown however that outdated or erroneous history stack data causes estimates to converge to wrong parameters. This is due to a compact set in the state-input space, caused by errors in the state derivatives, for which the derivative of the Lyapunov function is positive. Hence, no exact convergence to the true parameters is possible with erroneous history stack data. An approximation for the size of the compact set and therefore also the parameter error is given. In order to further overcome problems with erroneous data, an algorithm is suggested to detect and remove erroneous data from history stacks. A limitation of the proposed history stack management is the need of persistent excitation for the time frame in which erroneous data are to be replaced. A step for future work is to test the proposed algorithms in adaptive control systems.

6 Multiple Models-based Identification of PWA Systems

The previous chapters analyzed two challenging sub-problems in the identification of PWA systems. In Chapter 3, the state-space partitions were reconstructed based on knowledge of the actual subsystem parameters. In Chapter 4 and 5, it was assumed that the partitions are known and the subsystem parameters were estimated with parameter identifiers. This chapter presents a novel identification algorithm for continuous-time piecewise affine (PWA) systems in state-space form, which is capable of estimating both the subsystem dynamics and the state-space partitions. The algorithm is based on the performance index applied in Multiple Models Adaptive Control [81, 107, 176, 178]. We show that the performance index corresponds to a metric to quantify how far the current parameter estimates are away from the true system parameters. A swarm algorithm incorporates this parametric distance measure in order to redistribute its elements in a goal-oriented way. We prove that the algorithm produces a matching model for each subsystem of the PWA system under a dwell time assumption. The presented swarm algorithm enables estimation of all subsystem parameters for unknown switching signals and thus unknown state-space partitions. Finally, with the estimated subsystem parameters, the state-space partitions are reconstructed according to Chapter 3.

This chapter is structured as follows. We discuss the performance index in Multiple Models and its ability to express distances in the parameter space in Section 6.1. Based on the performance index, Section 6.2 presents an identification algorithm to estimate PWA subsystem parameters for unknown switching signals. Convergence of the algorithm is proven and some important properties such as the required number of swarm models and the inclusion of previous knowledge are discussed. Section 6.3 points out an approach to combine the performance index with the previously discussed parameter identifiers. Despite promising results, no proof of convergence is given. In Section 6.4, the identification algorithms are numerically tested in terms of an exemplary PWA system and a hybrid two-tank system. Conclusions are drawn in Section 6.5.

6.1 Distance Measure in Multiple Models

The concept of Multiple Models was introduced to the field of adaptive control in the 1990s in order to extend adaptive control from systems with slowly changing parameters to systems with rapidly changing parameters [176]. Before that, rapidly changing parameters were problematic as they correspond to large, sudden jumps in the parameter space. Traditional adaptive control would take a long time to adapt the controller gains through gradient descent approaches. *Multiple Models Adaptive Control (MMAC)* shortened the adaptation process by running multiple models of the process with different parametriza-

tions in parallel. This allows to reinitialize the adaptation with better parameter estimates by switching to the best model. The best model is selected in terms of a performance index.

As it is shown next, the performance index corresponds to a distance in the parameter space between the true system parameters and the model parameters. This distance constitutes the critical success factor for the identification of PWA systems proposed in this chapter. We previously compared the performance to other metrics for linear systems in canonical form in [275]. Here, we extend the previous results by showing that the performance index is also a valid metric to compare non-canonical, affine state-space systems.

Consider the affine system

$$\dot{x}(t) = Ax(t) + Bu(t) + f, \tag{6.1}$$

with unknown parameters $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times p}$ and $f \in \mathbb{R}^n$. Let (A, B) be controllable and assume A is invertible. In MMAC, a set of $N \in \mathbb{N}$ models with different parametrizations $\hat{A}_j,\ \hat{B}_j,\ \hat{f}_j,\ j = 1, \ldots, N$ is introduced. The N models are distributed in such a way that they cover the parameter space of the unknown affine system (6.1). The goal is then to determine at each time instance which of the models is closest to (i.e. approximates best) the true system (6.1). For that purpose each model predicts the state x of the true system by

$$\dot{\hat{x}}_j(t) = A_\mathrm{m}\hat{x}_j(t) + (\hat{A}_j - A_\mathrm{m})x(t) + \hat{B}_j u(t) + \hat{f}_j, \tag{6.2}$$

where $A_\mathrm{m} \in \mathbb{R}^{n\times n}$ is an asymptotically stable, diagonal matrix chosen by the designer to ensure that the predicted states $\hat{x}_j$ remain bounded. The diagonal entries of A_m are referred to with $a_{\mathrm{m},k},\ k = 1, \ldots, n$. The performance of each model is evaluated in terms of the error between predicted state and measured state: $\tilde{x}_j = \hat{x}_j - x$. The most general *performance index in Multiple Models* is

$$J_j(t) = \alpha\tilde{x}_j^\top(t)\tilde{x}_j(t) + \beta\int_0^t \mathrm{e}^{-\gamma(t-\tau)}\tilde{x}_j^\top(\tau)\tilde{x}_j(\tau)d\tau, \tag{6.3}$$

with design parameters $\alpha, \beta, \gamma \geq 0$. The parameters α and β define to which extent current and past prediction errors are taken into consideration in the performance index. The forgetting factor γ specifies how quickly the effect of past prediction errors vanishes.

Besides the design parameters α, β, γ and A_m, the performance index (6.3) also depends on the control inputs u. A desirable property of the input signal for the performance index – and system identification in general – is to be sufficiently rich. Recall Section A.2 for the definition of sufficiently rich signals and other relevant notions such as persistence of excitation. By Definition A.4, a signal consisting of $n + 1$ sinusoidal components with distinct frequencies is sufficiently rich of order $2(n + 1)$.

Before analyzing the performance index (6.3) further, apply the same notation as before and introduce $\theta \in \mathbb{R}^{n(n+p+1)}$, which concatenates all parameters of one model in a single column vector. Hence, the parameter vector of the affine system (6.1) is given by

$$\theta = \begin{bmatrix}\mathrm{vec}(A)^\top & \mathrm{vec}(B)^\top & f^\top\end{bmatrix}^\top. \tag{6.4}$$

The parameter vector of the j-th model in the Multiple Models framework (6.2) be $\hat{\theta}_j$. In the later sections dealing with PWA systems, let the parameter vector of a single affine subsystem be θ_i. Assume for now a single fixed subsystem $\theta = \theta_i$, and express the difference between the affine system and the j-th model by $\tilde{\theta}_j = \hat{\theta}_j - \theta$. The following lemma states a significant relationship between the parameter error $\tilde{\theta}_j$ and the performance index J_j.

Lemma 6.1 Assume that the affine system (6.1) and the predictor for the j-th model (6.2) are initialized at $t = 0$ with the same state $x(0) = \hat{x}_j(0)$. If both are excited by the same input signals u, each sufficiently rich of order $n + 1$ with distinct frequencies, then the performance index (6.3) at a time T corresponds to the square of a weighted vector norm:

$$J_j(T) = \tilde{\theta}_j^\top \mathcal{W}(T)\tilde{\theta}_j = \left(\|\tilde{\theta}_j\|_{\mathcal{W}(T)}\right)^2. \tag{6.5}$$

Proof: The proof of Lemma 6.1 is divided into two parts. First the relationship $J_j(T) = \tilde{\theta}_j^\top \mathcal{W}(T)\tilde{\theta}_j = \|\tilde{\theta}_j\|^2_{\mathcal{W}(T)}$ is derived, before the *weighting matrix* $\mathcal{W}(T)$ is shown to be positive definite. The positive definiteness of $\mathcal{W}(T)$ is a necessary condition for J_j to be a valid metric.

With $\tilde{\theta}_j$, the dynamics of the prediction error read as

$$\dot{\tilde{x}}_j(t) = \dot{\hat{x}}_j(t) - \dot{x}(t) = A_\mathrm{m}\tilde{x}_j(t) + \Psi^\top(t)\tilde{\theta}_j, \tag{6.6}$$

with

$$\Psi^\top(t) := \begin{bmatrix} x^\top(t) & u^\top(t) & 1 \end{bmatrix} \otimes I_n, \tag{6.7}$$

and with the Kronecker product $\otimes$ and the identity matrix $I_n \in \mathbb{R}^{n\times n}$.

By assumption in Lemma 6.1, the error dynamics (6.6) are initialized with $\tilde{x}_j(0) = 0$. The solution of (6.6) is therefore given by

$$\tilde{x}_j(t) = \int_0^t \mathrm{e}^{A_\mathrm{m}(t-\tau)}\Psi^\top(\tau)\tilde{\theta}_j \, d\tau = \underbrace{\int_0^t \mathrm{e}^{A_\mathrm{m}(t-\tau)}\Psi^\top(\tau)d\tau}_{=:\mathcal{V}^\top(t)} \tilde{\theta}_j = \mathcal{V}^\top(t)\tilde{\theta}_j, \tag{6.8}$$

where $\mathcal{V} \in \mathbb{R}^{n(n+p+1)\times n}$ and $\mathcal{V}(0) = 0$. Note that $\tilde{\theta}_j$ is assumed to be constant and can therefore be taken out of the integral. Inserting (6.8) into the general performance index in (6.3) yields

$$J_j(T) = \tilde{\theta}_j^\top \mathcal{W}(T)\tilde{\theta}_j \tag{6.9}$$

with

$$\mathcal{W}(T) = \alpha\mathcal{V}(T)\mathcal{V}^\top(T) + \beta\int_0^T \mathrm{e}^{-\gamma(T-t)}\mathcal{V}(t)\mathcal{V}^\top(t)dt, \tag{6.10}$$

$\mathcal{W} \in \mathbb{R}^{n(n+p+1)\times n(n+p+1)}$ and $\mathcal{W}(0) = 0$. This shows that the performance index $J_j(T)$ is a time varying quadratic function of the parameter difference $\tilde{\theta}_j$. It is left to show that $\mathcal{W}(T)$ is positive definite. Note that the product $\mathcal{V}(T)\mathcal{V}^\top(T)$ in (6.10) can only be positive semi-definite. Positive definiteness of $\mathcal{W}(T)$ may thus only stem from the positive definiteness

of the integral

$$\int_0^T \mathcal{V}(t)\mathcal{V}^\top(t)dt, \tag{6.11}$$

assuming $\gamma = 0$ for simplicity. To show positive definiteness of (6.11), begin by exploiting the sparsity of V induced by the Kronecker product in Ψ (see (6.7)) as

$$\mathcal{V}(t) = \int_0^t \Psi(\tau)\mathrm{e}^{A_\mathrm{m}^\top(t-\tau)}d\tau. \tag{6.12}$$

From (6.12), it follows that the k-th column of $\mathcal{V}(t)$ is

$$v_k(t) = \underbrace{\left(\int_0^t \mathrm{e}^{a_{\mathrm{m},k}(t-\tau)} \begin{bmatrix} x(\tau) \\ u(\tau) \\ 1 \end{bmatrix} d\tau\right)}_{=:\nu_k(t)} \otimes \ell_k = \nu_k(t) \otimes \ell_k, \tag{6.13}$$

with $\nu_k \in \mathbb{R}^{n+p+1}$, and where $\ell_k \in \mathbb{R}^n$ denotes the k-th column of the identity matrix I_n, i.e. only the k-th element of ℓ_k is 1. The sparsity of $\mathcal{V}$ allows to express (6.11) in terms of the columns v_k as

$$\begin{aligned}\int_0^T \mathcal{V}(t)\mathcal{V}^\top(t)dt &= \int_0^T \sum_{k=1}^n v_k(t)v_k^\top(t)dt = \int_0^T \sum_{k=1}^n \left(\nu_k(t) \otimes \ell_k\right)\left(\nu_k^\top(t) \otimes \ell_k^\top\right)dt \\ &= \sum_{k=1}^n \left(\int_0^T \nu_k(t)\nu_k^\top(t)dt\right) \otimes \left(\ell_k\ell_k^\top\right). \end{aligned} \tag{6.14}$$

Now note that $\ell_k\ell_k^\top$ is a matrix in which only the element at (k,k) is 1. In combination with the Kronecker product, this ensures that two elements of the sum in (6.14) are never placed at the same position and the resulting matrix is sparse. The sparsity is such that (6.14) is positive definite if and only if each integral in the summation is positive definite, i.e.

$$\int_0^T \nu_k(t)\nu_k^\top(t)dt \succ 0, \quad \forall k \in \{1,\ldots,n\}. \tag{6.15}$$

The positive definiteness in (6.15) is equivalent to ν_k being persistently excited (PE). To see this compare (6.15) to the PE Definition A.3 in Section A.2. The satisfaction of the PE requirement results from the assumption of sufficiently rich input signals u. Lemma 4.4 guarantees that the vector $z = [x^\top,\ u^\top,\ 1]^\top$ is PE if the following three conditions hold: 1. the system matrix A is invertible, 2. the pair (A, B) is controllable, and 3. the elements of u are sufficiently rich of order $n+1$ with distinct frequencies. By assumption, these conditions are satisfied. Hence, PE of z enables showing PE of ν_k.

Next, (6.13) indicates that the internal vector z relates to the vector ν_k by the following transfer function:

$$\nu_k(s) = (s - a_{\mathrm{m},k})^{-1} I_{(n+p+1)} z(s). \tag{6.16}$$

Due to the stability of A_m, this transfer function has a single pole in the open left half-

plane. Following Lemma 7 and Corollary 1 in [174], the PE condition of z together with the obtained transfer function guarantees PE of ν_k. Hence, the positive definiteness of $\mathcal{W}(T)$ follows for $T > 0$. □

Lemma 6.1 shows that the performance index in Multiple Models corresponds to the weighted vector norm of the parameter difference. This insight forms the foundation for the identification algorithm which is to be presented next.

6.2 Identification of PWA Systems with Multiple Models

While the previous section discussed a distance measure between two affine systems, we will now turn our focus onto the identification of PWA systems. As before, consider PWA systems in state-space form given by

$$\dot{x}(t) = A_{\sigma(t)}x(t) + B_{\sigma(t)}u(t) + f_{\sigma(t)}, \tag{6.17}$$

where the switching signal $\sigma(t)$ depends on the state-space partitions Ω_i by

$$\sigma(t) = i \quad \text{if} \quad \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} \in \Omega_i, \quad i = 1, \ldots, s \tag{6.18}$$

with

$$\Omega_i = \left\{ \begin{bmatrix} x \\ u \end{bmatrix} \in \mathbb{R}^{n+p} \,\middle|\, \mathcal{H}_i \begin{bmatrix} x \\ u \\ 1 \end{bmatrix} \preceq_{[i]} 0 \right\}. \tag{6.19}$$

In this chapter we assume all A_i, B_i, f_i (equivalently θ_i) and Ω_i to be unknown. Recall then that the subsystem identification problem can be phrased as the optimization problem

$$\underset{\{\hat{\theta}_j\}_{j=1}^N}{\text{minimize}} \; \sum_{i=1}^{s} \min_j \left\| \theta_i - \hat{\theta}_j \right\|, \tag{6.20}$$

where $N \geq s$. In other words, for each subsystem θ_i, one must find at least one model $\hat{\theta}_j$ with the same parameters. Unfortunately, the Euclidean distance in (6.20) can not be evaluated due to the unknown true parameters θ_i. This is where Lemma 6.1 comes in as an enabling factor for the solution of (6.20) and thus the identification of subsystems in PWA systems. As a result of Lemma 6.1, a large set of parameter vectors $\hat{\theta}_j$ can be sorted according to their distances to the true parameter vector θ_i measured in terms of the weighted vector norm induced by $\mathcal{W}(T)$. This is exploited in a swarm-like identification algorithm in which models are repeatedly redistributed in order to minimize the weighted vector norm in order to solve the optimization problem

$$\underset{\{\hat{\theta}_j\}_{j=1}^N}{\text{minimize}} \; \sum_{i=1}^{s} \min_j \left\| \theta_i - \hat{\theta}_j \right\|_{\mathcal{W}(t)}, \tag{6.21}$$

which shares the same optimal solution as (6.20).

Particle swarm algorithms are a popular tool to find optima in non-convex optimization problems. In particle swarm methods, a swarm of models is distributed in the parameter space. The parameters of each particle are updated based on its own position and performance as well as on the parameters and performance of other particles. While all particles are considered in standard particle swarm optimization, the algorithm in [165] allows only a subset of neighboring particles to contribute to the update, which avoids convergence to local minima. The paper compares the standard and the distributed particle swarm optimization algorithm in a test scenario of identifying a PWA model for a DC motor. Also, work in [73] proposes to identify Takagi-Sugeno fuzzy systems through chaotic particle swarm optimization. The chaotic component was added to increase the search space of the algorithm and avoid convergence to local minima. The use of particle swarm optimization with quantum infusion for the identification of adaptive infinite impulse response systems was proposed by [158]. Due to the heuristic nature of particle swarm optimization techniques, they usually lack a formal proof of convergence.

A provably converging swarm-like identification technique was introduced by [82] in the adaptive control community. It is stated that the performance index in Multiple Models Adaptive Control enables sorting a set of models according to their distances to the true system parameters. By updating or redistributing the set of models based on this distance measure, one can guarantee that for each subsystem there exists at least one model with the same correct parameters. We build on the ideas presented in [82] and extend them to PWA systems. Furthermore, we are going add an elegant way to include previous knowledge about the system and by that increase convergence rates and reduce computational complexity.

6.2.1 Performance-based Redistribution

We now introduce the identification algorithm for PWA systems in state-space form which is based on the distance measure discussed in the previous section. Begin by distributing a large number N of affine models $\hat{\theta}_j$ in the parameter space. Each model is implemented as a predictor (6.2) and initialized with the same state as the PWA system (which is assumed to be measurable). The idea of the proposed algorithm is to repeatedly redistribute a subset of the models in the parameter space. A goal-oriented redistribution strategy guarantees that for each of the PWA subsystems there is going to exist an affine model with the same values.

After initialization, sufficiently rich input signals excite the PWA system and all models. During excitation the performance indices J_j of all models are evaluated. Furthermore, $\mathcal{W}(T)$ is recorded through (6.10). At time instance T, pause the excitation and sort all models according to their performance indices J_j in an increasing order. Due to Lemma 6.1, it is know that the lowest J_j corresponds to the parameter configuration which is closest – in terms of the norm induced by $\mathcal{W}(T)$ – to the currently active subsystem θ_i. Therefore, the corresponding model $\hat{\theta}_{\min}$ is fixed as the best estimate in order to avoid it being redistributed in a later step.

In the next step, the algorithm redistributes a subset of $M \in \mathbb{N}$ models around the best estimate. The motivation behind this redistribution step is to obtain a new model which is closer to the currently active subsystem of the PWA system. Hence, the first M flexible

(not yet fixed) models from the sorted list are uniformly redistributed in a suitably chosen subspace $\mathcal{R}$ of the parameter space $\mathbb{R}^{n(n+p+1)}$.

An appropriate choice of $\mathcal{R}$ is essential for the success of the algorithm. Note that the best performance index $J_{\min} = \min_j J_j(T)$ together with the weighting matrix $\mathcal{W}(T)$ defines an ellipsoid in the parameter space centered at the parameter vector θ_i of the active subsystem:

$$(\hat{\theta}_{\min} - \theta_i)^\top \mathcal{W}(\hat{\theta}_{\min} - \theta_i) = J_{\min}. \tag{6.22}$$

A suitable choice for $\mathcal{R}$ is therefore a scaled version of (6.22), which is now centered at the best model $\hat{\theta}_{\min}$:

$$\mathcal{R} = \left\{ \hat{\theta} \in \mathbb{R}^{n(n+p+1)} \;\middle|\; (\hat{\theta} - \hat{\theta}_{\min})^\top \mathcal{W}(\hat{\theta} - \hat{\theta}_{\min}) \leq \delta_r J_{\min} \right\}, \tag{6.23}$$

where $\delta_r > 0$ is a design constant called redistribution factor to scale $\mathcal{R}$. The algorithm continues by repeating the sequence of excitation and redistribution either until it meets a stopping criterion on the prediction errors $\tilde{x}_j$ or it exceeds a maximum number of iterations. The corresponding pseudo-code is given by Algorithm 1.

Algorithm 1 Identification with Multiple Models

1: Initialize N models $\hat{\theta}_j,\ j = 1, \ldots, N$
2: **while** stopping criterion is not met **do**
3: Reset model states $\hat{x}_j(t) = x(t)$
4: Reset $\mathcal{W}(t) = 0$ and $J_j(t) = 0$
5: Excite PWA system and all models with a sufficiently rich input u over a period of at least T
6: Fix model $\hat{\theta}_{\min}$ with $J_{\min} = \min_j J_j(T)$
7: Sort J_j in increasing sequence
8: Redistribute first M flexible models of sorted list in $\mathcal{R}$
9: **end while**

An important assumption for the convergence of the algorithm is that the same subsystem θ_i of the PWA system is active for at least two full intervals of length greater or equal to T. That is, the PWA system is assumed to have a dwell time greater or equal to $3T$. Note that this assumption constitutes the biggest limitation of the algorithm, as some PWA systems cannot be kept in a certain region while at the same time fulfilling the sufficiently rich excitation.

6.2.2 Parameter Convergence

The convergence of Algorithm 1 is guaranteed by the following theorem.

Theorem 6.2 Consider the switched affine system (6.17) parametrized by $\theta_i,\ i = 1, \ldots, s$ with (A_i, B_i) controllable and A_i invertible. Let the system be excited by input signals in u, each sufficiently rich of order $n + 1$ with distinct frequencies. Also, assume that the

switching signal σ has a dwell time of at least $3T$ and repeatedly activates all subsystems. Then, Algorithm 1 guarantees that for each subsystem i there exists a model j such that $\hat{\theta}_j = \theta_i$ as $t \to \infty$.

Proof: The proof of Theorem 6.2 is of a probabilistic nature. By assumption, one subsystem is active for at least $3T$. Therefore, the identification algorithm witnesses at least two consecutive time frames of length T in which the same subsystem is active. The proof must thus show that the redistribution step between these two time frames generates a model which is closer to the active subsystem parameters with a probability greater than zero. The effect of time frames in which two different subsystems are active will be discussed afterwards.

Let θ_i be the currently active subsystem and assume that $\hat{\theta}_{\min}$ was found to be the closest model at time instance T_1 (after the first time frame). Then, the algorithm fixes $\hat{\theta}_{\min}$ and uniformly redistributes a subset of M models inside the ellipsoid $\mathcal{R}$ defined in (6.23). Note that $\mathcal{R}$ includes the currently active parameter vector θ_i if $\delta_r \geq 1$. After the second time frame (at time instance T_2), a different weighting matrix $\mathcal{W}(T_2)$ is obtained due to potentially different excitation. At that time, the algorithm fixes a model $\hat{\theta}_j \in \mathcal{R}$ only if its performance index $J_j(T_2)$ is lower than the one of the previously fixed model $\hat{\theta}_{\min}$, i.e. if

$$(\hat{\theta}_j - \theta_i)^\top \mathcal{W}(T_2)(\hat{\theta}_j - \theta_i) < (\hat{\theta}_{\min} - \theta_i)^\top \mathcal{W}(T_2)(\hat{\theta}_{\min} - \theta_i). \tag{6.24}$$

Figure 6.1 visualizes the two ellipsoids for an exemplary two dimensional parameter vector. The first (light blue) ellipsoid is associated with $\mathcal{W}(T_1)$ and describes the subspace in which the models are redistributed. The second (light red) ellipsoid is associated with $\mathcal{W}(T_2)$ and defines the subspace in which a model $\hat{\theta}_j$ would achieve a better performance than $\hat{\theta}_{\min}$ at time instance T_2. The figure furthermore highlights the subspace in which the redistributed models would be fixed (red and green subspace). This subspace is split once more into two: one subspace (red) with models of increased Euclidean distance to the active subsystem parameters θ_i and another one (green) with reduced Euclidean distance. Desirable are only those models with reduced Euclidean distance, i.e. $\|\hat{\theta}_j - \theta_i\|_2 < \|\hat{\theta}_{\min} - \theta_i\|_2$. The subspace of redistributed models that constitute better estimates of θ_i than $\hat{\theta}_{\min}$ is given by

$$\begin{aligned} \mathcal{R}_{\text{imp}} = \Big\{ \hat{\theta} \;\Big|\; & \|\hat{\theta} - \hat{\theta}_{\min}\|_{\mathcal{W}(T_1)} \leq \delta_r J_{\min} \quad \cap \quad \|\hat{\theta} - \hat{\theta}_i\|_{\mathcal{W}(T_2)} < \|\hat{\theta}_{\min} - \hat{\theta}_i\|_{\mathcal{W}(T_2)} \\ & \cap \quad \|\hat{\theta} - \theta_i\|_2 < \|\hat{\theta}_{\min} - \theta_i\|_2 \Big\}. \end{aligned} \tag{6.25}$$

In Fig. 6.1, the subspace $\mathcal{R}_{\text{imp}}$ is visualized in green. As this space is never empty, the identification with Multiple Models guarantees an improved model after each redistribution with a probability greater than zero. The probability of obtaining an improved model is

$$\rho = 1 - \left(1 - \frac{\mathsf{vol}(\mathcal{R}_{\text{imp}})}{\mathsf{vol}(\mathcal{R})}\right)^M, \tag{6.26}$$

where $\mathsf{vol}(\cdot)$ stands for the volume of a set.

Besides the case where two consecutive time frames of length T share the same active

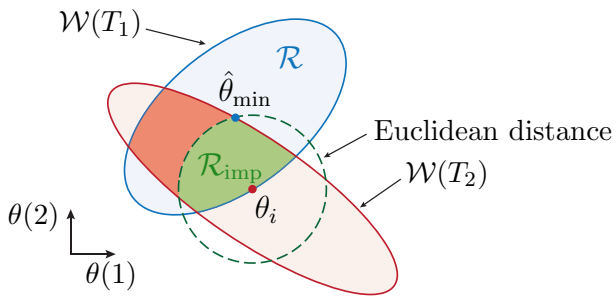


Fig. 6.1: Exemplary visualization of the sets $\mathcal{R}$ and $\mathcal{R}_{\text{imp}}$ encountered during redistribution of the particles (for $\theta \in \mathbb{R}^2$ and $\delta_r = 1$).

subsystem, another case exists in which the active subsystem switches within the time frame. In this case, Lemma 6.1 does not apply because the parameters of the unknown system are not constant. In turn, the performance indices do not allow for a valid sorting of the models. Consequently, fictitious parameters might be chosen that are neither close to the first nor the second subsystem. By assuming a sufficiently large number of models for redistribution, these fictitious models can be neglected. Also in view of the optimization problem in (6.20), it suffices to obtain just one model for each subsystem with the same parameters. □

Finally, note that with the estimated subsystem parameters obtained through Theorem 6.2, one can resort to the procedure presented in Chapter 3 in order to partition the state-input space and thus derive a complete model of the PWA system (6.17).

6.2.3 Properties and Extensions

A few comments on Algorithm 1, Theorem 6.2 and their implications are in order.

Mistakenly fixed models

One problem arising in Algorithm 1 is that too many models might be fixed in the neighborhood of a single subsystem. Especially with noisy measurements, the algorithm tends to accumulate plenty of fixed models around the true subsystem parameters. We attenuate this effect with an additional clustering routine once the stopping criterion is met. In the clustering step all fixed models are clustered in a way that the center of each cluster has a maximum Euclidean distance $\varepsilon_\theta > 0$ to all its elements. Note that ε_θ limits the minimum distance that two fixed models are allowed to have and by that also constitutes a trade-off between number of models and fit to a nonlinear system.

Besides very similar models due to noisy measurements, there is a second class of wrongly fixed models. These are intermediate models that were fixed as a good estimate during an early redistribution step but which are still very far away from the true parameters. Also closely related are the fictitious models that are fixed during an interval in which the PWA system undergoes a switch in dynamics. Both kinds can be easily identified as wrongly fixed models by the fact that their performance index is minimal only once. Good estimates on the other hand will repeatedly deliver the best performance index. In order to

eliminate intermediate and fictitious models, we define a maximum time $T_{\max}$ for which a model may be labeled as fixed without being chosen as the best match. If the time between the last selection of a fixed model and the current time instance exceeds $T_{\max}$, the model is set free again and may be redistributed in the next iteration. For this approach to work, however, it must be verified that the PWA system frequently returns to all subsystem, i.e. the time between two consecutive visits must be smaller than $T_{\max}$. Otherwise even correct models might be set free again.

Relationship between convergence and number of models

Next, note that the dimension of the parameter vectors θ_i strongly affects the speed of convergence. This is of course common to all PWA system identification algorithms. For the proposed algorithm, this can be seen best by looking at the probability ρ in (6.26). Assume that the system is excited in such a way that $\mathcal{R}$ always takes the shape of an hypersphere and let $\delta_r = 1$. In that case, the set $\mathcal{R}_{\text{imp}}$ is inner approximated by an hypersphere with half the radius of $\mathcal{R}$. Consequently we have that $\mathcal{V}(\mathcal{R}_{\text{imp}})/\mathcal{V}(\mathcal{R}) = 0.5^d$, where d is the dimension of the parameter space. This allows finding a lower bound for ρ. By rearranging this lower bound for M, one obtains the required number of redistributed models M_{req} as a function of the dimension d and the desired probability ρ_{des}:

$$M_{\text{req}} = \frac{\log(1-\rho_{\text{des}})}{\log(1-0.5^d)}. \tag{6.27}$$

This indicates that the number of redistributed models should grow exponentially with the number of unknown parameters. In order to keep computational complexity low, it is hence important (and luckily straightforward) to incorporate available previous knowledge about the PWA system such as sparsity or known system parameters.

Incorporation of previous knowledge

Assume that the parameter vector $\hat{\theta}$ can be divided into an unknown part $\theta_u \in \mathbb{R}^{d_u}$ and a known part $\theta_k \in \mathbb{R}^{d_k}$ with $d_u + d_k = d$. In that case, the known elements of the parameter vector lead to zero estimation errors, hence the parameter error of each model reduces to $\tilde{\theta}_j^\top = \begin{bmatrix}\tilde{\theta}_u^\top & 0^\top\end{bmatrix}^\top$. Furthermore, the weighting matrix $\mathcal{W}(T)$ associated with the performance index $J_j(T)$ can be divided, which changes (6.5) to

$$J_j(T) = \begin{bmatrix}\tilde{\theta}_u^\top & 0^\top\end{bmatrix} \begin{bmatrix}\mathcal{W}_{uu} & \mathcal{W}_{uk} \\ \mathcal{W}_{ku} & \mathcal{W}_{kk}\end{bmatrix} \begin{bmatrix}\tilde{\theta}_u \\ 0\end{bmatrix} = \tilde{\theta}_u^\top W_{uu} \tilde{\theta}_u, \tag{6.28}$$

where $\mathcal{W}_{uu} \in \mathbb{R}^{d_u \times d_u}$, $\mathcal{W}_{uk} \in \mathbb{R}^{d_u \times d_k}$, $\mathcal{W}_{ku} \in \mathbb{R}^{d_k \times d_u}$ and $\mathcal{W}_{kk} \in \mathbb{R}^{d_k \times d_k}$ are submatrices of $\mathcal{W}$. The important submatrix $\mathcal{W}_{uu}$ is obtained from $\mathcal{W}$ by deleting all columns and rows associated with known elements in θ. The subspace $\mathcal{R}$ in which the models are redistributed therefore reduces to

$$\mathcal{R}_u = \left\{ \hat{\theta}_u \in \mathbb{R}^{d_u} \;\middle|\; (\hat{\theta}_u - \hat{\theta}_{u,\min})^\top \mathcal{W}_{uu} (\hat{\theta}_u - \hat{\theta}_{u,\min}) \leq \delta_r J_{\min} \right\}, \tag{6.29}$$

where $\hat{\theta}_{u,\min}$ is the unknown part of the parameter vector with the best performance index at the last evaluation. The systematic application of previous knowledge in form of (6.29) helps to keep the number of models required by (6.27) in acceptable bounds.

State-sharing approach

In case the number of models remains unacceptable, one can resort to the state-sharing approach discussed in [112, 169, 170]. *State sharing* enables the prediction of N states $\hat{x}_j$, $j = 1, \ldots, N$ with different parameters $\hat{\theta}_j$ according to (6.2) based on a shared-state observers. The key feature is that the dimension of the shared-state observer is independent of the number of models. Instead of predicting N states $\hat{x}_j$ by

$$\dot{\hat{x}}_j = A_\mathrm{m}(\hat{x}_j - x) + \hat{A}_j x + \hat{B}_j u + \hat{f}_j = A_\mathrm{m}(\hat{x}_j - x) + \Psi^\top \theta_j, \tag{6.30}$$

the following two differential equations

$$\dot{x}_I = A_\mathrm{m} x_I - A_\mathrm{m} x \qquad \text{and} \qquad \dot{x}_{II} = A_\mathrm{m} x_{II} + \Psi^\top \tag{6.31}$$

with $x_I \in \mathbb{R}^n$ and $x_{II} \in \mathbb{R}^{n \times n(n+p+1)}$ are used. Note that the dynamics of x_I and x_{II} do not depend on the model parameters $\hat{\theta}_j$. Hence, it is rather surprising that the state prediction for an arbitrary $\hat{\theta}_j$ can be obtained from x_I and x_{II} by

$$\hat{x}_j = x_I + x_{II}\hat{\theta}_j. \tag{6.32}$$

To see that this is actually true, simply take the time derivative of (6.32), insert (6.31) and rearrange the result to obtain (6.30). To quantify the benefit of state-sharing observers, inspect the required number of states. In the parallel approach, one needs a total of nN states as compared to the $n + n^2(n+p+1)$ needed in (6.31). It can thus be concluded that state-sharing is beneficial in a sense that less states are needed if $N > 1 + n(n+p+1)$.

Offline identification

Finally, note that Algorithm 1 was so far described from the perspective of online identification, in which the system is constantly excited and models are frequently redistributed. The advantage of this setting is that the set of models reacts to changes in the parametrization of the PWA system due to for instance wear or aging. However, the application of the proposed algorithm for offline identification is also possible. Given recorded input and state measurements over a long time frame T_total, one would divide this long frame into shorter time frames of length T. For each frame the models are initialized with the recorded state and excited by the recorded input signal. After multiple iterations of Algorithm 1 over the time frame T_total with potentially different choices of T, the set of models converge to the subsystem parameters of the PWA system.

6.3 Parameter Identifiers with Performance-based Weighting

In the previous section, the Multiple Models performance index was used to sort and redistribute a large set of models according to their distance to the currently active subsystem parameters. This approach differs from the parameter identifiers discussed in previous chapters in a sense that adaptation takes place in discrete steps of redistribution rather than continuous adaptation. In this section, we discuss a combination of parameter identifiers with the performance index of Multiple Models which enables a continuous adaptation of parameter estimates.

Bako *et al.* used the performance index J in [12] to estimate at each time instance the active subsystem. It is assumed that the model that produces the best performance index corresponds to the active subsystem:

$$\hat{\sigma}(t) = \arg\min_j J_j(t). \tag{6.33}$$

The available input-output data is then used to update the parameters associated with the assumed to be active subsystem $\hat{\sigma}$. While [12] applies recursive least squares to identify a PWARX model, $\hat{\sigma}$ can equivalently replace the true switching signal in Chapter 4 in order to extend parameter identifiers to PWA systems with unknown switching signal. The simulation results in [12] indicate good performance. Nevertheless, there are no guarantees for convergence. The algorithm depends heavily on a suitable initialization of the parameter estimates, i.e. there should be an estimate in the vicinity of each subsystem. If one estimate is closest to all subsystem parameters, the algorithm tends to continuously adapt the same model which in turn can not converge to a single subsystem configuration.

Another promising approach to incorporate the performance index to the identification of switched systems is pursued in [81] and [131–134]. The idea is to move away from determining the single-best-performing estimate and update it. Instead, all estimates are updated simultaneously in a weighted way by the available input-output data. The performance indices J_j are used to define a weighting signal for each estimate that describes how well the estimate matches the current system behavior. The intuitive rationale behind the algorithm is to adjust strongly those estimates that predict well the current system behavior (i.e. have small J_j). On the other hand, models with large values in J_j are only updated weakly. The *performance weight* w_j of the j-th model is given by

$$w_j(t) = \frac{\frac{1}{J_j(t)}}{\sum_{k=1}^{N} \frac{1}{J_k(t)}} = \frac{\frac{1}{J_j(t)}}{\frac{1}{J_1(t)} + \frac{1}{J_2(t)} + \cdots + \frac{1}{J_N(t)}}. \tag{6.34}$$

This definition yields weights in the range $w_j \in (0, 1)$. Large values in J_j result in weights close to zero and small values in J_j result in weights close to one. Furthermore, all weights sum up to 1, i.e. $\sum_{k=1}^{N} J_k(t) = 1$.

The weights are included into the traditional parameter identifiers discussed in Chapter 4

in the following way:

$$\dot{\hat{A}}_j = -w_j \Gamma_1 P \tilde{x}_j x_j^\top, \qquad \dot{\hat{B}}_j = -w_j \Gamma_2 P \tilde{x}_j u_j^\top, \qquad \dot{\hat{f}}_j = -w_j \Gamma_3 P \tilde{x}_j. \tag{6.35}$$

Note that the parameter identifiers with performance-based weighting yield very promising simulation results. Nevertheless, the algorithm turns out to be surprisingly resistant to mathematical analysis and a proof of convergence is still missing [81]. One problem is that it is difficult to predict which estimate converges to which subsystem. While this assignment was fixed in the case of known state-space partitions in Chapter 4, it constitutes an additional combinatorial uncertainty in the above adaptation laws. In Chapter 4, it was guaranteed that $\hat{A}_i \to A_i$ for $t \to \infty$. With (6.35), however, it may well happen that $\hat{A}_i \to A_j$ for $t \to \infty$ with $i \neq j$. Hence, traditional Lyapunov-based stability proofs are unable to deal with this ambiguity.

Furthermore, the algorithm has some minor dependence on initial conditions. That is, when two models are initialized with exactly the same parameters or converge to equal parameters during adaptation, they remain equal for all times as they also result in the same performance indices. Therefore, showing parameter convergence under (6.35) remains a challenging task for future work.

6.4 Simulation Studies

In this section, the proposed algorithms are applied in three test cases. First, the subsystem parameters of an exemplary PWA system are estimated with the redistribution algorithm. Afterwards, the redistribution algorithm is used to identify a PWA model of a nonlinear hybrid two-tank system. Finally, the use of parameter identifiers with performance based weighting is demonstrated for a simple switched linear system.

6.4.1 Performance-based Redistribution – PWA System

Consider the exemplary PWA system with three subsystems given by

$$\begin{aligned}
\theta_1 &= \begin{bmatrix} 0 & -2 & 1 & -1 & 0 & 1.5 & 0 & 0.4 \end{bmatrix}^\top, & \Omega_1 &= \left\{ x \in \mathbb{R}^2 : \ -2.5 \leq x_1 \leq 2.5 \right\}, \\
\theta_2 &= \begin{bmatrix} 0 & -2.5 & 1 & -1 & 0 & 1.5 & 0 & 0.2 \end{bmatrix}^\top, & \Omega_2 &= \left\{ x \in \mathbb{R}^2 : \ x_1 > 2.5 \right\}, \\
\theta_3 &= \begin{bmatrix} 0 & -1.5 & 1 & -1 & 0 & 1.5 & 0 & -0.3 \end{bmatrix}^\top, & \Omega_3 &= \left\{ x \in \mathbb{R}^2 : \ x_1 < -2.5 \right\},
\end{aligned}$$

which was previously investigated in [274]. Let the system be excited by a sinusoidal signal $u(t) = \bar{u}(t) + \sin(0.5t) + \sin(3t) + \sin(11t)$, where $\bar{u}$ is piecewise constant and switches every 25 s between 0, -4 and 6 to drive the system into the three regions Ω_i. Gaussian noise $\mathcal{N}(0, 0.01)$ is added to the state measurements.

Algorithm 1, according to Theorem 6.2, is guaranteed to converge to the true parameters in the noiseless case. Initially, a total of $N = 500$ models $\hat{\theta}_j$ are randomly distributed such that each element of the parameter vector $\hat{\theta}_j$ is bounded by ± 10. With design parameters $\alpha = 0, \beta = 1, \gamma = 0$ and $A_\mathrm{m} = -I_2$, the algorithm redistributes a subset of $M = 400$ models 300 times. Each time interval, in which the performance of the models is evaluated, lasts

between 2.5 s and 5 s. Fixed models which are not chosen for $T_{\max} = 125\,\mathrm{s}$ are set free according to Section 6.2.3.

In order to analyze the proposed identification algorithm with randomized redistribution, a series of Monte Carlo simulation, each with 100 trials, was performed. Figure 6.2 shows the results. The figure displays, for the first 100 out of 300 iterations, the norm of the minimum distance between models and subsystem parameters, i.e.

$$\tilde{\theta}_{i,\min} = \min_{j=1,\ldots,N} \|\theta_i - \hat{\theta}_j\|_2.$$

The average value over 100 trials is given by the center line while the standard deviation is visualized by the shaded area. The blue plots on the left were obtained with a redistribution factor $\delta_r = 0.2$ whereas $\delta_r = 1.2$ was used for the red plots on the right. As can be seen from the figure, the choice of δ_r has a distinct effect on the convergence. For larger values of δ_r, the estimates approach the true values faster. For lower values of δ_r, the convergence is slower but ultimately yields a better result. This is also shown by the detailed values in Table 6.1. For $\delta_r = 0.2$ and without previous knowledge about the system, the averaged estimates over the 100 Monte Carlo simulations are

$$\begin{aligned}
\hat{\theta}_1 &= \begin{bmatrix} 0.0001 & -1.9884 & 0.9998 & -0.9999 & -0.0001 & 1.5003 & 0.0002 & 0.3825 \end{bmatrix}^\top \\
\hat{\theta}_2 &= \begin{bmatrix} -0.0004 & -2.5016 & 1.0005 & -1.0018 & 0.0009 & 1.4995 & -0.0029 & 0.2466 \end{bmatrix}^\top \\
\hat{\theta}_3 &= \begin{bmatrix} -0.0009 & -1.4844 & 0.9982 & -0.9936 & 0.0017 & 1.4933 & 0.0022 & -0.2604 \end{bmatrix}^\top .
\end{aligned}$$

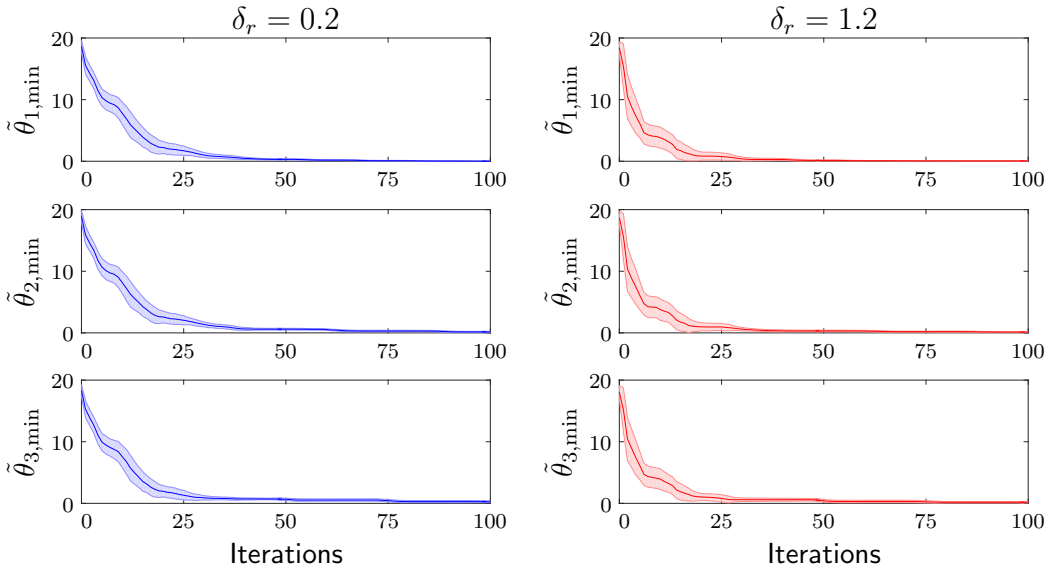
In Section 6.2.3, it was pointed out that the proposed algorithm can incorporate previous knowledge about the system. To show this, the Monte Carlo simulations are repeated with previous knowledge about the following elements of the parameter vectors

$$\theta_i(1) = \theta_i(5) = \theta_i(7) = 0 \quad \text{and} \quad \theta_i(3) = 1, \forall i.$$

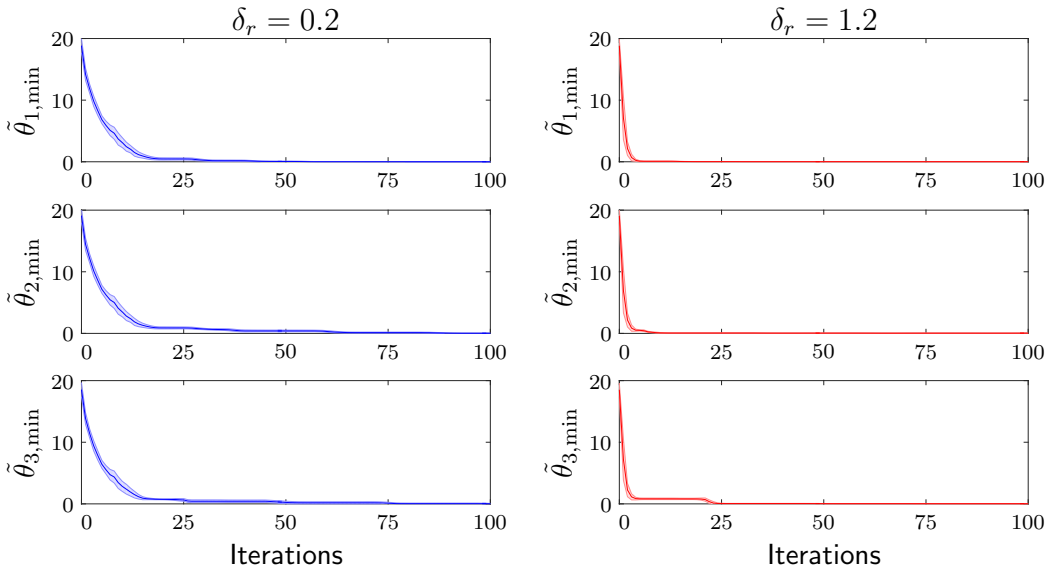
The averaged results of this second series of Monte Carlo simulations are shown in Fig. 6.2(b). Comparing Fig. 6.2(b) and Fig. 6.2(a), it can be concluded that the utilization of previous knowledge yields a faster convergence, independent of the redistribution factor δ_r. As before, smaller values of δ_r cause slower convergence but more precise estimates. Also see Table 6.1 for more details on the obtained estimates with and without previous

Tab. 6.1: Averaged results of 100 Monte Carlo simulations.

	no prev. knowl. $\delta_r = 0.2$	no prev. knowl. $\delta_r = 1.2$	prev. knowl. $\delta_r = 0.2$	prev. knowl. $\delta_r = 1.2$
$\tilde{\theta}_{1,\min}$	0.0059	0.0343	0.0040	0.0139
$\tilde{\theta}_{2,\min}$	0.0159	0.0829	0.0105	0.0396
$\tilde{\theta}_{3,\min}$	0.0107	0.0666	0.0064	0.0249
# fixed models	27.12	17.91	28.53	21.23



(a) Without previous knowledge.



(b) With previous knowledge.

Fig. 6.2: Identification result of Monte Carlo Simulation (a) without and (b) with previous knowledge about the system.

knowledge.

Table 6.1 furthermore states that the average number of fixed models after 100 iterations lies between 17 and 29 models. In the final step of the algorithm the fixed models are clustered to the desired or estimated number of subsystems. Figure 6.3 visualizes this final clustering step for one trial of the Monte Carlo simulation with previous knowledge. In this particular case, 35 fixed models are clustered into the three subsystem parameter vectors.

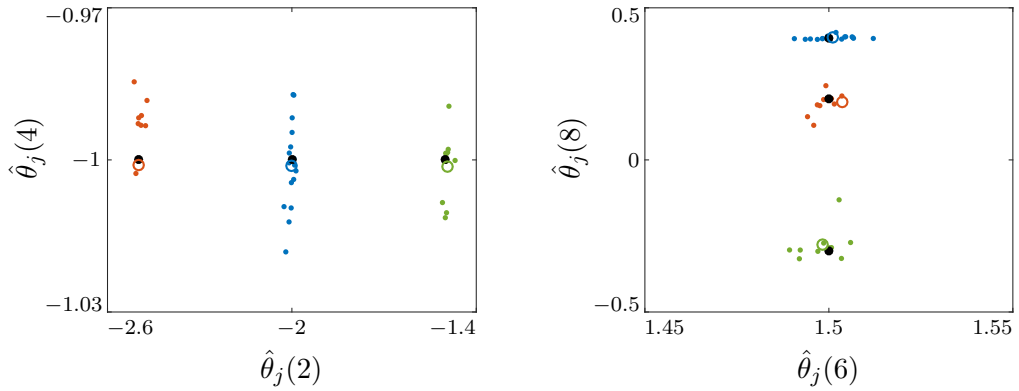


Fig. 6.3: Clustering of the fixed models at the end of one Monte Carlo simulation with previous knowledge.

6.4.2 Performance-based Redistribution – Hybrid Two-Tank System

We apply Algorithm 1 to identify the hybrid two-tank system considered in Section 5.4.4 (see Fig. 5.9). For the identification task, the inflow u consists again of two sinusoidal signals with frequencies 0.4 rad/s and 1 rad/s and a ramp-like offset. The offset consists of 4 elements, each 1000 s long. First the offset rises from $0.2u_{\max}$ to $0.9u_{\max}$, then stays there, then gradually falls to $0.2u_{\max}$ and stays there before the sequence repeats. The maximum inflow for which the tanks will not overflow is $u_{\max} = 0.127\,\mathrm{m^3s^{-1}}$. Gaussian noise $\mathcal{N}(0, 0.001)$ is added to the measured tank levels.

The length T of the time frames, in which the model performances are evaluated, is chosen randomly between 25 s and 50 s. In combination with the design parameters $A_{\mathrm{m}} = -0.01I_2$, $\alpha = \gamma = 0$ and $\beta = 1$, this produces a positive definite matrix $\mathcal{W}(T)$ for each time frame. Let the model set consist of $N = 5000$ models from which $M = 4500$ models are redistribute with the redistribution factor $\delta_r = 0.1$ at each iteration. Note that through the use of the state sharing approach discussed in Section 6.2.3, the dynamics of these 5000 models can be simulated with only 18 states. Furthermore, let unused models be set free according to Section 6.2.3 after $T_{\max} = 7000$ s.

Practical insights into the two-tank system enables the incorporation of previous knowledge about the system. It is clear that the inflow u does not directly affect the level in the right tank. Hence, the corresponding entry in the input matrices B_i (or equivalently the parameter vectors θ_i) can be set to zero in all iterations. With previous knowledge, the redistribution simplifies according to Section 6.2.3.

After a total of 10000 iterations, we obtain 5 clustered models. In the next step, the switching hyperplanes, which partition the state space for these 5 models, are estimated based on the approach proposed in Chapter 3 with design parameters $T_\delta = T_\Delta = 5$, $S_\Delta = X_\Delta = 0.05$ and L_i such that the poles of the observers $\hat{A}_i - L_i I_2$ lie at -5 and -5.5. The hyperplane estimation reveals that only four of the five models are activated. The inactive model may be due to outliers or fictitious models in the redistribution algorithm. Figure 6.4 shows the obtained state-space partitions and sampled data points of the state-space trajectory during identification. The data points are color-coded according to the polyhedral region they belong to. The reconstructed partitions resemble an intuitive guess

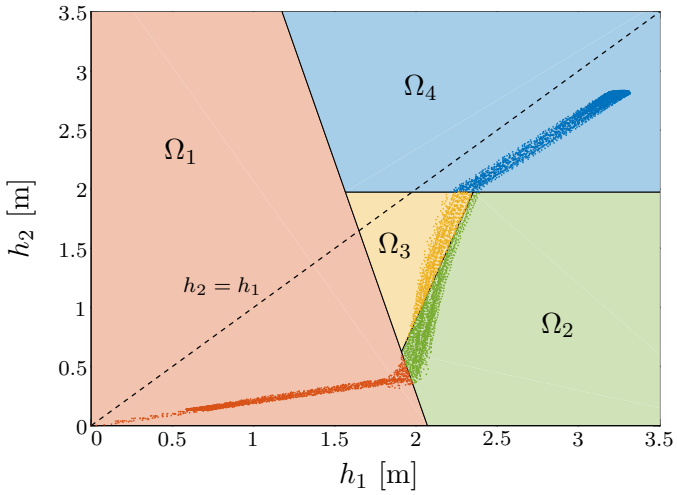


Fig. 6.4: Visualization of the obtained state-space partitions and the state-space trajectory which was used to identify the hybrid two-tank system. The sampled trajectory is color-coded based on the state-space partitions.

that the linearized dynamics should change depending on whether the tank levels are above or blow the connecting pipe at height 2 m. Furthermore, we see that the critical region of the state space in which $h_1 > h_c$ and $h_2 < h_c$ is approximated by the two subsystems $\hat{\theta}_2$ and $\hat{\theta}_3$ with regions Ω_2 and Ω_3. For the cases when both levels are below or above the connecting pipe, the single subsystems $\hat{\theta}_1$ and $\hat{\theta}_4$ are sufficient to approximate the nonlinear dynamics, respectively.

In order to validate the obtained PWA model, we perform the same validation experiment as in Section 5.4.4, where parameter identifiers with Concurrent Learning were evaluated. That means the PWA model is excited with the input signal u_{val} in Fig. 6.5 and the resulting trajectory is compared to the response of the actual nonlinear hybrid two-tank system. The results in Fig. 6.5 indicate a good approximation of the nonlinear system.

In addition, it is interesting to compare the obtained results with the Concurrent-Learning-based subsystem identification. For Concurrent Learning in Chapter 5, the designer chose to partition the state space into 8 regions as shown in Fig. 5.10 which led to the validation results shown in Fig. 5.11. Comparing both approaches, we can conclude that the identification with Multiple Models-based redistribution yields a PWA model with similar approximation capabilities as before but lower number of subsystems. This is linked to the fact that the algorithms proposed in this chapter are capable of identifying both subsystem dynamics and switching hyperplanes of PWA models with unknown state-space partitions.

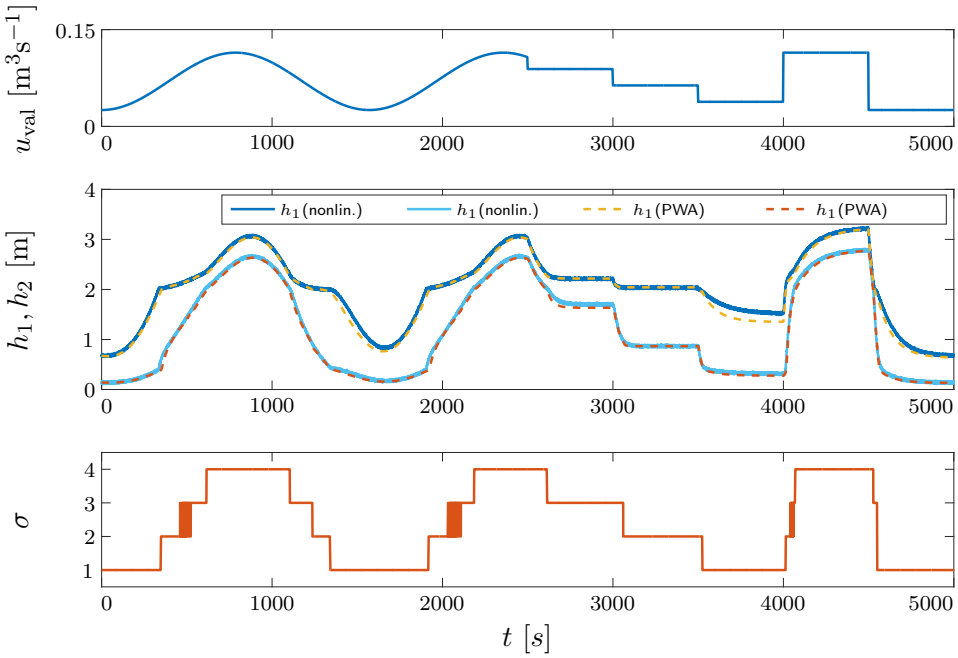


Fig. 6.5: Comparison of the hybrid two-tank system with the obtained PWA model for a validation input u_{val}. The PWA model follows the state-dependent switching signal σ.

6.4.3 Parameter Identifiers with Performance-based Weighting

To evaluate the parameter identifiers with performance-based weighting according to Section 6.3, consider the switched linear system

$$\dot{x} = a_\sigma x + b_\sigma u, \tag{6.36}$$

with subsystem parameters $a_1 = -b_1 = -1$, $a_2 = -b_2 = -2$ and $a_3 = -b_3 = -3$. Let the exogenous switching signal $\sigma(t)$ switch between the three subsystems every 50 s. Gaussian noise $\mathcal{N}(0, 0.01)$ is added to the state measurements.

We initialize three models $\hat{\theta}_j = [\hat{a}_j,\ \hat{b}_j]^\top$ with 10% of their nominal values, i.e., $0.1\theta_i$. This initialization ensures distinct performance signals for each model without placing the estimates too close to their nominal values. Then, the adaptation according to (6.35) is carried out with design parameters $\Gamma_1 = \Gamma_2 = 1$, $\alpha = 1$, $\beta = 0.1$, $\gamma = 10$ and $A_{\text{m}} = -1$. Figure 6.6 shows the estimated parameters $\hat{a}_j$ and $\hat{b}_j$, and the evolution of the weights w_j, $j = 1, \ldots, 3$. Besides showing a successful converge to all three subsystem parameters, the figure furthermore allows two interesting observations. First, we note that $\hat{\theta}_1 \to \theta_3$, $\hat{\theta}_2 \to \theta_2$ and $\hat{\theta}_3 \to \theta_1$. Hence, it can not be guaranteed beforehand that the i-th model converges to the i-th subsystem. This fact constitutes one of the main difficulties in proving convergence of the discussed algorithm. The second observation is related to the weight w_j. During the interval from 200 s to 800 s, the algorithm is still indecisive about the assignment of models to subsystems, which can be seen by the unorganized fluctuation of all the weights. Only after 800 s, the evolution of w_1, w_2 and w_3 is related to σ leading to the assignment mentioned above.

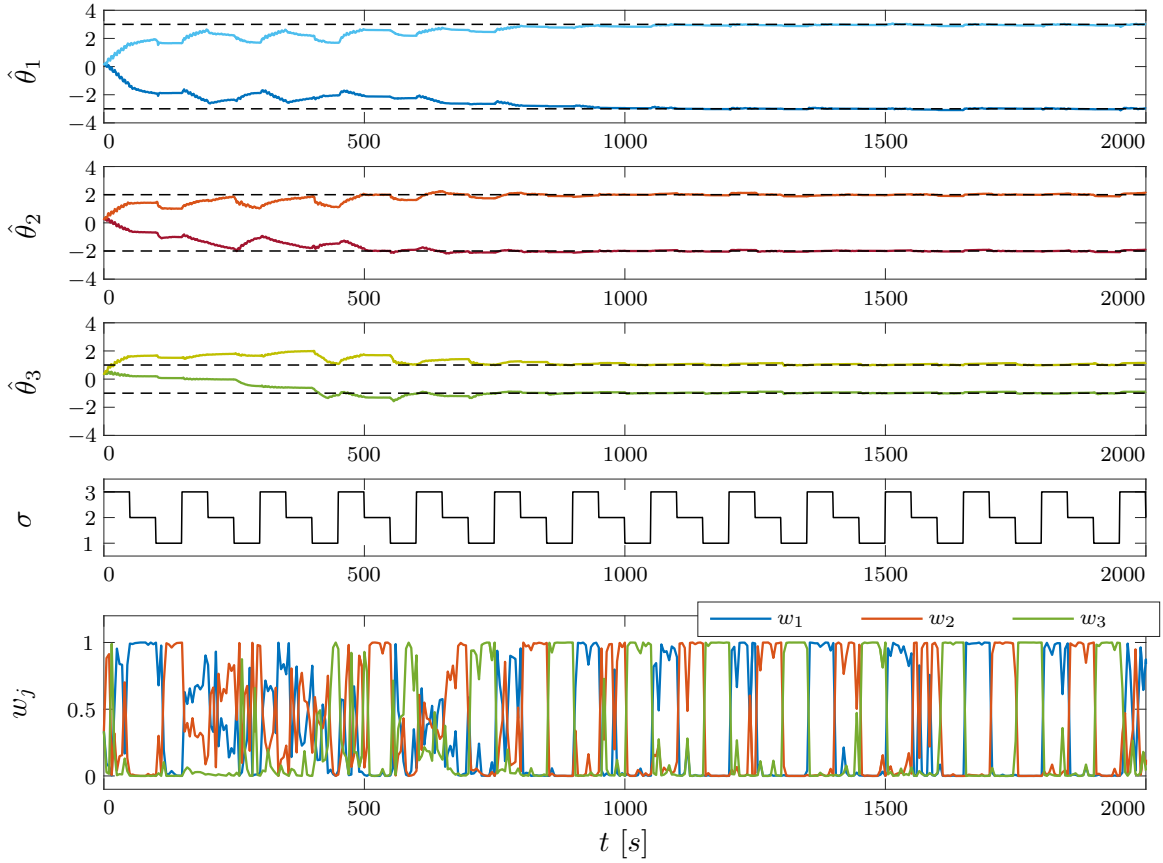


Fig. 6.6: Identification of a switched system with parameter identifiers and performance-based weighting.

6.5 Summary

This chapter presents two identification algorithms for continuous-time PWA systems in state-space form with unknown switching hyperplanes. Both algorithms are derived from insights on the performance signals in Multiple Models Adaptive Control. The performance signals are defined in terms of the integral over past prediction errors, which were also used in parameter identifiers, with exponential forgetting. By rearranging the state-space system dynamics in form of a single parameter vector instead of multiple system matrices, it was shown that the scalar performance signals also correspond to the square of a weighted vector norm of the parameter error between estimated parameters, used for the prediction, and the true parameters, if the system is persistently excited. This insight is exploited in both algorithms in different ways. The relationship between performance signals and parameter errors can only be derived for systems with full state-measurements, which restricts the class of applicable systems.

The first algorithm is related to particle swarms. A large set of models with different parameter vectors predicts the system state in parallel and calculate their performance signals. It was shown that the state sharing concept allows simulation of an infinite number of models in parallel based on a finite number of states. A subset of models is repeatedly

redistributed based on the obtained performance signals, which incorporate information about the distance to the actual parameters. It was concluded that the proposed, goal-oriented redistribution provides better parameter estimates over time and is guaranteed to converge to the true parameter vectors with probability one. Two limitations of the algorithm were pointed out. First, the algorithm puts a dwell time constraint on the switching between subsystems, to ensure that the same subsystem is active in two consecutive redistribution steps. Second, the required number of models grows exponentially with the dimension of the parameter space. It was proposed to compensate the second limitation by incorporating previous knowledge about the system, i.e. fixing known parameters to the true values. Additional benefits of incorporating previous knowledge are more precise estimates and faster convergence, which was analyzed in a Monte Carlo Simulation of an exemplary PWA system. The proposed algorithm may be applied both offline and online and was tested on a hybrid two-tank system. Based on the derived subsystem parameters, it is possible to estimate the switching hyperplanes with the algorithm outlined in Chapter 3.

The second algorithm transforms the performance signals into weights and incorporates them into the parameter identifiers presented in Chapter 4. This modification enables the application of parameter identifiers for switched and PWA systems with unknown switching signals and switching hyperplanes, respectively. A simplistic understanding of the approach is to adapt strongest the parameter configuration closest to the currently active ones. While simulation studies show convergence of the estimates in certain cases, there is no formal proof for it. Furthermore, convergence relies on a reasonable initialization.

For future work, it is thus on the one side desirable to find a convergence proof for parameter identifiers with performance-based weighting. On the other side, the performance-based redistribution algorithm may be modified to eliminate the dwell-time constraint.

Part II

Adaptive Control of Piecewise Affine Systems

7 Model Reference Adaptive Control for PWA Systems

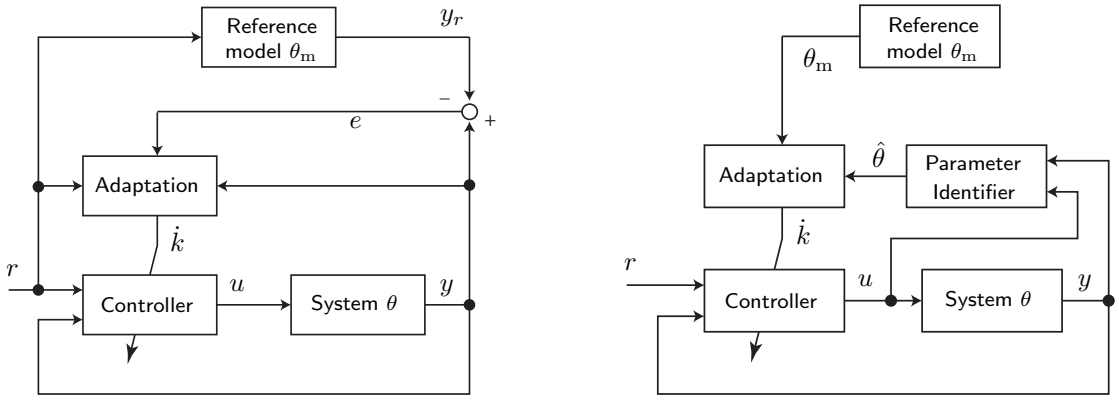
The first part of this thesis focused on the adaptive identification of PWA systems. The task was to recursively estimate the unknown system parameters based on continuous input and state measurements. In that context, it was assumed that the applied excitation in form of sufficiently rich input signals ensures stability of the unknown system. Given that the system parameters are unknown, however, the design of a stabilizing controller is a non-trivial problem, which constitutes the central motivation for adaptive control.

Generally speaking, adaptive control algorithms are applied in case a single fixed controller is unable to deliver the desired performance. This may for instance be due to large uncertainties in parameters of the system, such that no single, fixed robust controller can stabilize every possible system configuration. Another motivation are slowly time varying system parameters caused for example by aging or wear of the controlled system. In order to cope with such unknown or time varying parameters, the parameters of an adaptive controller are adjusted to match the current system parameters. Hence, a central task in adaptive control is to design update laws for control parameters which ensure stability of the controlled system. More than half a century of research on adaptive control led to various adaptive control concepts summarized in standard textbooks [6, 114, 144, 175, 240].

The focus of this chapter is on extending the concept of *Model Reference Adaptive Control (MRAC)* from linear systems to PWA systems. In MRAC, the desired closed-loop dynamics are imposed in form of a reference model. With suitably chosen update laws, the control gains are to be tuned automatically in such a way that the controlled systems follows the behavior of the reference model. Two approaches to achieve this goal are depicted in Fig. 7.1. In *direct MRAC*, sketched in Fig. 7.1(a), the output signal of the reference model is compared to the output signal of the controlled system and the control gains are directly updated based on this difference. In *indirect MRAC*, which is illustrated in Fig. 7.1(b), the first step is to estimate the unknown system parameters. One way to do so is with the help of parameter identifiers presented in the Chapter 4. The estimated parameters are continuously compared to the desired parameters of the reference model in order to extract information for the adaptation of the control gains.

Model reference adaptive control forces the unknown system to track the specified reference system. For the hybrid or switched systems considered in this thesis, MRAC hence bears the potential to remove undesired switching from the open-loop systems by choosing a suitable reference system. More precisely, a PWA system whose parameters are unknown can be controlled with MRAC in such a way that the resulting closed-loop dynamics are linear. Due to the complications arising from switching, this constitutes a highly desirable property. Existing approaches towards this goal can be divided into MRAC for general switched systems and MRAC for PWA systems.

For the general class of switched systems various adaptive controllers have been proposed



(a) Direct MRAC **(b)** Indirect MRAC

Fig. 7.1: Comparison between (a) direct adaptive control, where controller parameters k are directly tuned based on the reference signals r, the outputs y and the tracking error e; and (b) indirect adaptive control, where adaptation is based on the difference between estimated and desired system parameters.

in recent years [259, 260, 264, 265]. A model reference robust adaptive control algorithm for uncertain switched linear systems is proposed in [259]. The proof of uniform boundedness under arbitrary switching is based on multiple Lyapunov functions constructed from a sufficient LMI condition. Furthermore, a dead band is introduced in order to suppress the bursting phenomenon in adaptive systems. A critical assumption in MRAC for switched systems is exact knowledge of the switching signal. The case in which the reference system and controller do not switch synchronously with the plant is investigated in [264]. Dwell-time constraints are derived that ensure practical global stability of the switched error system, which consists of the interconnection of plant, reference model and adaptive controller. While work in [264] is restricted to switched linear systems, [265] extends the proposed framework to nonlinear switched systems with nonlinear reference systems. This extension enables more sophisticated performance requirements as shown by the example of a highly maneuverable aircraft. Wang and Zhao point out that deriving dwell-time constraints for switched systems may be infeasible without knowledge of the actual subsystem parameters [260]. Instead, their adaptive state tracking algorithm is based on a hyperstability criterion which relates to passivity. The proposed control, however, assumes that also the switching signal of the system can be controlled. This is unsuitable for PWA systems where the switching is always state-dependent.

This brings us to the class of PWA systems, for which there are various MRAC approaches available in the literature [64, 65, 68–70, 216–219]. Early work by di Bernardo *et al.* extends the minimal control synthesis algorithm to bimodal PWA systems [64, 68, 69]. Later, di Bernardo *et al.* proposed a hybrid MRAC strategy for multimodal PWA systems [70] in which the state-input space partitioning of the controlled system may be different form the reference system. Excluding sliding modes, stability in [70] was proven by passivity-based arguments as well as Lyapunov theory. For the Lyapunov-based proof, a common Lyapunov-function of the reference system's subsystems is required. The extension of this hybrid MRAC algorithm in [65] is concerned with potential instabilities of the original algorithms due to disturbances at the input in form of the affine terms of the

plant and reference system. The solution is based on a common Lyapunov function and can guarantee convergence of the error system even when the system enters a sliding mode. Note that the referenced work by di Bernardo *et al.* focuses on single input systems in control canonical form which restricts the class of applicable systems.

Work by Sang and Tao focuses on MRAC for piecewise linear (PWL) systems which approximate nonlinear systems by linearization at multiple operating points [216–219]. The state tracking algorithm proposed in [216, 217] ensures bounded signals and asymptotic tracking for arbitrary fast switching in case a common quadratic Lyapunov function exists. In case no such Lyapunov function exists, signals are shown to be bounded, and an upper bound on the tracking error is given. Asymptotic tracking can be ensured through parameter projection combined with persistent excitation and slow switching. A more detailed analysis of the proposed algorithm is given in [218] with careful treatment of various special cases related to the existence of a common Lyapunov function or the level of excitation in the reference signal. A multivariable extension of the algorithm is proposed in [219]. Finally, the dissertation by Sang [215] provides a nice overview of the above mentioned works on MRAC for piecewise linear systems.

Despite the revised list of publications on adaptive control of switched systems, there still is a lack of MRAC algorithms for multivariable PWA systems [241]. The contribution in this chapter is to fill this gap by means of two approaches: One direct and one indirect MRAC algorithm. First, we build upon the direct approach by Sang and Tao [218] to define direct MRAC for PWA systems. Especially for parameter convergence, the additional affine terms require further analysis. By adding one additional assumption and revising the proof for persistence of excitation, all gains are guaranteed to converge to their nominal values. This extension provides great advantages compared to the state of the art. It will be shown that working with affine subsystems requires less previous knowledge about the system. As opposed to [216–219], this enables application of the algorithm to systems with uncertain or unknown operating points. Then, we derive an indirect MRAC algorithm which is based on our previous work on parameter identifiers in [276] or Chapter 4. The unique advantage of this approach is that estimates of all system parameters are obtained in parallel to the control task. For both the direct and the indirect approach, the stability proofs for asymptotic state tracking rely on Lyapunov theory and dwell-time assumptions. Moreover, parameter convergence is guaranteed under the assumption of persistent excitation (PE). Therefore, this chapter presents the first MRAC law for multivariable PWA systems, which overcomes restrictions to PWL systems, single-input systems or systems in control canonical form, postulated by state of the art algorithms [64, 65, 68–70, 216–219].

This chapter is an excerpt of [280] and structured as follows. Section 7.1 introduces the applied control structure including controller and reference system, and formulates the MRAC problem. The direct and indirect MRAC algorithms are presented in Section 7.2 and Section 7.3, respectively. In both cases, we consider reference system tracking separately from the additional requirement of parameter convergence. The application of direct and indirect MRAC is demonstrated in Section 7.4 by means of numerical examples. Finally, Section 7.5 concludes the chapter.

7.1 The Model Reference Adaptive Control Problem

This section begins with formally introducing the MRAC problem for PWA systems and concludes with some preliminaries on the Filippov concept for sliding mode solutions.

We consider the PWA system with full state measurement and indicator functions χ_i as introduced in (2.3) of Section 2.1. Hence, let the state-input space be partitioned by $s \in \mathbb{N}$ polyhedral regions Ω_i and assign each region an affine state-space system from the family $\{A_i, B_i, f_i\}_{i=1}^{s}$, where $A_i \in \mathbb{R}^{n\times n}$, $B_i \in \mathbb{R}^{n\times p}$ and $f_i \in \mathbb{R}^n$ are the unknown, constant system parameters. The pairs (A_i, B_i) are assumed to be controllable and each B_i be of full column rank. For the common case $p \leq n$, the full column rank is usually satisfied by physical systems. The region which contains the current state-input vector determines which subsystem is active. This is expressed with the indicator functions

$$\chi_i(t) = \begin{cases} 1 \text{ , if } (x(t), u(t)) \in \Omega_i, \\ 0 \text{ , otherwise.} \end{cases} \tag{7.1}$$

As the regions Ω_i do not overlap, it follows that $\sum_{i=1}^{s} \chi_i(t) = 1$ and $\chi_i(t)\chi_j(t) = 0,\ i \neq j$. The dynamics of the PWA system are then given by

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) + f(t), \tag{7.2}$$

where $A(t) = \sum_{i=1}^{s} A_i\chi_i(t)$, $B(t) = \sum_{i=1}^{s} B_i\chi_i(t)$ and $f(t) = \sum_{i=1}^{s} f_i\chi_i(t)$. Hence, in the following, a matrix without index refers to the currently active matrix, i.e., $A \to A_i$ with $\chi_i = 1$.

7.1.1 Reference System Modeling

The goal in MRAC is to control the unknown or uncertain system in such a way that it exhibits a desired behavior. In MRAC, the desired behavior is given in form of a reference system, which is chosen by the designer. Adaptive control laws – to be developed in this chapter – then adjust the control gains such that the controlled system tracks the reference system despite unknown or uncertain parameters.

The *reference system* is chosen to be piecewise affine in order to give enough flexibility to the designer. Let the reference system and the controlled PWA system share the same state-space partitions Ω_i. Hence, we consider reference systems of the form

$$\dot{x}_\mathrm{m} = A_\mathrm{m}x_\mathrm{m} + B_\mathrm{m}r + f_\mathrm{m}, \tag{7.3}$$

where $x_\mathrm{m} \in \mathbb{R}^n$ and $r \in \mathbb{R}^p$ are the state of the reference system and the reference signals, respectively. The parameters of the reference system switch according to the same indicator functions χ_i as the controlled system in (7.1) and are therefore given by $A_\mathrm{m} = \sum_{i=1}^{s} A_{\mathrm{m}i}\chi_i$, $B_\mathrm{m} = \sum_{i=1}^{s} B_{\mathrm{m}i}\chi_i$ and $f_\mathrm{m} = \sum_{i=1}^{s} f_{\mathrm{m}i}\chi_i$ with $A_{\mathrm{m}i} \in \mathbb{R}^{n\times n}$, $B_{\mathrm{m}i} \in \mathbb{R}^{n\times p}$ and $f_{\mathrm{m}i} \in \mathbb{R}^n$, $\forall i = 1, \ldots, s$.

Note that simply applying the partitions of the controlled PWA system also for the reference system would unnecessarily limit the choice of reference systems. Take for example

the case in which the designer wants to introduce certain thresholds into the behavior of the closed-loop system at which the dynamics are supposed to change. In that case, the reference system may need to feature a switch at states where the controlled system is continuous. Hence, to gain additional freedom, an original partition Ω_i of the controlled PWA system may be divided into convex subsets Ω_k and Ω_l with $\Omega_i = \Omega_k \cup \Omega_l$. While the parameters of the controlled PWA system remain the same in both regions, e.g. $A_k = A_l = A_i$, the reference system parameters can be chosen differently in both sets, i.e. $A_{\mathrm{m}k} \neq A_{\mathrm{m}l}$. In practice, this enables to switch the reference system independently from the controlled system.

Stability of the reference system (7.3) is essential for the tracking problem. Assuming that each subsystem $A_{\mathrm{m}i}$ is stable, it follows that there exists a symmetric, positive definite Lyapunov matrix $P_i \in \mathbb{R}^{n\times n}$ for every symmetric, positive definite matrix $Q_{\mathrm{m}i} \in \mathbb{R}^{n\times n}$ such that

$$A_{\mathrm{m}i}^\top P_i + P_i A_{\mathrm{m}i} = -Q_{\mathrm{m}i}, \quad \forall i = 1, \ldots, s. \tag{7.4}$$

Hence, $V_i = x_{\mathrm{m}}^\top P_i x_{\mathrm{m}}$ is a quadratic Lyapunov function for the i-th subsystem of the reference system. For each subsystem i, there also exist constants $a_{\mathrm{m}i}, \lambda_{\mathrm{m}i} > 0$ such that $\|e^{A_{\mathrm{m}i}t}\| \leq a_{\mathrm{m}i} e^{-\lambda_{\mathrm{m}i}t}$. If all subsystems satisfy (7.4) for a common matrix P, i.e., $P_i = P, \forall i$, then $V = x_{\mathrm{m}} P x_{\mathrm{m}}$ is referred to as a *common quadratic Lyapunov function (CQLF)*. According to Theorem A.1, a reference system (7.3) with CQLF is stable, even for arbitrary fast switching. If, however, no CQLF exists, stability of the reference system (7.3) can be ensured through dwell-time considerations as revised in Section A.1.4. In other words, the reference system must reside in one mode for a sufficiently long (dwell) time T_0. The dwell-time constraint for the switched reference system (7.3) is captured in the following Lemma, where $\lambda_{\min}(P_i)$ and $\lambda_{\max}(P_i)$ refer to the minimum and maximum eigenvalue of P_i.

Lemma 7.1 (Dwell-time constraint for reference system) "The switched system $\dot{x}(t) = A_{\mathrm{m}}(t)x(t)$ is exponentially stable with decay rate $\sigma \in (0, 1/2\alpha)$ if T_0 satisfies

$$T_0 \geq T_{\mathrm{m}} = \frac{\alpha}{1 - 2\sigma\alpha} \ln(1 + \mu\Delta_{A_{\mathrm{m}}}), \quad \mu = \frac{a_{\mathrm{m}}^2}{\lambda_{\mathrm{m}}\beta} \max_i \|P_i\|,$$

where $\Delta_{A_{\mathrm{m}}} = \max_{i,j} \|A_{\mathrm{m}i} - A_{\mathrm{m}j}\|$, $\alpha = \max_i \lambda_{\max}(P_i)$, $\beta = \min_i \lambda_{\min}(P_i)$, $a_{\mathrm{m}} = \max_i a_{\mathrm{m}i}$ and $\lambda_{\mathrm{m}} = \max_i \lambda_{\mathrm{m}i}$." [218, Lem. 1]

Proof: The detailed proof of Lemma 7.1 can be found in [218]. Here, it suffices to point out that the proof is based on the Lyapunov function $V = x^\top \left(\sum_i^s P_i \chi_i\right) x$. The indicator functions χ_i cause discontinuities in the Lyapunov function at every switching time t_k. For later analysis in this chapter, the following approximation of the potential increase in V associated with each switch is of greater importance:

$$V(t_k) \leq (1 + \mu\Delta_{A_{\mathrm{m}}}) V(t_k^-), \tag{7.5}$$

where $V(t_k^-)$ and $V(t_k)$ are the values of the Lyapunov function immediately before and after the switch, respectively. □

Finally, given that the autonomous switched system $\dot{x} = A_{\mathrm{m}}(t)x(t)$ is exponentially stable for a dwell-time T_0, it can be concluded, with the same dwell-time constraint, that the state of the switched reference system (7.3) remains bounded for bounded reference signals [168].

7.1.2 Problem Formulation and Controller Design

The first problem addressed in this chapter reads as follows.

Problem 7.2 (State tracking) Given a PWA system (7.2) with known regions Ω_i and unknown subsystem parameters A_i, B_i and f_i, design a state feedback control law $u(t)$ which stabilizes the PWA system and forces the state $x(t)$ to asymptotically track the state $x_{\mathrm{m}}(t)$ generated by the reference system (7.3).

In the following two sections, a direct and an indirect approach to Problem 7.2 are derived. In both cases, the *control law* takes the form

$$u(t) = K_x(t)x(t) + K_r(t)r(t) + k_f(t), \tag{7.6}$$

with *estimated control gains* $K_x \in \mathbb{R}^{p\times n}$, $K_r \in \mathbb{R}^{p\times p}$ and $k_f \in \mathbb{R}^p$. Note that there is a different set of control gains $\{K_{xi}, K_{ri}, k_{fi}\}$ for each subsystem i and the control gains need to switch in synchrony with the system parameters. Hence, the control gain switching relies on the same indicator functions χ_i as the controlled system and the reference system, i.e., $K_x = \sum_{i=1}^{s} K_{xi}\chi_i$, $K_r = \sum_{i=1}^{s} K_{ri}\chi_i$ and $k_f = \sum_{i=1}^{s} k_{fi}\chi_i$. Therefore, as common in related works [64, 65, 68–70, 216–219, 273, 274, 276], Ω_i and χ_i are assumed to be known throughout this chapter. Despite this technical assumption, the later simulation studies will suggest some degree of robustness against asynchronous switching. This also implies that, given the partitions of the PWA system are unknown, one may define a sufficiently large number of fictitious partitions, which deliver a satisfactory approximation of the actual switching behavior. In combination with the robustness to asynchronous switching, this can yield an acceptable control performance.

Controlling the PWA system (7.2) with the state feedback controller (7.6) yields the closed-loop system

$$\dot{x} = \sum_{i=1}^{s} \chi_i\Big((A_i + B_iK_{xi})x + B_iK_{ri}r + B_ik_{fi} + f_i\Big), \tag{7.7}$$

which in order to solve Problem 7.2 should equal (7.3) for suitably chosen control gains. Therefore, to ensure feasibility of the tracking problem, it is assumed, as usual, that there exist *nominal control gains* K_{xi}^*, K_{ri}^* and k_{fi}^* which fulfill the so-called *matching conditions*

$$A_{\mathrm{m}i} = A_i + B_iK_{xi}^*, \quad B_{\mathrm{m}i} = B_iK_{ri}^*, \quad f_{\mathrm{m}i} = f_i + B_ik_{fi}^*, \quad \forall i = 1, \ldots, s. \tag{7.8}$$

One practical implication of these matching conditions is that the structure of the controlled system needs to be known to some extend. Otherwise the designer cannot guarantee that the chosen reference systems yield feasible matching conditions.

The proposed direct and indirect adaptive control laws produce and update estimates of the nominal control gains in different ways, but both solve Problem 7.2. A well known advantage of MRAC is that solutions to Problem 7.2 need not converge to the nominal control gains. However, parameter convergence improves robustness and transient performance in adaptive systems [75, 142]. Hence, in cases where the reference signal is not periodic or where it is desirable to monitor the nominal parameters, the convergence of estimated parameters according to the following problem is required.

Problem 7.3 (Convergence of parameter estimates) Besides the tracking according to Problem 7.2, ensure that all estimated parameters converge to their nominal values.

The proposed algorithms will be shown to comply with Problem 7.3 under the assumption of suitable reference signals.

7.1.3 Sliding Mode Solutions

One important aspect in the problem formulation has not been considered so far. That is, if the vector fields of two neighboring regions both point towards the switching hyperplane, the system trajectories cannot simply cross from one region to the other. Instead, the state of the system is unable to leave the hyperplane and an additional mode – the sliding mode – is created. As the solutions of such sliding modes may evolve in a substantially different manner from the individual subsystems, they must be carefully examined.

In the Filippov concept [89], the sliding mode solution is described by a unique convex combination of the contributing vector fields. In order to obtain such a convex combination in the presented framework, replace all indicator functions χ_i by $\bar{\chi}_i$ with the modified property that $\bar{\chi}_i$ now takes values in the interval $[0, 1]$. That means the property $\chi_i \chi_j = 0$ for $i \neq j$ does not hold for sliding modes. But for convexity $\sum_{i=1}^{s} \bar{\chi}_i = 1$ still holds. In the later proofs of Theorem 7.4 and Theorem 7.7, the Filippov concept is applied to show that the presented results also hold in the presence of sliding modes.

7.2 Direct Model Reference Adaptive Control

As pointed out in Section 2.1.4, PWA systems are advantageous compared to PWL systems for MRAC as they do not require a priori knowledge about the operating points of the system. This is reflected in the fact that algorithms can operate directly on the measured state and control inputs instead of the difference to operating points. Hence, this section builds upon the framework in [218] and derives a direct MRAC algorithm for PWA systems. This generalization modifies the PE conditions in [218] and hence demands for an extended proof of parameter convergence.

We begin by defining errors between the current control gains and the nominal values: $\tilde{K}_{xi} = K_{xi} - K_{xi}^*$, $\tilde{K}_{ri} = K_{ri} - K_{ri}^*$ and $\tilde{k}_{fi} = k_{fi} - k_{fi}^*$. Let us rewrite the dynamics of the closed-loop system in terms of the reference system and the gain errors by adding and

subtracting the expression $B_i K_{xi}^* x + B_i K_{ri}^* r + B_i k_{fi}^*$ in (7.7), which yields

$$\begin{aligned}\dot{x} &= \sum_{i=1}^{s} \chi_i \Big((A_i + B_i K_{xi}^*)x + B_i K_{ri}^* r + f_i + B_i k_{fi}^* \\ &\qquad + B_i \big((K_{xi} - K_{xi}^*)x + (K_{ri} - K_{ri}^*)r + (k_{fi} - k_{fi}^*) \big) \Big) \\ &= A_{\mathrm{m}} x + B_{\mathrm{m}} r + f_{\mathrm{m}} + \sum_{i=1}^{s} \chi_i B_i \left(\tilde{K}_{xi} x + \tilde{K}_{ri} r + \tilde{k}_{fi} \right). \end{aligned} \tag{7.9}$$

Following from (7.9) and (7.3), the dynamics of the *tracking error* $e(t) = x(t) - x_{\mathrm{m}}(t)$ are

$$\dot{e} = A_{\mathrm{m}} e + \sum_{i=1}^{s} \chi_i B_i \left(\tilde{K}_{xi} x + \tilde{K}_{ri} r + \tilde{k}_{fi} \right). \tag{7.10}$$

If the control gains K_{xi}, K_{ri} and k_{fi} take the nominal values defined in (7.8), then the error dynamics in (7.10) reduce to $\dot{e} = A_{\mathrm{m}} e$, which indicates that the tracking error e converges to zero exponentially for a stable the reference system.

Consider now, however, the case in which the system parameters A_i, B_i and f_i are unknown, which at the same time implies unknown nominal control gains. We make the usual assumption in multivariable MRAC [241] that there exist known quadratic matrices $S_i \in \mathbb{R}^{p \times p}$ for which $K_{ri}^* S_i$ is symmetric and positive definite: $K_{ri}^* S_i = (K_{ri}^* S_i)^\top \succ 0$. Then, initial estimates $K_{xi}(0) = K_{xi0}$, $K_{ri}(0) = K_{ri0}$ and $k_{fi}(0) = k_{fi0}$ are introduced and updated over time. Lyapunov analysis – which will be conducted shortly – suggests the following *direct update laws* for the estimated control gains:

$$\begin{aligned} \dot{K}_{xi} &= -\Gamma_{xi} S_i^\top B_{\mathrm{m}i}^\top P_i e x^\top \chi_i, \\ \dot{K}_{ri} &= -\Gamma_{ri} S_i^\top B_{\mathrm{m}i}^\top P_i e r^\top \chi_i, \\ \dot{k}_{fi} &= -\Gamma_{fi} S_i^\top B_{\mathrm{m}i}^\top P_i e \ \chi_i, \end{aligned} \tag{7.11}$$

where Γ_{xi}, Γ_{ri}, $\Gamma_{fi} \in \mathbb{R}^+$ are positive scaling constants, and where $P_i \in \mathbb{R}^{n \times n}$ is the symmetric, positive definite Lyapunov matrix of the i-th reference system satisfying $A_{\mathrm{m}i}^\top P_i + P_i A_{\mathrm{m}i} = -Q_{\mathrm{m}i}$ for some symmetric, positive definite matrix $Q_{\mathrm{m}i} \in \mathbb{R}^{n \times n}$. The tracking abilities under the adaptation (7.11) are summarized in the following theorem.

Theorem 7.4 (Direct MRAC for PWA systems with CQLF) Consider a reference system (7.3) for which a common quadratic Lyapunov function with $P = P_i$, $\forall i$ is known. Let the PWA system (7.2) with known regions Ω_i be controlled by the state feedback (7.6) with gains updated according to (7.11). Then, the state of the PWA system asymptotically tracks the state of the reference system.

Proof: Consider the following candidate Lyapunov function, which is quadratic in the tracking error e and the deviations $\tilde{K}_{xi}$, $\tilde{K}_{ri}$, $\tilde{k}_{fi}$ from the nominal control gains:

$$V = \frac{1}{2} e^\top P e + \frac{1}{2} \sum_{i=1}^{s} \Big(\operatorname{tr}\left(\tilde{K}_{xi}^\top M_{si} \tilde{K}_{xi} \right) + \operatorname{tr}\left(\tilde{K}_{ri}^\top M_{si} \tilde{K}_{ri} \right) + \tilde{k}_{fi}^\top M_{si} \tilde{k}_{fi} \Big), \tag{7.12}$$

where $M_{si} = (K_{ri}^* S_i)^{-1} \in \mathbb{R}^{p\times p}$. As mentioned before, the variables S_i are chosen such that the inverse exists and is positive definite. The time derivative of (7.12) along (7.10) is

$$\dot{V} = e^\top \left(\frac{1}{2}\sum_{i=1}^{s} \chi_i (A_{\mathrm{m}i}^\top P + P A_{\mathrm{m}i})\right) e + \sum_{i=1}^{s} \left(\chi_i e^\top P B_i \left(\tilde{K}_{xi} x + \tilde{K}_{ri} r + \tilde{k}_{fi}\right)\right) + \sum_{i=1}^{s} \left(\mathrm{tr}\left(\tilde{K}_{xi}^\top M_{si} \dot{\tilde{K}}_{xi}\right) + \mathrm{tr}\left(\tilde{K}_{ri}^\top M_{si} \dot{\tilde{K}}_{ri}\right) + \tilde{k}_{fi}^\top M_{si} \dot{\tilde{k}}_{fi}\right). \quad (7.13)$$

At this point, the rationale behind the update laws in (7.11) can be seen as a suitable choice to cancel all terms in the last two lines of (7.13). Note that, for simplicity and without loss of generality, the scaling constants Γ_{xi}, Γ_{ri}, Γ_{fi} are neglected throughout the proof. Inserting the update laws (7.11) with $P_i = P$ in (7.13) yields

$$\dot{V} = -e^\top \left(\frac{1}{2}\sum_{i=1}^{s} Q_{\mathrm{m}i}\chi_i\right) e + \sum_{i=1}^{s} \chi_i \Big(e^\top P B_i \tilde{K}_{xi} x - \mathrm{tr}\left(\tilde{K}_{xi}^\top M_{si} S_i^\top B_{\mathrm{m}i}^\top P e x^\top\right) + e^\top P B_i \tilde{K}_{ri} r - \mathrm{tr}\left(\tilde{K}_{ri}^\top M_{si} S_i^\top B_{\mathrm{m}i}^\top P e r^\top\right) + e^\top P B_i \tilde{k}_{fi} - \tilde{k}_{fi}^\top M_{si} S_i^\top B_{\mathrm{m}i}^\top P e \Big), \quad (7.14)$$

for which the terms under the second summation cancel out. To see this, note that $M_{si} S_i^\top B_{\mathrm{m}i}^\top = M_{si} S_i^\top (K_{ri}^*)^\top B_i^\top = M_{si} M_{si}^{-1} B_i^\top = B_i^\top$ simplifies the first trace to $\mathrm{tr}(\tilde{K}_{xi}^\top B_i^\top P e x^\top)$. As $\mathrm{tr}(X) = \mathrm{tr}(X^\top)$, this is equivalent to $\mathrm{tr}(x e^\top P B_i \tilde{K}_{xi})$, which can be rewritten as $\mathrm{tr}(e^\top P B_i \tilde{K}_{xi} x)$ due to the property $\mathrm{tr}(XYZ) = \mathrm{tr}(YZX)$. Next, note that the obtained expression inside the trace operator is a scalar and cancels the corresponding term in (7.14). The same steps can be performed to cancel the other terms in (7.14). It follows that the derivative of V is negative semidefinite:

$$\dot{V} = -e^\top \left(\frac{1}{2}\sum_{i=1}^{s} Q_{\mathrm{m}i}\chi_i\right) e \leq 0. \quad (7.15)$$

Considering additionally the case of potential sliding modes, one must furthermore evaluate the derivative of V along the sliding mode solution. Hence, with the Filippov solution concept all convex combinations of the vector fields contributing to the sliding modes are analyzed by replacing $\chi_i \in \{0, 1\}$ with $\bar{\chi}_i \in [0, 1]$ in the above derivation. As in [65], doing so confirms negative semidefiniteness of V as $\dot{V} = -e^\top \left(\frac{1}{2}\sum_{i=1}^{s} Q_{\mathrm{m}i}\bar{\chi}_i\right) e \leq 0$. Note that this derivation shows stability of the sliding mode solution without determining the uniquely defined Fillipov vector field of the actual sliding mode.

In summary, the error system is shown to be stable under arbitrary fast switching and sliding modes, and the errors e, $\tilde{K}_{xi}$, $\tilde{K}_{ri}$ and $\tilde{k}_{fi}$ are bounded ($\in \mathcal{L}^\infty$). Boundedness of e, together with a stable reference system (i.e. $x_{\mathrm{m}} \in \mathcal{L}^\infty$), yields $x \in \mathcal{L}^\infty$. Moreover, from (7.10) it follows that $\dot{e} \in \mathcal{L}^\infty$. This permits application of Barbalat's Lemma on (7.15) and yields $\lim_{t\to\infty} e(t) = 0$, which completes the proof. □

In most cases, it is desirable to choose an LTI reference system as it defines the same reference system for all subsystems or equivalently over the entire state-input space. This

choice causes the nonlinear system (7.2) to behave like a linear system and naturally satisfies the assumption of a common quadratic Lyapunov function in Theorem 7.4. Depending on the PWA system, however, the matching conditions (7.8) might not allow for an LTI reference system. In such cases, a PWA reference system must be employed, for which a common quadratic Lyapunov function might not exist. In this case, the following theorem considers a PWA reference system and shows that under a certain dwell-time assumption and under sufficiently rich reference signals, all tracking errors and parameter errors converge to zero.

Theorem 7.5 (Direct MRAC and parameter convergence for PWA systems without CQLF) Consider the reference system (7.3) without common quadratic Lyapunov function and let the PWA system (7.2) with known regions Ω_i be controlled by the state feedback (7.6) with gains updated according to (7.11). Let the reference signals in r be sufficiently rich of order $n+1$ with distinct frequencies. Furthermore, let the resulting switching signal be sufficiently slow with dwell time T_{dwell} and cause repeated activation of all subsystems. If the input matrices B_i have full column rank, if the system matrices $A_{\text{m}i}$ are invertible, and if the the pairs $(A_{\text{m}i}, B_{\text{m}i})$ are controllable, then all errors e, $\tilde{K}_{xi}$, $\tilde{K}_{ri}$ and $\tilde{k}_{fi}$ asymptotically converge to zero for $t \to \infty$.

Proof: The proof of Theorem 7.5 begins with analyzing the convergence of the active subsystem parameters. Consider a time interval $t \in [t_{k-1}, t_k)$ and let i be such that $\chi_i(t) = 1$. Thus the error equation (7.10) reads as

$$\dot{e} = A_{\text{m}i} e + B_i \begin{bmatrix} \tilde{K}_{xi} & \tilde{K}_{ri} & \tilde{k}_{fi} \end{bmatrix} \begin{bmatrix} x \\ r \\ 1 \end{bmatrix}. \tag{7.16}$$

We make use of the Kronecker product $\otimes$ and define

$$\Psi_r := \begin{bmatrix} x \\ r \\ 1 \end{bmatrix} \otimes I_n, \quad \tilde{\vartheta}_i^B := \operatorname{vec}\left(B_i \begin{bmatrix} \tilde{K}_{xi} & \tilde{K}_{ri} & \tilde{k}_{fi} \end{bmatrix} \right), \tag{7.17}$$

where the operator $\operatorname{vec}(\cdot)$ concatenates the columns of a matrix $(\cdot)$ in a single column vector. With (7.17), rewrite (7.16) as

$$\dot{e} = A_{\text{m}i} e + \Psi_r^\top \tilde{\vartheta}_i^B. \tag{7.18}$$

In the next step, the time derivative

$$\dot{\tilde{\vartheta}}_i^B = \operatorname{vec}\left(B_i \begin{bmatrix} \dot{\tilde{K}}_{xi} & \dot{\tilde{K}}_{ri} & \dot{\tilde{k}}_{fi} \end{bmatrix} \right) \tag{7.19}$$

is formulated in terms of the update laws (7.11) (with neglected scaling constants for simplicity). After inserting (7.11) and performing various steps of algebraic reformulation,

the derivative (7.19) relates to the tracking error e and Ψ_r in the following way:

$$\dot{\tilde{\vartheta}}_i^B = -\Psi_r \underbrace{B_i M_{si}^{-1} B_i^\top P_i}_{=:P_{i2}} e = -\Psi_r P_{i2} e. \tag{7.20}$$

Equations (7.18) and (7.20) can be combined in the following dynamic system, which is a standard form frequently appearing in adaptive systems:

$$\begin{bmatrix} \dot{e} \\ \dot{\tilde{\vartheta}}_i^B \end{bmatrix} = \begin{bmatrix} A_\mathrm{m} & \Psi_r^\top \\ -\Psi_r P_{i2} & 0 \end{bmatrix} \begin{bmatrix} e \\ \tilde{\vartheta}_i^B \end{bmatrix}. \tag{7.21}$$

With Lemma A.6, exponential stability of the equilibrium $e = 0$ and $\tilde{\vartheta}_i^B = 0$ can thus be concluded if $[x^\top, r^\top, 1]^\top$ is PE. Furthermore, with Lemma 4.4, which was introduced earlier for the convergence proof in parameter identifiers for PWA systems, we determine that $[x_\mathrm{m}^\top,\ r^\top,\ 1]^\top$ is PE under the given assumptions of controllable pairs $(A_{\mathrm{m}i}, B_{\mathrm{m}i})$, invertible $A_{\mathrm{m}i}$ and sufficiently rich reference signals with distinct frequencies. Finally, according to [37], PE of the reference system ensures PE of the controlled system. An alternative reasoning is that for $e \to 0$, we have $x = x_\mathrm{m}$ and thus if the reference system is PE also the controlled system is PE. Hence, on the interval $t \in [t_{k-1}, t_k)$, the errors e and $\tilde{\vartheta}_i^B$ converge towards zero exponentially.

Note that due to its assumed full column rank, the null space of B_i is empty. Therefore, the exponential convergence of $\tilde{\vartheta}_i^B \to 0$ also implies exponential convergence of $\tilde{\vartheta}_i$, where $\tilde{\vartheta}_i := \mathrm{vec}\left([\tilde{K}_{xi}, \tilde{K}_{ri}, \tilde{k}_{fi}]\right)$. Thus, for $z_i = [e^\top, \tilde{\vartheta}_i^\top]^\top$, there exist constants $a_i > 0$ and $\lambda_i > 0$ such that

$$\|z_i(t)\| \leq a_i e^{-\lambda_i (t - t_{k-1})} \|z_i(t_{k-1})\|. \tag{7.22}$$

It is well known that exponential stability of all subsystems alone does not imply stability of the switched system. In order to conclude asymptotic stability, or convergence in the present case, it has to be shown that the Lyapunov function values at switching times form a decreasing sequence [40]. Therefore, in the next step, the worst-case (i.e. minimal) decrease of the Lyapunov function (7.12) over the interval $t \in [t_{k-1}, t_k)$, for which $\chi_i(t) = 1$, is compared with the worst-case (i.e. maximal) increment of the Lyapunov function associated with the switch at time instance t_k. Note that the control gains of inactive subsystems are constant over the considered interval. Therefore, the term

$$C_{k-1} = \sum_{j=1, j\neq i}^{s} \mathrm{tr}(\tilde{K}_{xj}^\top M_{sj} \tilde{K}_{xj}) + \mathrm{tr}(\tilde{K}_{rj}^\top M_{sj} \tilde{K}_{rj}) + \tilde{k}_{fj}^\top M_{sj} \tilde{k}_{fj}$$

is also constant over $[t_{k-1}, t_k)$. Since V is quadratic in z_i, (7.22) can bound the decrease of V for the adapted gains and tracking error from above by

$$\begin{aligned} V(t_k^-) - C_{k-1} &\leq \lambda_{\max}(P_i) \|z(t_k^-)\|^2 \leq \alpha_i (a_i \mathrm{e}^{-\lambda_i (t_k^- - t_{k-1})} \|z_i(t_{k-1})\|)^2 \\ &\leq \frac{\alpha_i}{\beta_i} a_i^2 e^{-2\lambda_i (t_k^- - t_{k-1})} \left(V(t_{k-1}) - C_{k-1}\right), \end{aligned} \tag{7.23}$$

with $\alpha_i = \lambda_{\max}(P_i)$ and $\beta_i = \lambda_{\min}(P_i)$, and where the last step follows from $V(t_{k-1}) - C_{k-1} \geq \beta_i \|z_i(t_{k-1})\|^2$. In order for the Lyapunov function to form a decreasing sequence at switching times, the decrease in (7.23) must be greater than the increment induced by the switching at time t_k. Note that the switch does not affect the terms in C_{k-1}. Hence, the bound given in (7.5) only applies to $V(t_k) - C_{k-1}$ and $V(t_k^-) - C_{k-1}$, which yields the worst-case increase due to switching at t_k:

$$V(t_k) - C_{k-1} \leq (1 + \mu\Delta_{A_\mathrm{m}})(V(t_k^-) - C_{k-1}). \tag{7.24}$$

Combining the decrease in (7.23) with the increase in (7.24) characterizes the worst-case evolution between consecutive switches as

$$V(t_k) - C_{k-1} \leq \underbrace{\frac{\alpha_i}{\beta_i} a_i^2 (1 + \mu\Delta_{A_\mathrm{m}}) \mathrm{e}^{-2\lambda_i (t_k - t_{k-1})}}_{=:\rho} (V(t_{k-1}) - C_{k-1}). \tag{7.25}$$

For a decreasing sequence, it must hold that $\rho \leq 1$ or equivalently

$$t_k - t_{k-1} \geq \frac{1}{2\lambda_i} \ln\left(\frac{\alpha_i}{\beta_i} a_i^2 (1 + \mu\Delta_{A_\mathrm{m}})\right) =: T_i, \tag{7.26}$$

which defines a dwell time T_i for the i-th subsystem. We conclude that stability and convergence of the direct adaptive control algorithm is given if the switching of the controlled system obeys a dwell time $T_{\mathrm{dwell}} > \max\{T_\mathrm{m}, T_1, \ldots, T_s\}$, i.e., $t_k - t_{k-1} \geq T_{\mathrm{dwell}}, \forall k > 1$. □

Remark 7.6 Note that the assumption of known S_i constitutes – besides the known regions of the PWA system – the biggest limitation of the proposed algorithm. In the multi-input case the retrieval of suitable S_i may become non-trivial. For the single-input case, however, it reduces to $S_i = \mathrm{sign}(K_{ri}^*)$. Hence, S_i corresponds to the usual assumption of known input effectiveness. Relaxing the required knowledge about S_i is an important topic and has received some attention in the multivariable adaptive control community [241]. In [114], it is shown that for positive definite and symmetric nominal gain matrices K_{ri}^*, the design matrix can be chosen as identity matrix: $S_i = I$. Another approach, discussed in [240, Sec. 9.1.2], parametrizes K_{ri}^* in terms of an LDU decomposition with lower and upper triangular matrices L and U, and a diagonal matrix D. For the resulting stable convergent adaptive controller, only L and the sign of the diagonal entries in D need to be known.

This section introduced MRAC with direct gain adaptation according to (7.11). Compared to its piecewise linear counterpart in [218], the consideration of affine terms enlarges the class of applicable systems. Also it is not restricted to single input system in canonical form as [65, 70]. It was determined in Theorem 7.4 that the state tracking task can be achieved without imposing additional requirements on the excitation of the reference system. Theorem 7.5 on the other side suggests excitation with sufficiently rich reference signals in case convergence to the nominal control gains is required. This is for instance the case if a non-periodic reference signal needs to be tracked precisely or the nominal gains are to be monitored. For monitoring purposes it might, however, be more interesting

to analyze the system parameters A_i, B_i and f_i instead of the control gains. This is the motivation for the indirect MRAC algorithm derived in the next section, which has the advantage that estimates of the system parameters are generated in parallel to the control task.

7.3 Indirect Model Reference Adaptive Control

In the direct MRAC approach, the control gains are directly tuned based on the tracking error. In the indirect case, which is discussed next, the adaptation of control gains is based on time-varying estimates $\hat{A}_i(t)$, $\hat{B}_i(t)$ and $\hat{f}_i(t)$ of the system parameters A_i, B_i and f_i are introduced. Under the certainty equivalence principle, the estimates are handled as if they are the true parameters at all times [241]. Based on the estimates, the control gains K_{xi}, K_{ri} and k_{fi} are derived from the matching conditions (7.8).

Let the estimates $\hat{A}_i$, $\hat{B}_i$ and $\hat{f}_i$ be determined by the parameter identifiers presented in Chapter 4. Recall that parameter identifiers predict the state x in terms of the parameter estimates $(\hat{A}_i,\ \hat{B}_i,\ \hat{f}_i)$ of the currently active subsystem ($\chi_i = 1$). Transferring the prediction of $\hat{x} \in \mathbb{R}^n$ from the σ-representation in (4.20) to the representation with indicator functions χ_i yields

$$\dot{\hat{x}} = A_{\mathrm{m}}\hat{x} + \sum_{i=1}^{s} \left((\hat{A}_i - A_{\mathrm{m}i})x + \hat{B}_i u + \hat{f}_i \right) \chi_i, \tag{7.27}$$

where $A_{\mathrm{m}} = \sum_{i=1}^{s} A_{\mathrm{m}i}\chi_i$ is the switched system matrix of the stable reference system (7.3). Let $\tilde{x} = \hat{x} - x$ be the prediction error between the predictor (7.27) and the controlled system (7.2). This prediction error contains information about the mismatch between parameter estimates and true parameters, and is hence an essential element in the update law. The parameter update laws (4.23) introduced in Chapter 4 become

$$\dot{\hat{A}}_i = -\chi_i P_i \tilde{x} x^\top, \qquad \dot{\hat{B}}_i = -\chi_i P_i \tilde{x} u^\top, \qquad \dot{\hat{f}}_i = -\chi_i P_i \tilde{x},$$

where the indicator functions χ_i ensure that only the currently active subsystem parameters are updated. Let $\tilde{A}_i = \hat{A}_i - A_i$, $\tilde{B}_i = \hat{B}_i - B_i$ and $\tilde{f}_i = \hat{f}_i - f_i$ be the parameter errors. Then, the derivative of the prediction error in terms of these parameter errors is

$$\dot{\tilde{x}} = A_{\mathrm{m}}\tilde{x} + \sum_{i=1}^{s} \left(\tilde{A}_i x + \tilde{B}_i u + \tilde{f}_i \right) \chi_i. \tag{7.28}$$

Following the certainty equivalence principle, one can rearrange the matching conditions (7.8) and obtain the following algebraic equations for the controller gains:

$$K_{xi} = \hat{B}_i^{+}(A_{\mathrm{m}i} - \hat{A}_i), \qquad K_{ri} = \hat{B}_i^{+} B_{\mathrm{m}i}, \qquad k_{fi} = \hat{B}_i^{+}(f_{\mathrm{m}i} - \hat{f}_i),$$

where $\hat{B}_i^{+} = (\hat{B}_i^\top \hat{B}_i)^{-1}\hat{B}_i$ represents the Moore-Penrose pseudoinverse of $\hat{B}_i$. This pseudo inverse constitutes the main obstacle in this approach as it introduces singularities into the system. In case of singularities, the control gains take unbounded values which violates

the stability requirement. Hence, additional measures need to be taken to ensure stability. One approach is the use of projection operators to avoid singularities [143]. In order to use projection operators, however, additional knowledge about the system (more precisely K_{ri}^*) as compared to the direct adaptive control algorithm in Section 7.2 would be needed.

Instead of introducing projection operators, we follow ideas previously discussed in [76, 175] and define dynamic update laws for the control parameters (K_{xi}, K_{ri}, k_{fi}) which are based on the current parameter estimates $(\hat{A}_i, \hat{B}_i, \hat{f}_i)$. This preserves the indirect nature of the adaptation algorithm (hence provides estimates of the system parameters) and at the same time relies on the same previous knowledge about the system as the direct approach. Our contribution in this context is the extension of [175, Sec. 3.3] and [76] to switched affine systems. We also present a thorough proof of parameter convergence, which is not given in [76, 175].

We begin by defining the following *closed-loop estimation errors*, which can be interpreted as errors in the matching conditions (7.8):

$$\varepsilon_{Ai} = \hat{A}_i + \hat{B}K_{xi} - A_{\mathrm{m}i}, \qquad \varepsilon_{Bi} = \hat{B}_i K_{ri} - B_{\mathrm{m}i}, \qquad \varepsilon_{fi} = \hat{f}_i + \hat{B}k_{fi} - f_{\mathrm{m}i}. \tag{7.29}$$

The closed-loop estimation errors (7.29) characterize the mismatch between the desired system dynamics and the estimated closed-loop system dynamics, which are obtained for the estimated system parameters and control gains.

Lyapunov-based stability analysis (which will be presented shortly) suggests the next steps. Let the closed-loop estimation errors drive the adaptation of the control gains through the *indirect update laws*

$$\dot{K}_{xi} = -S_i^\top B_{\mathrm{m}i}^\top \varepsilon_{Ai}, \qquad \dot{K}_{ri} = -S_i^\top B_{\mathrm{m}i}^\top \varepsilon_{Bi}, \qquad \dot{k}_{fi} = -S_i^\top B_{\mathrm{m}i}^\top \varepsilon_{fi}, \tag{7.30}$$

where S_i is the same – assumed to be known – matrix as in the direct adaptive control algorithm, for which $K_{ri}^* S_i$ is symmetric and positive definite. In order to maintain convergence for the new control gain adaptation (7.30), also the adaptation of system parameters needs to include the closed-loop estimation errors. Stability analysis suggests

$$\begin{aligned} \dot{\hat{A}}_i &= -\chi_i P_i \tilde{x} x^\top - \varepsilon_{Ai}, \\ \dot{\hat{B}}_i &= -\chi_i P_i \tilde{x} u^\top - \varepsilon_{Ai} K_{xi}^\top - \varepsilon_{Bi} K_{ri}^\top - \varepsilon_{fi} k_{fi}^\top, \\ \dot{\hat{f}}_i &= -\chi_i P_i \tilde{x} - \varepsilon_{fi}. \end{aligned} \tag{7.31}$$

The following theorem characterizes indirect MRAC for PWA systems for reference systems whose subsystems have a common quadratic Lyapunov function.

Theorem 7.7 (Indirect MRAC for PWA systems with CQLF) Consider a reference system (7.3) for which a common quadratic Lyapunov function with $P = P_i$, $\forall i$ is known. Let the PWA system (7.2) with known regions Ω_i be controlled by the state feedback (7.6) with gains updated according to (7.30), which is based on (7.27), (7.29) and (7.31). Then, the state of the PWA system asymptotically tracks the state of the reference system.

Proof: The proof is based on the candidate Lyapunov function

$$V = \frac{1}{2}\tilde{x}^\top P\tilde{x} + \frac{1}{2}\sum_{i=1}^{s}\Big(\operatorname{tr}\left(\tilde{A}_i^\top\tilde{A}_i\right) + \operatorname{tr}\left(\tilde{B}_i^\top\tilde{B}_i\right) + \tilde{f}_i^\top\tilde{f}_i \\ + \operatorname{tr}\left(\tilde{K}_{xi}^\top M_{si}\tilde{K}_{xi}\right) + \operatorname{tr}\left(\tilde{K}_{ri}^\top M_{si}\tilde{K}_{ri}\right) + \tilde{k}_{fi}^\top M_{si}\tilde{k}_{fi}\Big),$$

whose time derivative along $\dot{\tilde{x}}$ in (7.28) is

$$\dot{V} = \tilde{x}^\top\left(\frac{1}{2}\sum_{i=1}^{s}(A_{\mathrm{m}i}^\top P + PA_{\mathrm{m}i})\chi_i\right)\tilde{x} + \sum_{i=1}^{s}\left(\tilde{x}^\top P\left(\tilde{A}_i x + \tilde{B}_i u + \tilde{f}_i\right)\chi_i\right. \\ \left. + \operatorname{tr}\left(\tilde{A}_i^\top\dot{\tilde{A}}_i\right) + \operatorname{tr}\left(\tilde{B}_i^\top\dot{\tilde{B}}_i\right) + \tilde{f}_i^\top\dot{\tilde{f}}_i + \operatorname{tr}\left(\tilde{K}_{xi}^\top M_{si}\dot{\tilde{K}}_{xi}\right) + \operatorname{tr}\left(\tilde{K}_{ri}^\top M_{si}\dot{\tilde{K}}_{ri}\right) + \tilde{k}_{fi}^\top M_{si}\dot{\tilde{k}}_{fi}\right).$$

At this point, with equalities of the form $\tilde{x}^\top P\tilde{A}_i x = \operatorname{tr}(\tilde{A}_i^\top P\tilde{x}x^\top)$, the first elements in the update laws (7.31) for $\hat{A}$, $\hat{B}$ and $\hat{f}$ prove suitable to cancel out $\tilde{x}^\top P(\tilde{A}_i x + \tilde{B}_i u + \tilde{f}_i)\chi_i$. The elements containing ε_{Ai}, ε_{Ai}, ε_{Ai} on the other side will turn out to cancel those terms associated with the control gains $\tilde{K}_{xi}$, $\tilde{K}_{ri}$, $\tilde{k}_{fi}$. To see this, insert the update laws (7.31) with $P_i = P$, which yields

$$\dot{V} = -\tilde{x}^\top\left(\frac{1}{2}\sum_{i=1}^{s}Q_{\mathrm{m}i}\chi_i\right)\tilde{x} + \sum_{i=1}^{s}\Big(-\operatorname{tr}\left(\tilde{A}_i^\top\varepsilon_{Ai}\right) - \operatorname{tr}\left(\tilde{B}_i^\top\left(\varepsilon_{Ai}K_{xi}^\top + \varepsilon_{Bi}K_{ri}^\top + \varepsilon_{fi}k_{fi}^\top\right)\right) \\ - \tilde{f}_i^\top\varepsilon_{fi} + \operatorname{tr}\left(\tilde{K}_{xi}^\top M_{si}\dot{\tilde{K}}_{xi}\right) + \operatorname{tr}\left(\tilde{K}_{ri}^\top M_{si}\dot{\tilde{K}}_{ri}\right) + \tilde{k}_{fi}^\top M_{si}\dot{\tilde{k}}_{fi}\Big).$$

With some algebraic reformulations, this expression suggest the proposed adaptation of control gains in (7.30). Inserting the update laws (7.30) for $\dot{\tilde{K}}_{xi}$, $\dot{\tilde{K}}_{ri}$ and $\dot{\tilde{k}}_{fi}$ gives

$$\dot{V} = -\tilde{x}^\top\left(\frac{1}{2}\sum_{i=1}^{s}Q_{\mathrm{m}i}\chi_i\right)\tilde{x} + \sum_{i=1}^{s}\Big(-\operatorname{tr}\left(\tilde{A}_i^\top\varepsilon_{Ai}\right) - \operatorname{tr}\left(\tilde{B}_i^\top\left(\varepsilon_{Ai}K_{xi}^\top + \varepsilon_{Bi}K_{ri}^\top + \varepsilon_{fi}k_{fi}^\top\right)\right) \\ - \tilde{f}_i^\top\varepsilon_{fi} - \operatorname{tr}\left(\tilde{K}_{xi}^\top M_{si}S_i^\top B_{\mathrm{m}i}^\top\varepsilon_{Ai}\right) - \operatorname{tr}\left(\tilde{K}_{ri}^\top M_{si}S_i^\top B_{\mathrm{m}i}^\top\varepsilon_{Bi}\right) - \tilde{k}_{fi}^\top M_{si}S_i^\top B_{\mathrm{m}i}^\top\varepsilon_{fi}\Big),$$

which by exploiting

$$M_{si}S_i^\top B_{\mathrm{m}i}^\top = M_{si}S_i^\top(B_iK_{ri}^*)^\top = M_{si}S_i^\top{K_{ri}^*}^\top B_i^\top = M_{si}M_{si}^{-1}B_i^\top = B_i^\top,$$

and grouping traces of matrices with equal dimension, results in

$$\dot{V} = -\tilde{x}^\top\left(\frac{1}{2}\sum_{i=1}^{s}Q_{\mathrm{m}i}\chi_i\right)\tilde{x} + \sum_{i=1}^{s}\Big(-\operatorname{tr}\left(\tilde{A}_i^\top\varepsilon_{Ai} + \tilde{K}_{xi}^\top B_i^\top\varepsilon_{Ai}\right) - \tilde{f}_i^\top\varepsilon_{fi} - \tilde{k}_{fi}^\top B_i^\top\varepsilon_{fi} \\ - \operatorname{tr}\left(\tilde{B}_i^\top(\varepsilon_{Ai}K_{xi}^\top + \varepsilon_{Bi}K_{ri}^\top + \varepsilon_{fi}k_{fi}^\top) + \tilde{K}_{ri}^\top B_i^\top\varepsilon_{Bi}\right)\Big). \quad (7.32)$$

Inside the trace operators, one can justify the transformations

$$\begin{aligned}\left(\tilde{A}_i^\top + \tilde{K}_{xi}^\top B_i^\top\right)\varepsilon_{Ai} &= \left(\hat{A}_i^\top - A_i^\top + K_{xi}^\top B_i^\top - {K_{xi}^*}^\top B_i^\top\right)\varepsilon_{Ai} = \left(\hat{A}_i^\top - A_{\mathrm{m}i}^\top + K_{xi}^\top B_i^\top\right)\varepsilon_{Ai}\\ &= \left(\hat{A}_i^\top - A_{\mathrm{m}i}^\top + K_{xi}^\top \hat{B}_i^\top - K_{xi}^\top \tilde{B}_i^\top\right)\varepsilon_{Ai} = \left(\varepsilon_{Ai} - K_{xi}^\top \tilde{B}_i^\top\right)\varepsilon_{Ai},\end{aligned}$$

and

$$\begin{aligned}&\tilde{B}_i^\top(\varepsilon_{Ai}K_{xi}^\top + \varepsilon_{Bi}K_{ri}^\top + \varepsilon_{fi}k_{fi}^\top) + \tilde{K}_{ri}^\top B_i^\top \varepsilon_{Bi}\\ &\quad= \tilde{B}_i^\top\varepsilon_{Ai}K_{xi}^\top + \tilde{B}_i^\top\varepsilon_{fi}k_{fi}^\top + \hat{B}_i^\top\varepsilon_{Bi}K_{ri}^\top - B_i^\top\varepsilon_{Bi}K_{ri}^\top + K_{ri}^\top B_i^\top\varepsilon_{Bi} - {K_{ri}^*}^\top B_i^\top\varepsilon_{Bi}\\ &\quad= \tilde{B}_i^\top\varepsilon_{Ai}K_{xi}^\top + \tilde{B}_i^\top\varepsilon_{fi}k_{fi}^\top + \hat{B}_i^\top\varepsilon_{Bi}K_{ri}^\top - {K_{ri}^*}^\top B_i^\top\varepsilon_{Bi}\\ &\quad= \tilde{B}_i^\top\varepsilon_{Ai}K_{xi}^\top + \tilde{B}_i^\top\varepsilon_{fi}k_{fi}^\top + \hat{B}_i^\top\varepsilon_{Bi}K_{ri}^\top - B_{\mathrm{m}i}^\top\varepsilon_{Bi}\\ &\quad= \tilde{B}_i^\top\varepsilon_{Ai}K_{xi}^\top + \tilde{B}_i^\top\varepsilon_{fi}k_{fi}^\top + \varepsilon_{Bi}^\top\varepsilon_{Bi}.\end{aligned}$$

Hence, together with

$$\begin{aligned}-\left(\tilde{f}_i^\top + \tilde{k}_{fi}^\top B_i^\top\right)\varepsilon_{fi} &= -\left(\hat{f}_i^\top - f_i^\top + k_{fi}^\top B_i^\top - {k_{fi}^*}^\top B_i^\top\right)\varepsilon_{fi} = -\left(\hat{f}_i^\top - f_{\mathrm{m}i}^\top + k_{fi}^\top B_i^\top\right)\varepsilon_{fi}\\ &= -\left(\hat{f}_i^\top - f_{\mathrm{m}i}^\top + k_{fi}^\top \hat{B}_i^\top - k_{fi}^\top \tilde{B}_i^\top\right)\varepsilon_{fi} = -\varepsilon_{fi}^\top\varepsilon_{fi} + k_{fi}^\top \tilde{B}_i^\top\varepsilon_{fi},\end{aligned}$$

the derivative of the Lyapuonv function in (7.32) can be rewritten as

$$\begin{aligned}\dot{V} &= -\tilde{x}^\top\left(\frac{1}{2}\sum_{i=1}^{s} Q_{\mathrm{m}i}\chi_i\right)\tilde{x} + \sum_{i=1}^{s}\Big(-\operatorname{tr}\left(\varepsilon_{Ai}^\top\varepsilon_{Ai}\right) - \operatorname{tr}\left(\varepsilon_{Bi}^\top\varepsilon_{Bi}\right) - \varepsilon_{fi}^\top\varepsilon_{fi}\\ &\qquad + \operatorname{tr}\left(K_{xi}^\top\tilde{B}_i^\top\varepsilon_{Ai}\right) - \operatorname{tr}\left(\tilde{B}_i^\top\varepsilon_{Ai}K_{xi}^\top + \tilde{B}_i^\top\varepsilon_{fi}k_{fi}^\top\right) + k_{fi}^\top\tilde{B}_i^\top\varepsilon_{fi}\Big)\\ &= -\tilde{x}^\top\left(\frac{1}{2}\sum_{i=1}^{s} Q_{\mathrm{m}i}\chi_i\right)\tilde{x} - \sum_{i=1}^{s}\Big(\operatorname{tr}\left(\varepsilon_{Ai}^\top\varepsilon_{Ai}\right) + \operatorname{tr}\left(\varepsilon_{Bi}^\top\varepsilon_{Bi}\right) + \varepsilon_{fi}^\top\varepsilon_{fi}\Big),\end{aligned}$$

which is negative semidefinite.

As in the direct case, special attention must be given to potential sliding modes. In order to analyze all convex combinations of vector fields contributing to the sliding mode, replace the binary indicator functions χ_i by $\bar{\chi}_i$ in the above derivation. Since all steps are also valid for $\bar{\chi}_i$, the property $\dot{V} < 0, \forall\tilde{x} \neq 0$ is verified for all possible sliding mode solutions.

It follows that V is a Lyapunov function decreasing also along possible sliding mode solutions and therefore $\tilde{x}$, $\hat{A}_i$, $\hat{B}_i$, $\hat{f}_i$, K_{xi}, K_{ri} and k_{fi} are all bounded (equivalently $\in \mathcal{L}^\infty$). At the same time, according to (7.29), this implies boundedness of ε_{Ai}, ε_{Bi} and ε_{fi}. Furthermore, the Lyapunov function and its derivative imply that ε_{Ai}, ε_{Bi}, $\varepsilon_{fi} \in \mathcal{L}^2$. It remains to be shown that also x, $\hat{x}$ and u are bounded, which cannot be concluded directly from boundedness of $\tilde{x}$. Instead, rewrite (7.27) with feedback (7.6) in terms of ε_{Ai}, ε_{Bi} and

ε_{fi} as

$$\begin{aligned}\dot{\hat{x}} &= A_\mathrm{m}\hat{x} + (\hat{A} - A_\mathrm{m})x + \hat{B}u + \hat{f} = A_\mathrm{m}\hat{x} + (\hat{A} - A_\mathrm{m})x + \hat{B}(K_x x + K_r r + k_f) + \hat{f} \\ &= A_\mathrm{m}\hat{x} + \varepsilon_A x + \hat{B}(K_r r + k_f) + \hat{f} = (A_\mathrm{m} + \varepsilon_A)\hat{x} - \varepsilon_A\tilde{x} + (B_\mathrm{m} - \varepsilon_B)r + f_\mathrm{m} - \varepsilon_f,\end{aligned} \tag{7.33}$$

where $\varepsilon_A := \sum_{i=1}^{s} \varepsilon_{Ai}\chi_i$, $\varepsilon_B := \sum_{i=1}^{s} \varepsilon_{Bi}\chi_i$ and $\varepsilon_f := \sum_{i=1}^{s} \varepsilon_{fi}\chi_i$. By design, reference signals are assumed to be bounded: $r \in \mathcal{L}^\infty$. With stable A_m, boundedness of $(B_\mathrm{m} - \varepsilon_B)r + f_\mathrm{m} - \varepsilon_f$, and $\varepsilon_A \in \mathcal{L}^2$, it can be concluded from (7.33) that $\hat{x} \in \mathcal{L}^\infty$. Since $x = \hat{x} - \tilde{x}$, and since both $\hat{x}$ and $\tilde{x}$ are bounded, also the system state is bounded ($x \in \mathcal{L}^\infty$). This implies boundedness of u due to (7.6). Finally, bounded x and u imply boundedness of $\dot{\tilde{x}}$, $\dot{\varepsilon}_A$, $\dot{\varepsilon}_B$ and $\dot{\varepsilon}_f$ due to $\dot{\tilde{x}} = A_\mathrm{m}\tilde{x} + \tilde{A}x + \tilde{B}u + \tilde{f}$ and (7.29). Thus, it follows that $\ddot{V} \in \mathcal{L}^\infty$ and $\dot{V} \in \mathcal{L}^2 \cap \mathcal{L}^\infty$ which makes Barbalat's Lemma applicable and therefore $\lim_{t\to\infty} \dot{V}(t) = 0$ [175, Corollary 2.9]. Hence, $\tilde{x}$, ε_A, ε_B, $\varepsilon_f \to 0$ as $t \to \infty$.

Inspecting the dynamics of $\hat{x}$ in (7.33), we conclude that

$$\lim_{t\to\infty} \dot{\hat{x}}(t) = A_\mathrm{m}\hat{x} + B_\mathrm{m}r + f_\mathrm{m}$$

and therefore the predicted state $\hat{x}$ approaches x_m asymptotically. As $\tilde{x} \to 0$ it follows that also the state x of the controlled system asymptotically tracks the state x_m of the reference system, i.e., $\lim_{t\to\infty} e(t) = 0$. □

Remark 7.8 There is an interesting difference between the state prediction carried out for the parameter identifiers in Chapter 4 and the prediction for indirect MRAC. While the predicted state was reset to the measured state at every switch in the former case, no state reset is allowed in the case of indirect MRAC. This is due to the fact that boundedness of the system state x needs to be shown in the adaptive control setting. For identification on the other side it was assumed that a stabilizing control input is applied. If the same state reset as in the identification case (shown in Fig. 4.1) was applied for indirect MRAC one could not claim boundedness of $\hat{x}$ (and in turn x by (7.33)). Hence, stability could not be guaranteed.

The advantage of indirect approaches lies in the ability to estimate the actual system parameters. As can be seen from the proof of Theorem 7.7, however, convergence of the parameter estimates is not necessary for the tracking task. Before we show which additional assumptions need to be imposed on the reference signals in order to achieve parameter converge, we need the following lemma on the PE condition of some signal vector in the reference system.

Lemma 7.9 (PE of internal signals – indirect MRAC) Let the reference system (7.3) be realized as the unknown system $\dot{x}_\mathrm{m} = A_i x_\mathrm{m} + B_i u_\mathrm{m} + f_i$ with nominal controller $u_\mathrm{m} = K_{xi}^* x_\mathrm{m} + K_{ri}^* r + k_{fi}^*$. Furthermore, let the reference signals in r be sufficiently rich of order $n + 1$ with distinct frequencies. If the system matrices $A_{\mathrm{m}i}$ are invertible, and if the pairs $(A_{\mathrm{m}i}, B_{\mathrm{m}i})$ are controllable, then the signal vector $z_\mathrm{m} = [x_\mathrm{m}^\top, u_\mathrm{m}^\top, 1]^\top$ is PE.

The technical proof of Lemma 7.9 is given in Appendix D.3 for improved readability. Based on Lemma 7.9, the following theorem specifies that all estimated parameters converge to the nominal values under the assumption of sufficiently rich reference signals.

Theorem 7.10 (Indirect MRAC and parameter convergence for PWA systems with CQLF) Consider a reference system (7.3) for which a common quadratic Lyapunov function with $P = P_i,\ \forall i$ is known. Let the PWA system (7.2) with known regions Ω_i be controlled by the state feedback (7.6) with gains updated according to (7.30), which is based on (7.27), (7.29) and (7.31). Let the reference signals in r be sufficiently rich of order $n+1$ with distinct frequencies and such that all subsystems are repeatedly activated. If the input matrices B_i have full column rank, if the system matrices $A_{\mathrm{m}i}$ are invertible, and if the pairs $(A_{\mathrm{m}i}, B_{\mathrm{m}i})$ are controllable, then the state of the PWA system asymptotically tracks the state of the reference system and the estimated parameters $\hat{A}_i$, $\hat{B}_i$ and $\hat{f}_i$ as well as the estimated gains K_{xi}, K_{ri} and k_{fi} converge to their nominal values as $t \to \infty$.

Proof: The proof of Theorem 7.7 showed that the adaptation is stable and that state tracking is achieved. For the proof of Theorem 7.10, it is thus left to show that the parameter estimates of the active subsystem converge asymptotically to the true parameters. During this proof, we thus neglect the index i with the understanding that all steps shown here apply for the active subsystem. Hence, if all subsystems are activated in a persistent manner, all parameter estimates converge.

The proof consists of three steps. First, the convergence of estimated subsystem parameters is tied to a PE condition of internal signals. Then, Lemma 7.9 is applied to verify this PE requirement and thus show convergence of the estimated subsystem parameters. In the final step, it is shown that converging parameter estimates yield converging control gains.

Using a similar notations as in (7.17), begin by introducing

$$\Psi_u := \begin{bmatrix} x \\ u \\ 1 \end{bmatrix} \otimes I_n, \qquad \tilde{\theta} := \mathrm{vec}\left(\begin{bmatrix} \tilde{A} & \tilde{B} & \tilde{f} \end{bmatrix}\right). \tag{7.34}$$

Then, from (7.28) and (7.31), likewise (7.21), the closed-loop system for both the prediction error $\tilde{x}$ and parameter errors $\tilde{\theta}$ can be written as

$$\begin{bmatrix} \dot{\tilde{x}} \\ \dot{\tilde{\theta}} \end{bmatrix} = \begin{bmatrix} A_{\mathrm{m}} & \Psi_u^\top(t) \\ -\Psi_u(t) P & 0 \end{bmatrix} \begin{bmatrix} \tilde{x} \\ \tilde{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ \varepsilon \end{bmatrix}, \tag{7.35}$$

where $\varepsilon = -\mathrm{vec}([\varepsilon_A,\ \varepsilon_A K_x^\top + \varepsilon_B K_r^\top + \varepsilon_f k_f^\top,\ \varepsilon_f])$. Note that from the proof of Theorem 7.7 it can be concluded that $\varepsilon \in \mathcal{L}^2$. Hence, convergence is derived from the analysis of the first summand in (7.35).

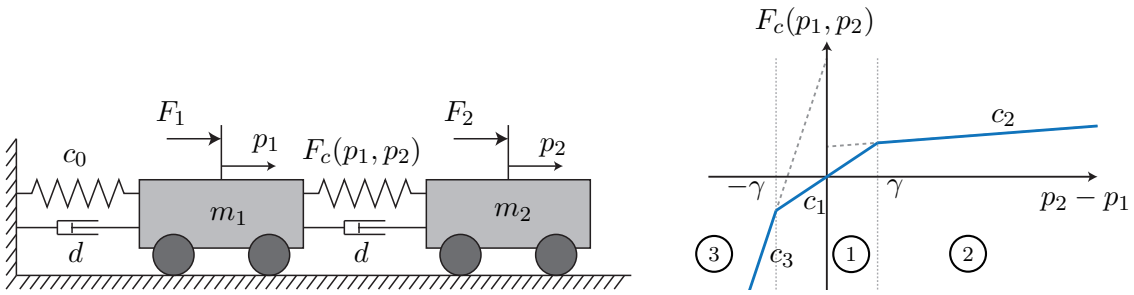
For (7.35) with $\varepsilon = 0$, Lemma A.6 ensures exponential stability of the equilibrium $\tilde{x} = 0$ and $\tilde{\theta} = 0$ if $z_u = [x^\top, u^\top, 1]^\top$ is PE. As Theorem 7.7 showed $e \to 0$, it follows that z_u is PE if $z_{\mathrm{m}} = [x_{\mathrm{m}}^\top, u_{\mathrm{m}}^\top, 1]^\top$ is PE, where u_{m} can be thought of as an internal signal in the reference system if it was realized in terms of the matching conditions, i.e. $u_{\mathrm{m}} = K_x^* x_{\mathrm{m}} + K_r^* r + k_f^*$. Under the given assumptions, Lemma 7.9 guarantees PE of z_{m}. Hence, the PE condition

on z_u is also satisfied, which – according to Lemma A.6 – implies asymptotic convergence of $\tilde{\theta}$ to zero and thus $\hat{A} \to A$, $\hat{B} \to B$ and $\hat{f} \to f$ as $t \to \infty$. Note that the exponential convergence stated in Lemma A.6 does not hold here due to the presence of ε in (7.35).

Finally, combine the convergence of $\hat{A}$, $\hat{B}$ and $\hat{f}$ with the convergence of ε_A, ε_B and ε_f shown in the proof of Theorem 7.7. For the assumed full column rank of B, it can then be concluded that also the control gains converge to their nominal values: $K_x \to K_x^*$, $K_r \to K_r^*$ and $k_f \to k_f^*$ as $t \to \infty$. □

After deriving direct and indirect MRAC control algorithms for PWA systems, let us briefly revise their preferred use cases as well as advantages and disadvantages. The direct MRAC algorithm presented in Section 7.2 is ideally suited for reference system tracking alone. In case a linear reference system is chosen, or a CQLF for the reference subsystems is known, Theorem 7.4 guarantees asymptotic state tracking without imposing additional assumptions on the reference signals. If, however, the choice of reference systems is restricted and no CQLF is found, one must resort to Theorem 7.5 and impose additional requirements on the system as well as the reference signals in order to achieve asymptotic state tracking. The dominant requirement are sufficiently rich reference signals to ensure PE and in turn guarantee convergence of the control gains to the nominal gains. As converged gains imply perfect reference system matching, direct MRAC according to Theorem 7.5 should also be applied if precise tracking under non-periodic reference signals is required. Independent of the existence of a CQLF this task is only achievable for arbitrary reference signals if the nominal control gains are applied.

The ideal use case for the indirect MRAC algorithm presented in Section 7.3 is when monitoring of subsystem parameters becomes the main goal besides reference system tracking alone. Applications can be the early detection of parameter drifts or aging components in the system. From this point of view, indirect MRAC according to Theorem 7.7 has limited practical relevance as it only achieves the asymptotic state tracking of a reference system for which a CQLF is known. No guarantees are given for the convergence of parameter estimates. Hence, in this case, the benefit of a simpler structure in the direct MRAC law is more convincing. Yet, the advantage of the indirect MRAC strategy is that control gains are updated indirectly over the intermediate closed-loop estimation errors, such as ε_A. Therefore, the control gains are updated less aggressively which can reduce the wear associated with strongly oscillating control gains. The convincing full potential of indirect MRAC is given in form of Theorem 7.10. That is, by exciting the system with sufficiently rich reference signals, it is guaranteed that the estimated subsystem parameters converge to the true values, which is ideal for monitoring purposes. Unfortunately the application of indirect MRAC is restricted to reference systems with CQLF as the convergence of subsystem parameters is asymptotic. Therefore, a stability proof as carried out for direct MRAC is infeasible as the discontinuities in the Lyapunov function can not be compensated by a combination of exponential convergence and dwell-time constraints.



(a) Two-cart system **(b)** Characteristic of spring stiffness

Fig. 7.2: The two-cart system in (a), whose connecting spring has the piecewise affine stiffness characteristic shown in (b), can be described by a PWA model.

7.4 Simulation Studies

For numerical testing of the proposed adaptation schemes, consider the mass-spring-damper system shown in Fig. 7.2(a). The system consists of two carts with masses $m_1 = 5\,\mathrm{kg}$ and $m_2 = 1\,\mathrm{kg}$ connected to each other and the environment by springs and dampers. All dampers have fixed damping coefficients $d = 1\,\mathrm{Ns/m}$. The left cart is connected to the environment by a constant spring with stiffness $c_0 = 1\,\mathrm{N/m}$. Let the stiffness of the spring between the two masses be piecewise affine with characteristic $F_c(p_1, p_2)$ shown in Fig. 7.2(b), which suggest a PWA system with three subsystems. The characteristic is assumed to be linear with slope $c_1 = 10\,\mathrm{N/m}$ for $|p_2 - p_1| < \gamma = 1\,\mathrm{m}$. If the spring is extended beyond γ, the slope of the spring characteristic decreases to $c_2 = 1\,\mathrm{N/m} < c_1$. For strong spring compression, i.e. $p_2 - p_1 < -\gamma$, the slope of the spring characteristic increases to $c_3 = 100\,\mathrm{N/m} > c_1$. As Fig. 7.2(b) indicates, the change in slope yields affine terms in regions Ω_2 and Ω_3 of the form $\gamma(c_1 - c_2)$ and $-\gamma(c_1 - c_3)$, respectively. The two-cart system in Fig. 7.2 can be modeled by a PWA system with three subsystems. Subsystem 2 for instance takes the form

$$\dot{x} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{(c_0+c_2)}{m_1} & -\frac{2d}{m_1} & \frac{c_2}{m_1} & \frac{d}{m_1} \\ 0 & 0 & 0 & 1 \\ \frac{c_2}{m_2} & \frac{d}{m_2} & -\frac{c_2}{m_2} & -\frac{2d}{m_2} \end{bmatrix}}_{A_2} x + \underbrace{\begin{bmatrix} 0 & 0 \\ \frac{1}{m_1} & 0 \\ 0 & 0 \\ 0 & \frac{1}{m_2} \end{bmatrix}}_{B_2} u + \underbrace{\begin{bmatrix} 0 \\ \frac{\gamma(c_1-c_2)}{m_1} \\ 0 \\ -\frac{\gamma(c_1-c_2)}{m_2} \end{bmatrix}}_{f_2}$$

with state vector $x = [p_1, \dot{p}_1, p_2, \dot{p}_2]^\top$ and input vector $u = [F_1, F_2]^\top$. The two inputs, in form of forces F_1 and F_2 applied to the two carts, make the system multivariable. The adaptive control laws proposed in this chapter are the first for such multivariable PWA systems. For the following simulation studies we assume all system parameters in A_i, B_i and f_i, $i \in \{1, 2, 3\}$ to be unknown.

With the proposed algorithms, the system can be forced to track the dynamics of a reference system. For the reference system it is desirable to remove the nonlinearity as well as the coupling between the two masses. This motivates the following choice of reference

system

$$\dot{x}_\mathrm{m} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ -25 & -10 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -25 & -10 \end{bmatrix}}_{A_\mathrm{m}} x_\mathrm{m} + \underbrace{\begin{bmatrix} 0 & 0 \\ 25 & 0 \\ 0 & 0 \\ 0 & 25 \end{bmatrix}}_{B_\mathrm{m}} r,$$

which corresponds to a transfer function matrix with all poles located at -5:

$$\begin{bmatrix} p_1(s) \\ p_2(s) \end{bmatrix} = \begin{bmatrix} \frac{1}{(0.2s+1)^2} & 0 \\ 0 & \frac{1}{(0.2s+1)^2} \end{bmatrix} \begin{bmatrix} r_1(s) \\ r_2(s) \end{bmatrix}.$$

Hence, the reference signals r_1 and r_2 can be used as decoupled set points for the cart positions p_1 and p_2, respectively.

In the following, the direct and indirect MRAC algorithms are analyzed in three settings. First, the tracking ability according to Theorems 7.4 and 7.7 is validated for simple reference signals. Afterwards, the convergence derived in Theorems 7.5 and 7.10 is confirmed for sufficiently rich reference signals.

7.4.1 Reference System Tracking

In order to test the tracking ability, let the reference signals be

$$r_1(t) = 3\sin(0.5t) \quad \text{and} \quad \bar{r}_2(t) = \begin{cases} -2, & \text{for } kT \le t < kT + 10\,\mathrm{s} \\ 0, & \text{for } kT + 10\,\mathrm{s} \le t < kT + 20\,\mathrm{s} \\ 2, & \text{for } kT + 20\,\mathrm{s} \le t < kT + 30\,\mathrm{s} \end{cases}$$

where $k \in \mathbb{N}$ and $T = 30\,\mathrm{s}$. These signals yield a sinusoidal set point for the left cart and a piecewise constant set point for the right one. Furthermore, the reference signals drive the controlled system through all three partitions of the state space. The state of the reference system and two-cart system are initialized with $x_\mathrm{m} = 0_n$ and $x = [-2, 0, 2, 0]^\top$, respectively. Moreover, Gaussian noise with zero mean and 0.1 variance is added to the state measurements x during simulation. All controller gains and parameter estimates are initialized with zeros.

The design parameters were chosen as follows. Let all scaling constants $(\Gamma_{xi}, \Gamma_{ri}, \Gamma_{fi})$ be equal to one, and let the design matrices be

$$Q_\mathrm{m} = \begin{bmatrix} 100 & 10 & 0 & 0 \\ 10 & 100 & 0 & 0 \\ 0 & 0 & 100 & 10 \\ 0 & 0 & 10 & 100 \end{bmatrix}, \qquad P = \begin{bmatrix} 140 & 2 & 0 & 0 \\ 2 & 5.2 & 0 & 0 \\ 0 & 0 & 140 & 2 \\ 0 & 0 & 2 & 5.2 \end{bmatrix},$$

which also incorporates the desired decoupling of the two carts. Since the input effectiveness of the forces F_1 and F_2 with respect to the accelerations $\ddot{p}_1$ and $\ddot{p}_2$ is known, and since each force does not directly affect the opposite cart, and since the same holds for the reference system, it can be anticipated that all K_{ri}^* are symmetric and positive definite.

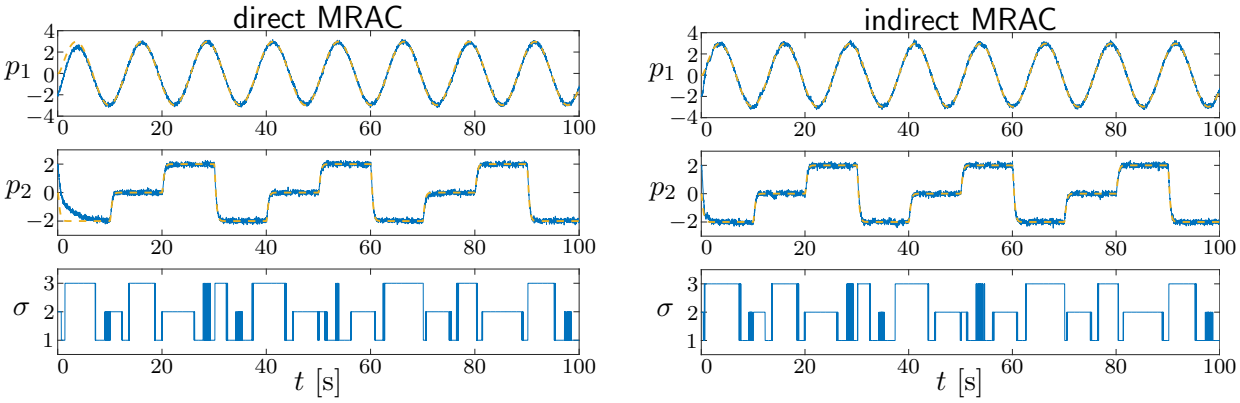


Fig. 7.3: Direct and indirect MRAC of the PWA two-cart system enables decoupled control of both carts despite frequent switching. Solid lines represent the measured states of the system and dashed lines indicate the states of the reference system.

This, according to Remark 7.6, suggests the choice $S = I_2$.

Figure 7.3 shows the tracking performance of the direct and indirect MRAC law, respectively. As can be seen from the figures, both algorithms yield the desired performance despite frequent switching and the considerable parameter variations between the subsystems. The same performance could not be obtained by a single adaptive controller operating in all partitions.

Note that the frequent switching is a consequence of noisy state measurements. The switching signal used in the controller is therefore not equal to the switching signal of the actual system. As the closed-loop system nonetheless exhibits the desired behavior, the simulation results suggest that the proposed algorithm contains some robustness against uncertainties in the assumed partitioned state space.

7.4.2 Parameter Estimation

Next, the derived convergence properties of the direct and indirect MRAC laws are verified with the reference signals

$$\begin{aligned} r_1(t) &= 0.4\sin(t) + 0.3\sin(4t) + 0.2\sin(11t) \\ r_2(t) &= \bar{r}_2(t) + 0.4\sin(2t) + 0.3\sin(10t) + 0.2\sin(14t), \end{aligned}$$

where $\bar{r}_2(t)$ is the piecewise constant reference signal used in the tracking example above. Besides exciting the system in all modes, these reference signals ensure the required properties of being sufficiently rich of at least order 5 with distinct frequencies. All design parameters and initial values are the same as in the previous example. The added measurement noise has zero mean and variance 10^{-3}.

Figure 7.4 exemplifies the evolution of gains for subsystem 1 under direct MRAC. Note that gains for region 2 and 3 evolve similarly and are thus not shown for clarity. The figure shows that adaptation pauses in time intervals in which the system evolves in region 2 and 3, visualized here by the dotted segments. Despite repeated switching, all gains converge to the ideal values visualized by dashed lines and specified by the matching conditions (7.8).

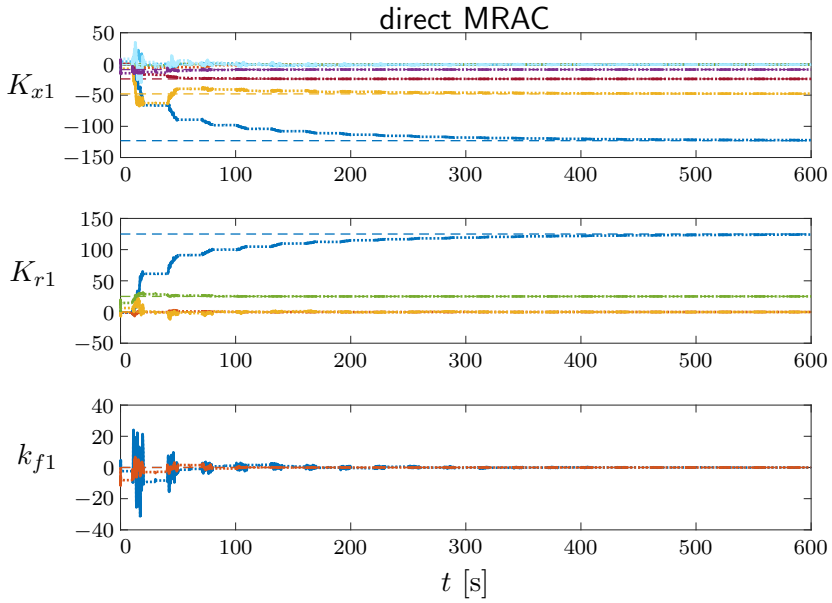


Fig. 7.4: Convergence of controller gains under direct MRAC for subsystem 1. Dashed lines represent ideal gains; solid lines correspond to the estimated gains while the subsystem 1 is active ($\chi_1 = 1$) and dotted lines visualize the estimated gains during inactive periods of subsystem 1 ($\chi_1 = 0$).

This meets the expectations raised by Theorem 7.5.

The parameter trend for subsystem 1 under indirect MRAC is given in Fig. 7.5 (again, the performance in region 2 and 3 is similar and therefore omitted). Besides the control gains K_{x1}, K_{r1} and k_{f1}, the figure also visualizes the norm of the error in the estimated parameters: $\|\tilde{\theta}_1\| = \|\mathrm{vec}([\tilde{A}_1, \tilde{B}_1, \tilde{f}_1])\|$. As derived in Theorem 7.10, all errors tend to zero asymptotically despite switching.

One difference between direct and indirect MRAC is the strength with which control gains are adjusted. Especially k_{f1} is adjusted less aggressively in the indirect case (note that the scales differ by a factor of ten).

7.5 Summary

This chapter presents a direct and an indirect algorithm for model reference adaptive control of multivariable piecewise affine systems. With the presented algorithms, a PWA system is controlled in such a way, that its input-output behavior asymptotically follows the one of a reference system chosen by the designer. For this purpose the control structure is a combination of state-feedback and input feed-forward. Due to unknown system parameters, the correct control parameters, which render the controlled system equal to the reference model are initially unknown. Hence, initial estimates for the control gains need to be adjusted over time to match the controlled system behavior to the reference system. In the direct case, the tracking error between reference model and controlled system directly drives the adaptation of control gains. In the indirect case, the system parameters are estimated with parameter identifiers introduced in Chapter 4, and the control gains are

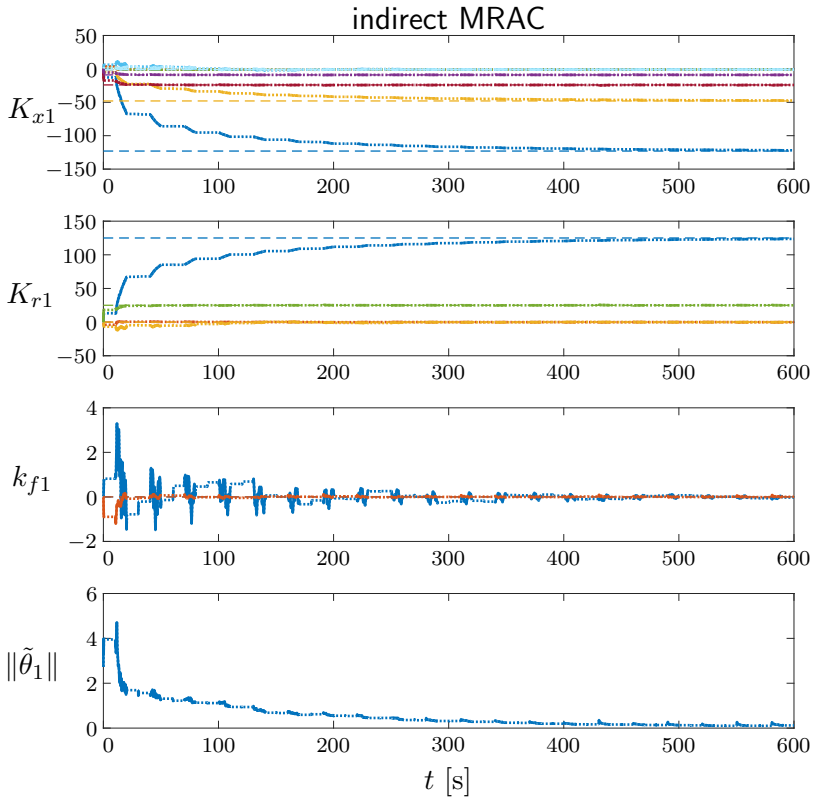


Fig. 7.5: Parameter convergence under indirect MRAC for subsystem 1. Solid lines represent parameters during the active periods of subsystem 1 ($\chi_1 = 1$) and dotted lines visualize the inactive periods of subsystem 1 ($\chi_1 = 0$). The ideal control gains are given by dashed lines.

dynamically adjusted based on estimated system parameters. In both cases, asymptotic tracking of the reference system is shown for arbitrary fast switching and sliding modes in case a common Lyapunov function for the reference system is known. As the reference system is chosen by the designer this, is usually the case. Without a common Lyapunov function, a dwell-time constraint on the switching signal is given for the direct approach. Furthermore, convergence of all parameter errors and tracking errors to zero is guaranteed for sufficiently rich reference signals.

The limitations of the proposed algorithms are twofold. First, the full state feedback restricts the class of applicable systems and limits the choice of reference systems. This limitation can be overcome by output feedback, which is to be studied in future work. The second limitation of the proposed algorithms is the assumption of full state measurement and known state-space partitions. While the conducted simulation studies show some degree of robustness against asynchronous switching, the proposed algorithms become inapplicable for completely unknown state-space partitions. Therefore, future work should also focus on extending the presented ideas to uncertain switching hyperplanes. From a practical point of view, also the reduction of discontinuities in the control inputs at switching times is an important topic for future work.

Part III

Similarity-based Controller Selection

8 Coverage Control for Model Distribution in Multiple Models Adaptive Control

In this thesis, switching phenomena were so far assumed to be an inherent property of the considered system, which is to be identified or controlled. In that context, switching constitutes an obstacle and complicates the task. In contrast to being an obstacle, however, switching is also applied in control engineering to overcome limitations of linear control concepts. The concept of Multiple Models Adaptive Control (MMAC) for instance is used to control a linear system with very large parameter uncertainties by switching between a finite set of robust controllers [2, 43, 44, 112, 130, 177]. As opposed to traditional adaptive control, such as the MRAC concept applied in Chapter 7, where control gains are slowly updated until they match the new system parameters, switching is faster and results in less transient effects. Hence, in this context switching is an enabling factor in dealing with systems characterized by large uncertainties.

This chapter solves a long standing problem in the Multiple Models Adaptive Control framework, which is the selection of the set of candidate controllers. Based on the Coverage Control principle two algorithms are derived which yield an optimal distribution of the candidate controller set. The remainder of this chapter is an excerpt from the journal article [279] and structured as follows. First, Section 8.1 gives an overview of the state of the art in MMAC and the controller selection in particular. Afterwards, Section 8.2 gives a more detailed introduction into the MMAC framework. Then, Section 8.3 revises some robust control results needed to characterize a finite controller covering of the uncertainty set and design suitable robust controllers. The problem of optimal model distribution is formulated in Section 8.4 as a coverage control problem, for which two algorithms are proposed. Testing the proposed algorithms in two benchmark simulations is subject in Section 8.5. Conclusions are drawn in Section 8.6.

8.1 Introduction

Uncertain models are a common challenge in the control of dynamical systems. In order to deal with small uncertainties, research on robust control theory led to various schemes, such as μ-synthesis or $\mathcal{H}_\infty$ loop-shaping, to design a single controller with fixed parameters, which is capable of robustly stabilizing all systems from a given uncertainty set with specified performance properties. For large uncertainties, however, a single robust controller, stabilizing all possible parameter configurations, may be infeasible. Consider for instance, faults in the system, which result in considerable changes in parameters and therefore render the nominal controller unable to stabilize the faulty system. Hence, very large uncertainties require adaptive mechanisms in order to adjust the controller in such a way that it matches the current system parameters.

One direction followed in adaptive control is to continuously tune the controller gains and show that adaptation causes the gains to converge to values ensuring both stability and tracking. Difficulties in continuous adaptation arise in case the parameter change is large and sudden, which induces undesirable transient effects. In order to overcome this limitation, the concept of multiple model adaptive control with switching and tuning [107, 177] may be helpful. By reinitializing adaptation from a finite set of possible parameter configurations, i.e. multiple models, the transient effects are reduced. Note that the continuous adaptation of control gains in [107, 177] precludes the application of robust control concepts due to the complicated relationship between robust and adaptive control structures.

Later, it was noted that neglecting the continuous tuning of control gains enables the application of results from robust control in the multiple models framework. Such algorithms are referred to as *Multiple Model Adaptive Control (MMAC)* [2, 3, 112] or *Supervisory Control* [110, 111, 169, 170, 190]. They have in common that the large uncertainty set is divided into smaller subsets, such that for each subset there exists a fixed controller to robustly stabilize all possible systems in the subset. In [2], it was shown that, under mild assumptions on the uncertainty, a finite set of candidate model-controller pairs suffices to cover the uncertainty set. The task then is to select at each time instance from the set of candidate controllers the most suitable one. For that purpose, a first component of the framework characterizes the match between the true system and the multiple models. This is often carried out by predicting the output of the true systems with multi-estimators (observers) or a bank of Kalman-filters. Then, performance signals are defined as the conditional probabilities obtained from the Kalman-filters or by suitably integrating the prediction errors. Based on the performance signals, a second component, frequently referred to as supervisor, selects, weights or blends the candidate controllers. In early papers, the focus was on a discontinuous switching logic, which selects one of the controllers and puts it into feedback with the systems. In that context, various switching strategies haven been investigated involving dwell-time constraints and hysteresis switching to ensure safe switching [3, 110, 112]. Stability is concluded for constant system parameters if the switching logic ceases switching in finite time and settles upon a stabilizing controller. One problem arising in discontinuous switching schemes are transients induced by controller initialization. Some results on bumpless transfer [51] and optimal reset maps [206] aim to reduce such effects. Besides hard switching among candidate controllers, some mixing and blending strategies have been proposed [21, 83, 100, 130]. Overall, adaptive schemes which rely on mixing or blending replace the discontinuous switching logic with a smooth and stable interpolation. While this avoids undesirable effects of the switching behavior, such algorithms react slower to large parameter changes than switching-based algorithms can.

The unfalsified control framework [25, 214] introduced by Safonov *et al.* is closely related to MMAC algorithms in a sense that an uncertain system is stabilized by one controller taken from a set of controllers. The difference is that the controller set is designed without a set of nominal models, and the controllers are selected with the help of fictitious reference signals that enable a virtual evaluation of all controllers at the same time. Poorly performing controllers are falsified (discarded) until a set of unfalsified controllers remains, which satisfy the given performance specifications. Recent extensions by Baldi *et al.* introduced

more robustness in the controller design and extended unfalsified control to multi-input systems [19, 20].

In conclusion, MMAC can be decomposed into designing switching strategies and finding a finite set of controllers to cover the uncertainty set. While a brief introduction into the framework including switching will be given, the primary focus of this paper is on the systematic controller distribution. Approximately optimal algorithms to determine the minimum number of nominal models needed and their distribution in the uncertainty set will be derived. Unavoidable problem approximations and assumptions leading to suboptimal solutions will be highlighted and discussed as they are introduced.

The performance of multiple models-based algorithms depends to a large extent on the selected set of candidate controllers. The dominant questions in this context are: How many candidate models are needed and how do they need to be distributed in the uncertainty set to achieve the best result? Most publications on the topic, however, avoid the question about how to optimally select candidate model-controller pairs. Most often, a set of suitable candidate controllers is simply assumed to be given [49, 110, 147] or obtained by trial and error [169]. Some authors acknowledge the design of candidate controller sets as a challenging open problem [2, 83, 147]. Anderson *et al.* for instance conclude with "It further remains to find a computationally efficient method to determine such a finite set, so that the number of models needed is not overly conservative" [2].

In spite of its importance, only a few systematic algorithms exist for determining the model distribution in MMAC. The authors in [83] propose an approach to determine the required number of models in their robust MMAC algorithm, which is essentially a greedy search, i.e., beginning from the upper bound of the uncertainty set, the subset assigned to the first model is extended until the desired performance is reached. At this point the next model is initialized and the process is repeated until the entire uncertainty set is covered. The increased complexity of such greedy algorithms in higher dimensional parameter spaces constitutes the main limitation of this approach. A similar greedy approach involving the ν-gap metric [252, 253] to characterize stability margins was proposed in [2] with the same limitations. In [159], four heuristics also involving the ν-gap metric are given for determining model distributions, one of which is a more systematic version of the greedy approach in [2]. A bisectional method to find a robustly stabilizable partitioning of the initial uncertainty set is proposed in [94]. Starting with the entire uncertainty set as the first subset, the proposed algorithm performs repeated bisections of all subsets for which a single robust controller is infeasible. Finally, for various early algorithms which did not incorporate the desired robustness properties, see [148] and the references therein.

In this chapter we propose a new systematic approach for model distribution in MMAC. In contrast to existing greedy search algorithms, the proposed approach builds upon a formal optimization problem formulated with tools form the robust control literature. More precisely, we make use of the ν-gap metric to characterize distances between candidate model-controller pairs and the uncertain plant configurations. Furthermore, the ν-gap metric enables stability statements to be made. Finally, robust candidate controllers are designed with standard $\mathcal{H}_\infty$ tools. To solve the optimization problem we propose two coverage control-inspired algorithms. With both algorithms, two types of optimization are practicable. Either, the number of model-controller pairs may be reduced while ensuring

stabilizability of all systems in the uncertainty set. Or, for a fixed number of model-controller pairs, their distribution can be optimized to increase control performance as rated by the scalar coverage functional. This chapter demonstrates the effectiveness of the proposed algorithms in the context of two benchmark examples.

Note that MMAC is a highly modular framework in which various units such as estimators, candidate controllers or the design of switching strategies can be considered independently and independent methodologies can be easily substituted. A very detailed and modular treatment of MMAC is given in [43] and [44]. Hence, the synthesis of a finite controller covering presented in this paper is quite general and can be combined with other switching strategies or frameworks related to MMAC. Also note that the considered question of how to divide uncertainty sets also has applications in the field of linear time varying (LTV) and linear parameter varying (LPV) systems as pointed out for instance in [256]: "How to divide and what the number of subsets is are interesting research questions of their own". Also in [255], the gap metric is used to determine the operating points of local linear models in LPV systems. The LPV system is then obtained by interpolating the local models. Another greedy search algorithm to increase validity regions in LPV systems is discussed in [108]. Besides LTV and LPV system synthesis, another field of possible application for the proposed coverage control algorithms is gain scheduling.

The ideas presented in this chapter are based on the ν-gap metric, which provides stability guarantees ensuring boundedness of all signals. However, no guarantees can be given regarding the avoidance of unsafe states. Hence, another interesting application are formal methods to provide guarantees in time domain [212, 213].

8.2 Multiple Models and Supervisory Control

Consider a system $\mathcal{G}(q)$ with uncertain parameter vector $q \in \mathbb{R}^n$ taking values from the compact *uncertainty set* $\mathcal{Q} \subset \mathbb{R}^n$. More precisely, let the system matrices $A(q)$, $B(q)$, $C(q)$ and $D(q)$ of the state-space realization define

$$\mathcal{G}(q) \quad \begin{cases} \dot{x} = A(q)x + B(q)u \\ y = C(q)x + D(q)u, \end{cases} \tag{8.1}$$

and depend continuously on q. Assume furthermore that the uncertainty set $\mathcal{Q}$ is too large for a single fixed controller to stabilize all possible configurations. A central result in the MMAC literature [2, Th. 2.1] states that, under the mild assumption of continuity of $\mathcal{G}(q)$ in the ν-gap metric, there exists a finite set of controllers, whose corresponding set of robustly stabilized plants covers the uncertainty set. In other words, we can always find a finite set of candidate controllers $\{\mathcal{C}_i\}_{i=1}^N$ such that each system $\mathcal{G}(q)$ in the uncertainty set $\mathcal{Q}$ is stabilized by at least one candidate controller. Let i be an index taking values from the index set $\mathcal{I} := \{1, \ldots, N\}$ to express which one of the N models or controllers is considered.

One way to design the set of candidate controllers is to choose a finite set of *nominal configurations* $p_i \in \mathcal{Q}$ and design a robust controller for the corresponding *nominal model* $\mathcal{G}(p_i)$. How many and how to choose the nominal configurations will be presented in

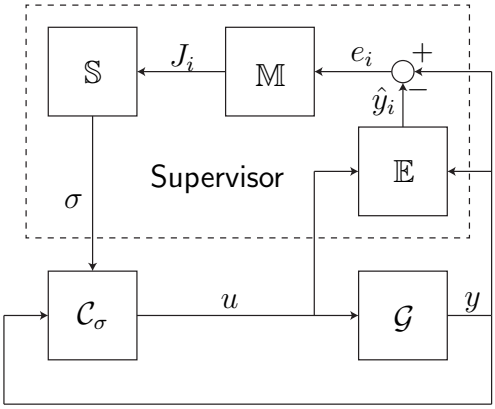


Fig. 8.1: Supervisor in Multiple Model Adaptive Control.

Section 8.4 as the main contribution of this chapter. Given the set of nominal model-controller pairs, MMAC evaluates which nominal model best describes the current system input-output behavior and applies the corresponding nominal controller. Hence, the heart of every MMAC algorithm is a multi-controller

$$\begin{aligned} \dot{x}_{\mathcal{C}} &= A_{\mathcal{C}_{\sigma(t)}} x_{\mathcal{C}} + B_{\mathcal{C}_{\sigma(t)}} y \\ u &= C_{\mathcal{C}_{\sigma(t)}} x_{\mathcal{C}} + D_{\mathcal{C}_{\sigma(t)}} y \end{aligned} \tag{8.2}$$

with shared state $x_{\mathcal{C}}$, whose matrices switch between the state-space realizations of *nominal controllers* $\mathcal{C}_i := \{A_{\mathcal{C}_i}, B_{\mathcal{C}_i}, C_{\mathcal{C}_i}, D_{\mathcal{C}_i}\}$ designed for the nominal models $A(p_i), B(p_i), C(p_i), D(p_i)$. The switching is represented in (8.2) by the switching signal $\sigma(t) : \mathbb{R}^+ \to \mathcal{I}$.

Various strategies for logic-based switching in MMAC have been proposed since its establishment. We consider here the one proposed in [112], due to its simplicity. Note that due to the modularity of the MMAC framework, the model distribution presented in this chapter can and should also be combined with other switching strategies. In this context, we also want to stress that issues related to stability of the switching logic, the number of switches and chattering effects strongly depend on the applied switching logic and will therefore not be considered here.

Figure 8.1 shows the supervisor proposed by Hespanha *et al.* [112], which contains the three components multi-estimator, monitoring signal generator and hysteresis switching logic to orchestrate the switching in the multi-controller (8.2). First, a *multi-estimator* $\mathbb{E}$ generates estimated system outputs $\hat{y}_i$ for each nominal model $\mathcal{G}(p_i), i \in \mathcal{I}$ based on the measured control inputs u and outputs y of the uncertain system. With the estimated system outputs $\hat{y}_i$, estimation errors

$$e_i = \hat{y}_i - y \tag{8.3}$$

are formulated. Inside the *monitoring signal generator* $\mathbb{M}$, these estimation errors are translated into performance signals $J_i > 0$ which express the fitness of the nominal models.

A popular choice is to consider a weighted integral over the estimation errors, such as

$$J_i(t) = \alpha e_i^\top(t) e_i(t) + \beta \int_0^t \mathrm{e}^{-\lambda(t-\tau)} e_i^\top(\tau) e_i(\tau) d\tau, \tag{8.4}$$

with design constants $\alpha, \beta, \lambda \geq 0$. The third component, called *switching logic* $\mathbb{S}$, generates the switching signal $\sigma(t)$ based on the performance signals. The simplest switching logic is to select the nominal controller with the best performance, i.e. smallest error signal. With noisy measurements, however, this causes too frequent switching, which motivated the hysteresis switching logic in [112]. Suppose that the last switch occurred at time t_j and the switching signal is now $\sigma(t) = k \in \mathcal{I}$. Under hysteresis switching, the value of σ remains fixed until a time instance $t_{j+1} > t_j + \epsilon$ is reached, at which

$$(1+h) \min_{i \in \mathcal{I}} J_i(t_{j+1}) \leq J_k(t_{j+1}), \tag{8.5}$$

with hysteresis constant $h > 0$. The dwell ϵ is then bounded below by a finite value. At t_{j+1} the switching signal takes the value $\sigma(t) = \arg\min_{i \in \mathcal{I}} J_i(t_{j+1})$ until the hysteresis switching condition (8.5) is satisfied the next time.

The stability proof for the interconnection of the uncertain system (8.1) with the state-shared multi-controller (8.2) under hysteresis switching (8.5) is by now well understood. It must first be shown that for a constant parameter vector q, the supervisor ceases switching in finite time with a finite number of switches. Then, the controller on which the supervisor settles must be confirmed to stabilize the linear time invariant system $\mathcal{G}(q)$. For continuous uncertainty sets, convergence to a single most appropriate controller is not guaranteed under the above hysteresis switching logic. Hence, additional measures need to be taken on the side of the supervisor such as hierarchical hysteresis switching [110] or continuous mixing of nominal controllers [130]. As details on the supervisor are not in the scope of this thesis, we refer the interested reader to [110, 112, 169, 170] for more details.

8.3 Robust Controller Design

In order to implement MMAC, one needs to ensure that there exists a stabilizing controller for each plant in the uncertainty set. Such guarantees can be given with results from the robust control literature [98, 101, 166, 226, 252, 253], to be presented in this section. First, we revise how to express distances between linear time invariant systems in terms of the ν-gap metric and how these distances relate to the stability margin of nominal model-controller interconnections. Afterwards, the calculation of the best possible stability margin for a given system is described. Finally, the synthesis of a robust controller with desired stability margin is detailed.

8.3.1 Characterizing stabilizable Systems with the ν-Gap Metric

The ν-gap metric was introduced to the field of robust control by Vinnicombe [252, 253] in the 1990s, providing a measure of the maximum distance between a nominal and a perturbed system under which the controller, which was designed for the nominal system,

still stabilizes the perturbed system. For the supervisory control setting, it is necessary to determine those systems

$$\mathcal{G}(s) = C\left(sI - A\right)^{-1} B + D \tag{8.6}$$

from the uncertainty set, which are sufficiently close to the i-th nominal model

$$\mathcal{G}_i(s) = C_i\left(sI - A_i\right)^{-1} B_i + D_i \tag{8.7}$$

and that they are stabilized by the corresponding controller $\mathcal{C}_i$. Here, I is the identity matrix of matching dimension.

The *ν-gap* between $\mathcal{G}_i$ and $\mathcal{G}$ is defined as

$$\delta_\nu(\mathcal{G}_i, \mathcal{G}) := \left\| (I + \mathcal{G}\mathcal{G}^*)^{-\frac{1}{2}} (\mathcal{G} - \mathcal{G}_i)(I + \mathcal{G}_i^*\mathcal{G}_i)^{-\frac{1}{2}} \right\|_\infty \tag{8.8}$$

given that the following winding number conditions [1]

$$\begin{aligned} \det(I + \mathcal{G}^*\mathcal{G}_i)(j\omega) \neq 0 \quad \forall\omega \quad &, \text{ and} \\ \text{wno } \det(I + \mathcal{G}^*\mathcal{G}_i) + \kappa(\mathcal{G}_i) - \kappa(\mathcal{G}) &= 0, \end{aligned} \tag{8.9}$$

are satisfied [252], where $\mathcal{G}^*$ denotes the complex conjugate of $\mathcal{G}$. If the conditions are not satisfied, then let $\delta_\nu(\mathcal{G}_i, \mathcal{G}) = 1$. With the definition above, the value of δ_ν is between 0 and 1. Systems that are close to each other in the ν-gap metric have a value close to 0, whereas systems that are very different have a maximum distance of 1. Efficient implementations of the ν-gap metric are available, such as the function `gapmetric` in the Robust Control Toolbox of Matlab.

The most important property of the ν-gap metric is related to the *generalized stability margin* $b_{\mathcal{G}_i,\mathcal{C}_i}$ of a nominal model $\mathcal{G}_i$ with the nominal controller $\mathcal{C}_i$ in the feedback loop. If the feedback interconnection $(\mathcal{G}_i, \mathcal{C}_i)$ is stable, the generalized stability margin is defined by

$$b_{\mathcal{G}_i,\mathcal{C}_i} := \left\| \begin{bmatrix} \mathcal{G}_i \\ I \end{bmatrix} (I - \mathcal{C}_i\mathcal{G}_i)^{-1} \begin{bmatrix} -\mathcal{C}_i & I \end{bmatrix} \right\|_\infty^{-1}, \tag{8.10}$$

otherwise $b_{\mathcal{G}_i,\mathcal{C}_i} = 0$. The following statement reveals the importance of the ν-gap metric [252]: Given a nominal model $\mathcal{G}_i$, a nominal controller $\mathcal{C}_i$ and a number β, then $(\mathcal{G}, \mathcal{C}_i)$ is stable for all systems $\mathcal{G}$, satisfying $\delta_\nu(\mathcal{G}_i, \mathcal{G}) \leq \beta$ if, and only if, $b_{\mathcal{G}_i,\mathcal{C}_i} > \beta$. That means the nominal controller $\mathcal{C}_i$, which in feedback with the nominal model $\mathcal{G}_i$ achieves a generalized stability margin greater than β, also stabilizes all systems $\mathcal{G}$ located in an β-radius (in the ν-gap metric) around the nominal model $\mathcal{G}_i$.

[1] Here, wno is the winding number evaluated on the standard Nyquist contour indented at the imaginary axis poles of $\mathcal{G}$ and $\mathcal{G}_i$. The number of poles of a transfer function $\mathcal{G}$ in the open right half plane are denoted by $\kappa(\mathcal{G})$.

8.3.2 Determining the maximum Radius of stabilizable Systems

Now, given a nominal model $\mathcal{G}_i$, two useful questions may be asked: What is the most robust controller in the ν-gap metric for this system? And, what is the maximum generalized stability margin that can be achieved? To answer these questions, let $\mathcal{G}_i$ be written in terms of the normalized left coprime factorization (LCF)

$$\mathcal{G}_i = M_i^{-1} N_i, \tag{8.11}$$

where normalization means

$$N_i N_i^* + M_i M_i^* = I, \tag{8.12}$$

which is equivalent to the matrix $[N_i, M_i]$ being coinner. (See Lemma 2.1 in [101] for how to calculate a normalized LCF).

The most robust controller is found as the solution to the following $\mathcal{H}_\infty$-optimization problem [101]:

$$\inf_{\mathcal{C}_i} \left\| \begin{bmatrix} \mathcal{C}_i \\ I \end{bmatrix} (I - \mathcal{G}_i \mathcal{C}_i)^{-1} M_i^{-1} \right\|_\infty = \left(1 - \left\| \begin{bmatrix} N_i & M_i \end{bmatrix} \right\|_H^2 \right)^{-\frac{1}{2}}, \tag{8.13}$$

where $\|\cdot\|_H$ is the Hankel norm. Denote the obtained infimum as $\gamma_{\mathcal{G}_i,\min}$ and note that it also yields an upper bound for the achievable generalized stability margin of the system $\mathcal{G}_i$. In [98, 101], it is shown that

$$\left\| \begin{bmatrix} \mathcal{C}_i \\ I \end{bmatrix} (I - \mathcal{G}_i \mathcal{C}_i)^{-1} M_i^{-1} \right\|_\infty = \left\| \begin{bmatrix} \mathcal{G}_i \\ I \end{bmatrix} (I - \mathcal{C}_i \mathcal{G}_i)^{-1} \begin{bmatrix} -\mathcal{C}_i & I \end{bmatrix} \right\|_\infty, \tag{8.14}$$

which is due to the fact that $[M_i, N_i]$ is coinner and that the $\mathcal{H}_\infty$ norm is invariant under right multiplication by coinner functions [226, Ex. 9.7]. Hence, the maximum generalized stability margin for a system $\mathcal{G}_i$ – as considered in the ν-gap metric – is

$$b_{\mathcal{G}_i,\max} = \gamma_{\mathcal{G}_i,\min}^{-1} = \left(1 - \left\| \begin{bmatrix} N_i & M_i \end{bmatrix} \right\|_H^2 \right)^{\frac{1}{2}} > 0. \tag{8.15}$$

8.3.3 Designing Controllers with specific Stability Margin

Note that the application of the most robust controller is often undesirable as controller synthesis is a trade-off between robustness and performance. Hence, the most robust controller may not deliver the desired performance. Assume therefore that a desired stability margin $b_{\mathcal{G}_i,\mathrm{des}} < b_{\mathcal{G}_i,\max}$ is chosen by the designer and a suitable controller satisfying this stability margin is to be found. By reducing the desired stability margin, the resulting controller can be configured for improved performance, but guarantees closed-loop stability only for smaller perturbations of the nominal system.

In [101], suitable controllers are synthesized by solving a Nehari extension problem. Here, we follow the approach presented in [226, Sec. 9.4]. Given a minimal state-space

representation $[A_i, B_i, C_i, D_i]$ for $\mathcal{G}_i$ and a desired threshold $\gamma_{\mathcal{G}_i,\text{des}} > \gamma_{\mathcal{G}_i,\min}$, then let

$$R_i := I + D_i D_i^\top, \qquad S_i := I + D_i^\top D_i \tag{8.16}$$

and solve the two algebraic Riccati equations

$$\left(A_i - B_i S_i^{-1} D_i^\top C_i\right)^\top X_i + X_i \left(A_i - B_i S_i^{-1} D_i^\top C_i\right) - X_i B_i S_i^{-1} B_i^\top X_i + C_i^\top R_i^{-1} C_i = 0 \tag{8.17}$$

and

$$\left(A_i - B_i S_i^{-1} D_i^\top C_i\right) Z_i + Z_i \left(A_i - B_i S_i^{-1} D_i^\top C_i\right)^\top - Z_i C_i^\top R_i^{-1} C_i Z_i + B_i S_i^{-1} B_i^\top = 0 \tag{8.18}$$

for the unique matrices X_i and Z_i. With

$$F_i := -S_i^{-1} \left(D_i^\top C_i + B_i^\top X_i\right) \tag{8.19}$$

$$L_i := (1 - \gamma_{\mathcal{G}_i,\text{des}}^2) I + X_i Z_i, \tag{8.20}$$

the state-space realization of a controller $\mathcal{C}_i$ satisfying $b_{G_i,\text{des}} = \gamma_{\mathcal{G}_i,\text{des}}^{-1}$ is given by

$$\begin{aligned} A_{\mathcal{C}_i} &:= A_i + B_i F_i + \gamma_{\mathcal{G}_i,\text{des}}^2 (L_i^\top)^{-1} Z_i C_i^\top (C_i + D_i F_i) \\ B_{\mathcal{C}_i} &:= \gamma_{\mathcal{G}_i,\text{des}}^2 (L_i^\top)^{-1} Z_i C_i^\top \\ C_{\mathcal{C}_i} &:= B_i^\top X_i \\ D_{\mathcal{C}_i} &:= -D_i^\top. \end{aligned} \tag{8.21}$$

Finally, note that the synthesis of robust controllers with specific stability margin can be combined with performance criteria in the form of loop shaping. An introduction on how to include loop shaping ideas into the $\mathcal{H}_\infty$-controller synthesis above is given in [166] and [226, Sec. 9.4.2].

To conclude the revision of tools from the robust control literature, let us repeat the three essential findings which will enable optimized model distribution in MMAC. First, given a nominal model $\mathcal{G}_i$, one can determine its maximal stability margin $b_{\mathcal{G}_i,\max}$ with (8.15). Second, a nominal controller $\mathcal{C}_i$ for $\mathcal{G}_i$ satisfying a specific stability margin $b_{\mathcal{G}_i,\text{des}} < b_{\mathcal{G}_i,\max}$ can be synthesized according to (8.21). Third, the fact that $\mathcal{C}_i$ stabilizes all systems $\mathcal{G}$ for which $\delta_\nu(\mathcal{G}_i, \mathcal{G}) < b_{\mathcal{G}_i,\mathcal{C}_i}$ paves the way to partitioning the uncertainty set into various subsets, each associated with one nominal model-controller pair. Previous approaches to MMAC such as [2, 83, 94, 159] incorporate the same idea to investigate – based on the ν-gap metric – whether the chosen set of nominal controllers can stabilize all possible system configurations. So far the focus was, however, only on providing stability guarantees. In the next section we present a new algorithm, which not only ensures the existence of a stabilizing controller for each system configuration but also provides an optimized partitioning of the uncertainty set in a sense that less nominal systems are needed.

8.4 Systematic Model Distribution through Coverage Control

In this section, the optimal distribution of nominal models is phrased as a coverage control problem. The coverage control problem is extensively studied in the field of multi-agent systems in the robotics community, where a set of robots/agents autonomously covers an area of interest for monitoring purposes [33, 61, 77, 229]. Most algorithms build upon the classic work by Lloyd [153] on optimal quantizer selection, which was first applied to the distributed robotic coverage problem in [61]. Initial algorithms were concerned with convex environments and the Euclidean distance. More recent works focus on coverage control algorithms for non-convex domains or manifolds and arbitrary distances [33, 77, 229] as well as heterogeneous agents [204] (e.g. robots with different sensing capabilities). For MMAC, we transfer the coverage-control framework from the physical space, in which robotic agents operate, to an abstract space in form of the model uncertainty, in which the nominal models need to be placed. In this analogy, we replace the target environment of the robotic agents by the uncertainty set. The abstract counterpart to the robotic agents are the nominal model-controller pairs, which will therefore also be referred to as agents. Distances in the MMAC setting are measured in terms of the ν-gap metric instead of the Euclidean distance.

This section begins with the formulation of the coverage control optimization problem in 8.4.1, followed by a continuous gradient descent approach in 8.4.2. For computational reasons a discretized version of the coverage control problem is introduced in 8.4.3. Then, a first algorithm performing a gradient descent on the discrete graph is derived in 8.4.4. Afterwards, a second algorithm, which is based on pairwise optimizations, is presented in 8.4.5. Finally, a brief summary in 8.4.6 lists potential sources for suboptimality.

8.4.1 Continuous Coverage Control Problem

Consider the *uncertainty set* $\mathcal{Q} \subset \mathbb{R}^n$ in which the uncertain parameters q of the controlled system reside. Hence, each $q \in \mathcal{Q}$ represents a potential configuration of the system. We assume that the uncertainty set is characterized by a convex polytope[2]. In other words, there exist known bounds on the entries of the parameter vector q, e.g. $q_{(1),\min} < q_{(1)} < q_{(1),\max}$. Next, let the *distribution density function* $\phi : \mathcal{Q} \to \mathbb{R}_{\geq 0}$ introduce a notion of relevance/importance of each configuration in the set $\mathcal{Q}$. This importance can be seen as some prior on the probabilities of certain parameter configurations. Thus more likely or more important parameter configurations should be assigned a larger value in ϕ. If all configurations are equally important, let $\phi(q) = 1, \forall q \in \mathcal{Q}$.

The goal in MMAC is to stabilize all possible system configurations from the uncertainty set $\mathcal{Q}$ by a total of $N \in \mathbb{N}$ nominal systems and their associated controllers. We refer to these systems as *nominal system configurations* $p_i \in \mathcal{Q}$ and let all chosen configurations be combined in the set $\mathcal{P} = \{p_1, \ldots, p_N\}$. It is desirable to distribute the models in such

[2]The assumption of convex uncertainty sets is made for simplicity. In case $\mathcal{Q}$ is non-convex or even unconnected, one may consider a convex hull $\bar{\mathcal{Q}}$ of the original uncertainty set and assign a distribution density function of zero to all configurations which are not from the original set, i.e. $\phi(q) = 0, \forall q \notin \mathcal{Q}$.

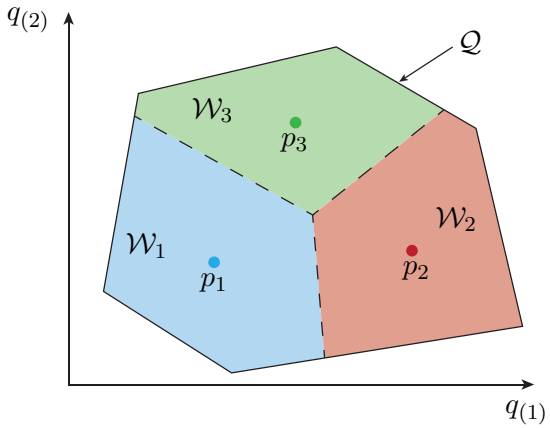


Fig. 8.2: Visualization of the coverage problem.

a way that for each potential system configuration $q \in \mathcal{Q}$ there is at least one model $p_i \in \mathcal{P}$ whose controller stabilizes the system q. Let this relationship be expressed by a *partitioning* $\mathcal{W} = \{\mathcal{W}_1, \ldots, \mathcal{W}_N\}$ of the uncertainty set $\mathcal{Q}$. The partitions are such that their union corresponds to the complete uncertainty set $\cup_{i=1}^{N} \mathcal{W}_i = \mathcal{Q}$, such that there is no overlap $\mathcal{W}_i \cap \mathcal{W}_j = \emptyset, i \neq j$, and such that each partition is nonempty $\mathcal{W}_i \neq \emptyset$ and connected, $\forall i, j \in \mathcal{I} := \{1, \ldots, N\}$. Hence, each system configuration $q \in \mathcal{Q}$ belongs to just one partition $\mathcal{W}_i$ and its nominal model configuration p_i. The controller associated with the model p_i shall be responsible to stabilize all potential systems $q \in \mathcal{W}_i$. Figure 8.2 visualizes an example of how the uncertainty set $\mathcal{Q}$ of a two dimensional parameter vector is covered by three models and their partitions.

Next, introduce a *metric* $d(q, p_i) : \mathcal{Q} \times \mathcal{Q} \to \mathbb{R}_{\geq 0}$ which defines the distance between a configuration q and the nominal model configuration p_i. As pointed out above, the ν-gap metric in (8.8) is ideally suited for such distance measures for control design applications. Hence let $d(q, p_i) := \delta_\nu(\mathcal{G}(q), \mathcal{G}(p_i))$. It is furthermore well known that the $\mathcal{H}_\infty$ performance of the closed-loop system, whose controller was designed for the nominal configuration p_i, degrades with growing distance from the nominal configuration. Let this relationship be specified by a function $f(d) : \mathbb{R} \to \mathbb{R}_{\geq 0}$, which we refer to as the *inverse control performance*. Let $f(d(q, p_i))$ be strictly increasing with the distance $d(q, p_i)$. Hence a small value in $f(d(q, p_i))$ indicates that the controller designed for p_i performs well for q. A large value on the other side indicates that the system q cannot be robustly stabilized by the nominal controller for p_i. The increase in f with decreasing performance motivates the name inverse control performance. A common choice for f in coverage control with homogeneous agents/robots is $f(d) = d^2$.

In the present setting of coverage control for model distribution in MMAC, however, we observe that the nominal models (i.e. agents) may be very heterogeneous due to varying stability margins. Take for instance two nominal models p_1 and p_2 with different stability margins, e.g. $b_{\mathcal{G}_1,\max} < b_{\mathcal{G}_2,\max}$. In this case, a perturbed system $\mathcal{G}(q)$ may be stabilized only by the nominal controller associated with p_2 even if $\delta_\nu(\mathcal{G}(q), \mathcal{G}(p_1)) < \delta_\nu(\mathcal{G}(q), \mathcal{G}(p_2))$, given that $\delta_\nu(\mathcal{G}(q), \mathcal{G}(p_2)) < b_{\mathcal{G}_2,\max}$ and $\delta_\nu(\mathcal{G}(q), \mathcal{G}(p_1)) > b_{\mathcal{G}_1,\max}$. In order to accommodate this heterogeneity, we apply ideas presented in [204] and introduce the *stability-distance*

$$d_P(q, p_i) := \delta_\nu(\mathcal{G}(q), \mathcal{G}(p_i)) - b_{\mathcal{G}_i,\max}, \tag{8.22}$$

which essentially incorporates the maximum stability radius into the distance measure. Compared to the distance $d(q, p_i)$, the stability-distance $d_P(q, p_i)$ contains more information in a sense that systems q with $d_P(q, p_i) < 0$ can, and systems with $d_P(q, p_i) > 0$ cannot, be stabilized by the (most robust) nominal controller $\mathcal{C}_i$ with stability margin $b_{\mathcal{G}_i,\max}$. In order to reflect the instability for $d_P > 0$, the inverse control performance is defined to be

$$f(d_P) = e^{\Gamma d_P}, \tag{8.23}$$

which grows exponentially with d_P and the scalar design parameter $\Gamma > 0$.

With the above definitions, introduce the following cost function, which rates the expected performance of the distribution $\mathcal{P}$ and the partitioning $\mathcal{W}$,

$$\mathcal{H}(\mathcal{P}, \mathcal{W}) := \sum_{i=1}^{N} \int_{\mathcal{W}_i} f(d_P(q, p_i))\phi(q)\mathrm{d}q. \tag{8.24}$$

This cost function is also referred to as *coverage function* as it indicates how well the set of model-controller configurations $\mathcal{P}$ covers the considered parameter uncertainties $\mathcal{Q}$ partitioned by $\mathcal{W}$. The lower the cost $\mathcal{H}$, the better are the models distributed and the better is the expected control performance. Hence, the goal of optimally distributing the nominal models p_i is formulated as

$$\underset{\mathcal{P},\mathcal{W}}{\text{minimize}}\, \mathcal{H}(\mathcal{P}, \mathcal{W}). \tag{8.25}$$

It is well known [7, 204] that the so called *Voronoi diagrams* $\mathcal{V} = \{\mathcal{V}_1, \ldots, \mathcal{V}_N\}$ with

$$\mathcal{V}_i(\mathcal{P}) := \{q \in \mathcal{Q} \ : d_P(q, p_i) \leq d_P(q, p_j),\ \forall j \neq i\} \tag{8.26}$$

constitute the optimal choice for $\mathcal{W}_i$ given the model configuration $\mathcal{P}$. A Voronoi diagram is a partitioning of $\mathcal{Q}$ in which the elements $q \in \mathcal{V}_i$ have the closest stability-distance to the nominal model p_i. As $\mathcal{V}$ is a function of all nominal models in $\mathcal{P}$, it suffices to optimize the parameter configuration $\mathcal{P}$ and apply the optimal Voronoi diagrams, i.e. $\mathcal{H}_\mathcal{V}(\mathcal{P}) = \mathcal{H}(\mathcal{P}, \mathcal{V}(\mathcal{P}))$, which simplifies (8.25) to

$$\underset{\mathcal{P}}{\text{minimize}}\, \mathcal{H}_\mathcal{V}(\mathcal{P}). \tag{8.27}$$

Above, it was assumed that the partitions are connected. In order to ensure this property for the Voronoi diagrams $\mathcal{V}_i$, an additional assumption needs to be imposed on the system. That is, the system $\mathcal{G}$ and its uncertainty set $\mathcal{Q}$ are such that the distance in the ν-gap metric $\delta_\nu(\mathcal{G}(p), \mathcal{G}(q_2))$ is monotonically increasing for all points q_2 along the shortest path in $\mathcal{Q}$ between p and q_1. Figure 8.3 visualizes this property for the scalar case. This assumption is hard to validate in practice due to the possibly complicated relationship between a parameter q and the ν-gap. Note however that violating this assumption only leads to a minor disadvantage in the form of an increased number of nominal systems.

In practice, the number N of nominal model-controller pairs in the optimization problem (8.27) is usually unknown and must also be determined. For this purpose note that finding

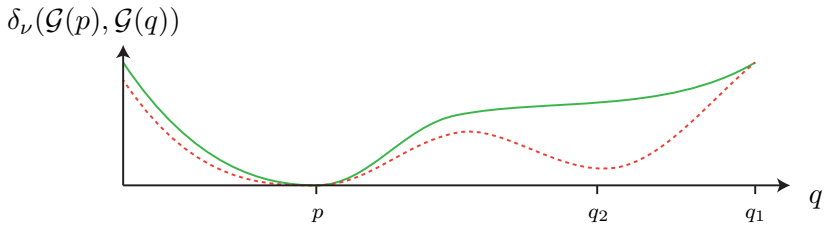


Fig. 8.3: While the first system (green solid line) satisfies the assumption of the monotonically increasing ν-gap along the shortest path between p and q_1, the second system (red dashed line) does not, which may result in additional nominal models.

the optimum in (8.27) alone does not ensure that each configuration q is stabilized by at least one nominal controller p_i. Hence, the additional inequality constraints

$$\min_{i \in \mathcal{I}} d_P(q, p_i) < 0, \quad \forall q \in \mathcal{Q} \tag{8.28}$$

need to be imposed and checked after optimization. Hence, the minimum number of required model-controller pairs may either be determined by increasing N from $N = 1$ until the optimum in (8.27) satisfies the constraints (8.28). Otherwise, if a conservative number $\bar{N}$ is known, one can start from $N = \bar{N}$ and decrease N until (8.27) cannot be solved without violating (8.28).

An important assumption regarding the partitions is that the switching logic selects for a system $q \in \mathcal{V}_i$ the nominal controller p_i, $\forall i \in \mathcal{I}$. If this is not the case, the obtained performance is at best suboptimal, i.e. a model with greater distance in the ν-gap metric and therefore worse control performance would be chosen. If p_i was the only nominal controller to stabilize q, this may even cause instability. This assumption is to be ensured by the switching logic, which is not discussed here. Our primary concern is to obtain a finite covering for which there exists at least one stabilizing controller for each system in the uncertainty set.

8.4.2 Gradient Descent

As shown in the coverage control literature [33, 61, 229], the coverage problem (8.27) can be solved by a distributed gradient descent, in which each agent (here each nominal model) only needs to know the location of the agent in its neighboring partitions. Let $\mathcal{P}$ be initialized with some initial models $p_i(0)$ and let the uncertainty set be partitioned by the Voronoi diagrams $\mathcal{V}_i$. Then, a gradient-descent approach for (8.27) moves each model p_i in the direction of the steepest descent given by $-\frac{\partial \mathcal{H}}{\partial p_i}$. This motivates updating the nominal model parameters in $\mathcal{P}$ by

$$\dot{p}_i = -\rho \frac{\partial \mathcal{H}}{\partial p_i}, \quad i \in \mathcal{I} = \{1, \ldots, N\}, \tag{8.29}$$

with scaling constant $\rho > 0$. Note that the domains of integration in (8.24) depend on $\mathcal{V}_i$, which in turn depends on the configurations in $\mathcal{P}$. Hence, the update $\dot{p}_i$ would intuitively depend not only on p_i and $\mathcal{V}_i$ but also on neighboring partitions. However, with

differentiation under integration it was shown in [204] that the gradient with respect to the i-th model reduces to

$$\frac{\partial \mathcal{H}}{\partial p_i} = \int_{\mathcal{V}_i} \frac{\partial f(d_P(q,p_i))}{\partial p_i} \phi(q) \mathrm{d}q \tag{8.30}$$

$$= \int_{\mathcal{V}_i} \frac{\partial f(d_P(q,p_i))}{\partial d_P(q,p_i)} \left(\frac{\partial d(q,p_i)}{\partial p_i} - \frac{\partial b_{\mathcal{G}_i,\max}}{\partial p_i} \right) \phi(q) \mathrm{d}q. \tag{8.31}$$

Note that the integral in (8.31) requires evaluating the gradient of the distance metric, i.e.

$$\frac{\partial d(q,p_i)}{\partial p_i}, \tag{8.32}$$

for all $q \in \mathcal{V}_i$, which is computationally expensive. In order to reduce the computational load, it can be exploited that the gradient (8.32) is the same for all other points η on the shortest path – in terms of $d(q,p_i)$ – between q and p_i [33]. That means, moving p_i towards q along the shortest path causes equal reduction in $d(q,p_i)$ and $d(\eta,p_i)$. Hence, for each point q, consider another point $\eta(q) \in \mathcal{Q}$ in the vicinity of p_i which lies on the shortest path between p_i and q, and exploit that

$$\frac{\partial d(q,p_i)}{\partial p_i} = \frac{\partial d(\eta(q),p_i)}{\partial p_i}. \tag{8.33}$$

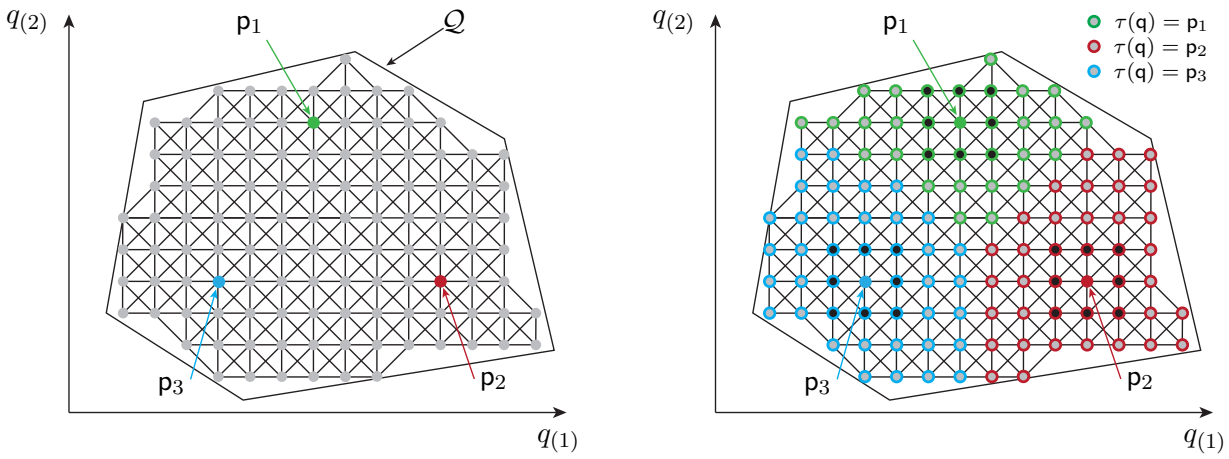
In that case, the evaluation of the gradient at a lot of points q is reduced to the evaluation of the gradient at fewer points $\eta(q)$.

The gradient (8.31) is then given by

$$\frac{\partial \mathcal{H}}{\partial p_i} = \int_{\mathcal{V}_i} \frac{\partial f(d_P)}{\partial d_P} \left(\frac{\partial d(\eta(q),p_i)}{\partial p_i} - \frac{\partial b_{\mathcal{G}_i,\max}}{\partial p_i} \right) \phi(q) \mathrm{d}q. \tag{8.34}$$

Using the gradient descent (8.34) with Voronoi diagrams (8.26) and the ν-gap metric yields a modified application of the continuous-time Lloyd algorithm [153] for optimal model distribution in MMAC. The derived algorithm updates an initial model configuration $\mathcal{P}$ and results in an optimized coverage $\mathcal{H}(\mathcal{P},\mathcal{V})$. The obtained optimum is a so-called *Centroidal Configuration*, i.e., a configuration $(\mathcal{P},\mathcal{V})$ in which $\mathcal{P}$ gives rise to the Voronoi diagrams $\mathcal{V}$ and at the same time each $p_i \in \mathcal{P}$ corresponds to the optimal location in $\mathcal{V}_i$. Note that centroidal configurations are not unique and depend on the initial configuration of $\mathcal{P}$. Therefore, the obtained optimum is only local and the algorithm may have to be initialized from various initial configurations in order to find the global optimum. For more details on centroidal configurations refer to [74].

Unfortunately, the application of the modified Lloyd algorithm remains computationally expensive due to the numerous evaluations of the ν-gap metric in form of d_P under the integral in (8.34). Therefore, in the next step, the uncertainty set is quantized, which enables a more efficient graph-based realization of the coverage problem. This step is in line with algorithms proposed in [77] and [33], where the consideration of the path-length metric in non-convex environments requires quantization.



(a) Initial setting **(b)** Discrete Voronoi partitioning

Fig. 8.4: Visualization of the discrete Voronoi partitioning obtained by Algorithm 2. The nominal vertices p_i in (a) are extended and yield the partitioning shown in (b).

8.4.3 Coverage Control as Graph Search

Let the uncertainty set $\mathcal{Q}$ be approximated by a finite set of models q_k, $k = 1, \ldots, s$ as shown in Fig. 8.4a, where the gray circles correspond to models q_k. These models can be understood as a quantization of the original uncertainty set. Next, define an *undirected weighted graph* $G(Q, E, d)$, whose *vertex set* $Q(G)$ is the set of quantized models, i.e., $Q(G) := \{\mathsf{q}_1, \ldots, \mathsf{q}_s\}$. While q_k refers to a vertex in the graph, let $q(\mathsf{q}_k) \in \mathcal{Q}$ refer to the continuous parameter associated with the vertex q_k. Let neighboring vertices be connected by a weighted edge in the graph (see Fig. 8.4a). Hence, the *edge set* of the graph is $E(G) \subseteq Q(G) \times Q(G)$. Define the *neighborhood* of q_k as $\mathcal{N}(\mathsf{q}_k) = \{\mathsf{q}_l \in Q(G) \ : (\mathsf{q}_k, \mathsf{q}_l) \in E(G)\}$. The considered metric d serves as the *weight map* on the edges $d : E(G) \to \mathbb{R}_+$. The edge weights thus introduce the metric distance between adjacent vertices. In other words, if two nodes $\mathsf{q}_k, \mathsf{q}_l \in Q(G)$ are connected by an edge, i.e. $(\mathsf{q}_k, \mathsf{q}_l) \in E(G)$, then the connecting edge takes the weight $d(q(\mathsf{q}_k), q(\mathsf{q}_l))$. Introducing the graph greatly reduces the computational complexity in a sense that the ν-gap metric only needs to be computed once for each edge during initialization as compared to various combinations $d_P(q, p_i)$ under the integral in (8.34) at each iteration of the gradient descent.

The distance between two arbitrary vertices of G is defined in terms of paths. A *path* in G is a sequence of vertices, in which every pair of consecutive vertices belongs to the edge set $E(G)$. The sum of the edge weights along a path defines the *weight of the path*. The distance between two vertices q_k and q_l in G is given by the weight of the path with the lowest weight and we denote it by $\mathsf{d}_G(\mathsf{q}_k, \mathsf{q}_l)$. As G is undirected, it follows that $\mathsf{d}_G(\mathsf{q}_k, \mathsf{q}_l) = \mathsf{d}_G(\mathsf{q}_l, \mathsf{q}_k)$.

The graph-based representation above restricts the metric d from the continuous uncertainty set $\mathcal{Q}$ to the discrete vertex set $Q(G)$. Doing so is motivated by the computational complexity in the continuous space. The restriction allows for a more efficient graph-search on G. It must be noted, however, that this quantization essentially yields an approximation of the original problem. First, quantization restricts the choice of nominal models to vertices of the graph which may not contain the optimal configurations. Sec-

ond, determining distances on the graph by summing up the edge weights of the shortest path yields an approximation. Due to the triangle inequality it follows that the distance on the graph is always greater or equal to the distance between the same points in $\mathcal{Q}$, i.e., $\mathsf{d}_G(\mathsf{q}_k, \mathsf{p}_i) \geq d(q(\mathsf{q}_k), q(\mathsf{p}_i))$. While the error associated with the graph-based approximation can to some extend be regulated by the density of quantized models q_k, the quantization nonetheless remains a trade off: Accuracy and elegance of the continuous representation against computational efficiency of the discrete graph.

For the coverage problem, select N *nominal vertices* $\mathsf{p}_i, i = 1, \ldots, N$ and refer to $\mathsf{P} = \{\mathsf{p}_1, \ldots, \mathsf{p}_N\}$ as the *set of nominal vertices*. Furthermore, let G be partitioned into N subgraphs $\mathsf{W}_i \subset G$ with the same intention as in the continuous uncertainty set. That is, the nominal models $p_i = q(\mathsf{p}_i)$ should be distributed in such a way that their controllers stabilize all systems associated with the vertices in the *partition* W_i. As in partitioning of the continuous uncertainty $\mathcal{Q}$, the partitions W_i are such that their union corresponds to the complete graph $\cup_{i=1}^{N} \mathsf{W}_i = G$, such that there is no overlap $\mathsf{W}_i \cap \mathsf{W}_j = \emptyset, i \neq j$ and such that each partition is non-empty $\mathsf{W}_i \neq \emptyset$ and connected.

In order to introduce the discrete counterpart to the Voronoi diagrams, let the stability-distance on the graph G be

$$\mathsf{d}_{G,S}(\mathsf{q}, \mathsf{p}_i) := \mathsf{d}_G(\mathsf{q}, \mathsf{p}_i) - b_{\mathcal{G}_i,\max}, \tag{8.35}$$

where $b_{\mathcal{G}_i,\max}$ is the maximal stability margin for the nominal model associated with $q(\mathsf{p}_i)$. Then a Voronoi diagram for G is given by the subgraphs

$$\mathsf{V}_i(\mathsf{P}) = \{\mathsf{q}_k \in Q : \mathsf{d}_{G,S}(\mathsf{q}_k, \mathsf{p}_i) \leq \mathsf{d}_{G,S}(\mathsf{q}_k, \mathsf{p}_j),\ \forall j \neq i\}\,. \tag{8.36}$$

For one of the proposed algorithms, a notion of *adjacent partitions* is needed. Two partitions W_i and W_j with $i \neq j$ are adjacent if there exist two vertices $\mathsf{q}_k \in \mathsf{W}_i$ and $\mathsf{q}_l \in \mathsf{W}_j$ that are connected by an edge in the graph $(\mathsf{q}_k, \mathsf{q}_l) \in E(G)$.

Next, we rewrite the coverage function (8.24) in terms of G. Replacing the integration over $\mathcal{W}_i$ by summation over W_i yields the *discrete coverage function*

$$\mathsf{H}(\mathsf{P}, \mathsf{W}) = \sum_{i=1}^{N} \sum_{\mathsf{q}_k \in \mathsf{W}_i} f\left(\mathsf{d}_{\mathsf{W}_i,S}(\mathsf{q}_k, \mathsf{p}_i)\right) \phi(\mathsf{q}_k). \tag{8.37}$$

For later analysis, we refer to

$$\check{\mathsf{H}}(\mathsf{p}_i, \mathsf{W}_i) = \sum_{\mathsf{q}_k \in \mathsf{W}_i} f\left(\mathsf{d}_{\mathsf{W}_i,S}(\mathsf{q}_k, \mathsf{p}_i)\right) \phi(\mathsf{q}_k) \tag{8.38}$$

as the *coverage of a single partition*.

Finally, let us revise two notions of optimality and their relationship derived in [77]. First, a configuration (P, W) is called *Centroidal Configuration*, if W is the Voronoi diagram according to (8.36), i.e. $\mathsf{W}_i = \mathsf{V}_i(\mathsf{P}), \forall i$, and if $\mathsf{p}_i = \arg\min_{\mathsf{r} \in \mathsf{V}_i} \check{\mathsf{H}}(\mathsf{r}, \mathsf{V}_i), \forall i$. Second, a configuration (P, W) is called *Pairwise-Optimal Configuration* if, for every pair of adjacent partitions W_i and W_j with nominal vertices $\mathsf{p}_i = \arg\min_{\mathsf{r} \in \mathsf{W}_i} \check{\mathsf{H}}(\mathsf{r}, \mathsf{W}_i)$ and $\mathsf{p}_j = \arg\min_{\mathsf{r} \in \mathsf{W}_j} \check{\mathsf{H}}(\mathsf{r}, \mathsf{W}_j)$, the value of the coverage function is lower than for any other

two-partition of the union of W_i and W_j, i.e.,

$$\check{\mathsf{H}}(\mathsf{p}_i, \mathsf{W}_i) + \check{\mathsf{H}}(\mathsf{p}_j, \mathsf{W}_j) \leq \check{\mathsf{H}}(\mathsf{p}_a, \mathsf{W}_a) + \check{\mathsf{H}}(\mathsf{p}_b, \mathsf{W}_b), \tag{8.39}$$

with $\mathsf{p}_a, \mathsf{p}_b \in \mathsf{W}_i \cap \mathsf{W}_j$ and $\mathsf{W}_a, \mathsf{W}_b \subset \mathsf{W}_i \cap \mathsf{W}_j$ and $\mathsf{W}_a \cap \mathsf{W}_b = \emptyset$. An intuitive understanding for the Centroidal Configuration is that W is the optimal partitioning for P, and P is the optimal choice of nominal models for the partitioning W. Hence, it turns out that Centroidal Configurations constitute (local) minima of the discrete coverage function (8.37). The pairwise-optimality adds that for every union of adjacent partitions, one cannot find a better partition. It follows that every Pairwise-Optimal Configuration (P, W) is at the same time a Centroidal Configuration [77, Prop.2.8]. The opposite is not true.

With the preliminaries discussed so far, we are ready to present two algorithms inspired by [77] and [33] to solve the coverage problem (8.25) for near optimal model distribution in MMAC.

8.4.4 Discretized Llyod Algorithm

Building upon ideas presented in [33], we propose a modification of Dijkstra's algorithm [59, 71], which performs both the partitioning with Voronoi diagrams (8.26) and the gradient descent (8.34) on the discrete graph G. The pseudo code of the proposed approach is given by Algorithm 2 with explanations given in the following paragraphs.

First, the partitioning of $Q(G)$ in Algorithm 2 is straightforward and essentially requires three parameters per vertex: The value of $\tau(\mathsf{q}_k)$ determines the closest nominal model, and $g(\mathsf{q}_k)$ and $g_S(\mathsf{q}_k)$ determine the distance and stability-distance along the shortest path between q_k and the vertex of the closest nominal model. Initially, the algorithm assigns infinite distances to each vertex in line 2 and leaves the closest nominal model unspecified. Afterwards, the values of nominal model vertices are overwritten in line 5 and 6 with zero distances, negative stability-distance s and their own indexes (a nominal model is naturally closest to itself). The while-loop from line 12 to 29 describes the main part of the algorithm. At every iteration, the vertex with the lowest $g_S(\mathsf{q}_k)$ (initially one of the nominal model vertices) is chosen and the distance of its neighbors is calculated by adding the weight of the connecting edge (see line 16). Based on the stability-distance obtained in line 17, the algorithm reasons in line 18 whether the currently expanded node is to be added to the same nominal model as its source. By doing so, the set of vertices assigned to one model expands from the nominal model vertex and step by step replace all initially infinite distances by the true values. The wavefronts, expanding from different nominal vertices, collide at the boundary of their Voronoi diagrams. In Fig. 8.4b for instance, the values of $\tau(\mathsf{q}_k)$ obtained with Algorithm 2 are included as color-coded circles.

Algorithm 2 also performs a discretized version of the gradient descent (8.34), for which a few more comments are needed. As was concluded by (8.33), it is beneficial (from a computational point of view) to express the partial derivative of the distance in terms of a reduced set of points η in the vicinity of p_i. In the graph-based representation, the set of neighboring vertices to p_i constitutes a suitable choice for η. During the expansion phase of the Dijkstra algorithm one thus has to keep track of which vertex in the vicinity of p_i lies on the shortest path to a vertex q_k. For this purpose, the variable $\eta(\mathsf{q}_k)$ assigns to

Algorithm 2 Power diagram & gradient descent

for all $\mathsf{q}_k \in Q(G)$ **do** ▷ Initialization
  Set $\tau(\mathsf{q}_k), \eta(\mathsf{q}_k) := \emptyset$, and $g(\mathsf{q}_k), g_P(\mathsf{q}_k) := \infty$
end for
for all $\mathsf{p}_i \in \mathrm{P}$ **do** ▷ Init. vertices of nominal models
  Set $g(\mathsf{p}_i) := 0$, and $g_P(\mathsf{p}_i) := -b_{G_i,\max}$
  Set $\tau(\mathsf{p}_i) := \mathsf{p}_i$, and $\nabla_{\mathsf{H}}(\mathsf{p}_i) := 0$
  for all $\mathsf{q}_k \in \mathcal{N}(\mathsf{p}_i)$ **do**
    Set $\eta(\mathsf{q}_k) := \mathsf{q}_k$
  end for
end for
$\mathsf{Q} := Q(G)$ ▷ List of not-yet-extended nodes
while $\mathsf{Q} \neq \emptyset$ **do**
  Get $\mathsf{r} := \arg\min_{\mathsf{q}_k \in \mathsf{Q}} g_P(\mathsf{q}_k)$ ▷ Extend to node r
  Let $\mathsf{Q} := \mathsf{Q} \backslash \mathsf{r}$
  for all $\mathsf{w} \in \mathcal{N}(\mathsf{r})$ **do** ▷ Update adjacent vertices
    Let $g' := g(\mathsf{r}) + d(\mathsf{r}, \mathsf{w})$
    Let $g'_P := q' - b_{G_{\tau(r)},\max}$
    if $g'_P < g_P(\mathsf{w})$ **then**
      Set $g(\mathsf{w}) := g'$, and $g_P(\mathsf{w}) := g'_P$,
       and $\tau(\mathsf{w}) := \tau(\mathsf{r})$
      if $\eta(\mathsf{r}) \neq \emptyset$ **then**
        Set $\eta(\mathsf{w}) := \eta(\mathsf{r})$
      end if
    end if
  end for
  if $\eta(\mathsf{r}) \neq \emptyset$ **then** ▷ Contribution of r to gradient
    $\nabla_{\mathsf{H}}(\tau(\mathsf{r})) := \nabla_{\mathsf{H}}(\tau(\mathsf{r})) + \frac{\partial f(d_P)}{\partial d_P}\Big|_{d_P = g_P(\mathsf{r})} \cdot \left(\frac{\partial d(\eta(\mathsf{r}), p_i)}{\partial p_i}\Big|_{p_i = \tau(\mathsf{r})} - \frac{\partial b_{G_i,\max}}{\partial p_i}\Big|_{p_i = \tau(\mathsf{r})} \right) \phi(\mathsf{r})$
  end if
end while
for all $\mathsf{p}_i \in \mathrm{P}$ **do** ▷ Choose next nominal models
  Replace p_i by $\arg\max_{\mathsf{w} \in \mathcal{N}(\mathsf{p}_i)} -\nabla_{\mathsf{H}}(\mathsf{p}_i) \cdot (q(\mathsf{w}) - q(\mathsf{p}_i))$
end for

each q_k the unique vertex which is a) in the neighborhood of its associated nominal vertex $\tau(\mathsf{q}_k)$, i.e. $\eta(\mathsf{q}_k) \in \mathcal{N}(\tau(\mathsf{q}_k))$, and b) on the shortest path between q_k and the nominal vertex $\tau(\mathsf{q}_k)$. All possible vertices, that $\eta(\mathsf{q}_k)$ can point to, are highlighted in Fig. 8.4b by circles with black filling. With these definitions in mind, the continuous gradient in (8.34) is approximated by

$$\nabla_{\mathsf{H}}(\mathsf{p}_i) = \sum_{\mathsf{q}_k \in \mathsf{V}_i} \frac{\partial f(d_P)}{\partial d_P}\bigg|_{d_P = \mathsf{d}_{G,S}(\mathsf{q}_k, \mathsf{p}_i)} \cdot \left(\frac{\partial d(\eta(\mathsf{q}_k), p_i)}{\partial p_i} - \frac{\partial b_{G_i,\max}}{\partial p_i} \right) \phi(\mathsf{q}_k). \tag{8.40}$$

Note that for a specific p_i the partial derivative $\frac{\partial d(\eta, p_i)}{\partial p_i}$ is now restricted to a finite set

of cardinality $|\mathcal{N}(\mathsf{p}_i)|$. It is therefore reasonable to compute and store all possible values during the initialization period of the graph.

Analyzing the details of Algorithm 2, one can see the initialization of vertices that qualify as η in line 8. Afterwards, during the expansion period, line 22 determines $\eta(\mathsf{q}_k)$ for each vertex. Based on $\eta(\mathsf{q}_k)$, line 27 calculates the contribution of vertex q_k to the gradient in (8.40). Finally, the best next vertices are determined in line 31 by evaluating the benefit of all possible steps (induced by the grid) in terms of the approximated gradients $\nabla_{\mathsf{H}}(\mathsf{p}_i)$.

Algorithm 2 causes the nominal vertex set P to converge to a centroidal configuration, which corresponds to a local minimum. As the convergence of the gradient descent is deterministic, the same initial configurations will always result in the same centroidal configuration. Hence, the algorithm may have to be executed multiple times with different initial configurations in order to determine the global optimum. In the second algorithm, which is proposed next, the set of minima to which the algorithm converges is reduced by seeking pairwise-optimal configurations.

8.4.5 Pairwise-Optimal Partitioning Approach

In the following, we present a second approach to optimize the distribution of nominal vertices p_i in the vertex set $Q(G)$, which is a modification of the gossip approach presented in [77]. The task in [77] is to monitor a non-convex environment with a team of robots. It is proposed to use the sparse communication (gossip) between adjacent robots in order to compute pair-wise optimal configurations. For the coverage problem in this paper, we can neglect the sparse communication and consider a centralized version of the gossip approach. The pseudo-code is given by Algorithm 3 with additional explanations given in the following paragraphs.

The goal of Algorithm 3 is to decrease the coverage function (8.37) by optimizing pairs of adjacent partitions. Hence, after the initial distribution and partitioning in line 1 of the algorithm, a pair of adjacent partitions is chosen at random in line 3. Next, the adjacent partitions are combined in line 4 before all possible combinations of nominal vertices are listed in S in line 5. Then, the code from line 7 to line 15 iterates over all possible vertex combinations and determines the corresponding coverage function values. The current choice of nominal vertices is replaced in lines 12–13 by a new combination that leads to a lower value in the coverage function. This corresponds to determining the two-pair optimal partition for the union of the adjacent partitions W_i and W_j. Then, the nominal vertices and partitions of the considered partitions are updated with the obtained optimum. This update always yields an improved coverage if p_i, p_j and W_i, W_j are not yet pairwise optimal. Repeating this process for a random sequence of pairs of adjacent partitions ultimately results in a Pairwise-Optimal Configuration. Hence, the algorithm terminates in line 16 if the nominal vertex set P and the corresponding partitions W form a Pairwise-Optimal Partition.

Note that the computational complexity of the pairwise optimization in Algorithm 3 grows with the number of combinations in S. Hence, the finer the quantization of the uncertainty set $\mathcal{Q}$, the more combinations need to be evaluated. For intermediate iterations, this effect can be circumvented by returning sub-optimal improvements.

Algorithm 3 Pairwise-Optimal Partitioning

Initialize p_i and let $\mathsf{W}_i := \mathsf{V}_i(\mathsf{P})$, $i = 1, \ldots, N$
repeat
  Randomly choose p_i, p_j with adjacent W_i, W_j
  Let $\mathsf{U} := \mathsf{W}_i \cup \mathsf{W}_j$
  Let S be a list of all pairs of vertices in U
  Set $\mathsf{p}_a^* := \mathsf{p}_i$, $\mathsf{W}_a^* = \mathsf{W}_i$, and $\mathsf{p}_b^* := \mathsf{p}_j$, $\mathsf{W}_b^* = \mathsf{W}_j$
  for all $(\mathsf{q}_a, \mathsf{q}_b) \in S$ **do**
    Compute partitions
      $\mathsf{W}_a := \{\mathsf{q} \in \mathsf{U} \ : \mathsf{d}_U(\mathsf{q}, \mathsf{q}_a) \leq \mathsf{d}_U(\mathsf{q}, \mathsf{q}_b)\}$
      $\mathsf{W}_b := \{\mathsf{q} \in \mathsf{U} \ : \mathsf{d}_U(\mathsf{q}, \mathsf{q}_a) > \mathsf{d}_U(\mathsf{q}, \mathsf{q}_b)\}$
    if $\check{\mathsf{H}}(\mathsf{q}_a, \mathsf{W}_a) + \check{\mathsf{H}}(\mathsf{q}_b, \mathsf{W}_b) < \check{\mathsf{H}}(\mathsf{p}_a^*, \mathsf{W}_a^*) + \check{\mathsf{H}}(\mathsf{p}_b^*, \mathsf{W}_b^*)$ **then**
      Set $\mathsf{p}_a^* := \mathsf{q}_a$, $\mathsf{W}_a^* := \mathsf{W}_a$,
      and $\mathsf{p}_b^* := \mathsf{q}_b$, $\mathsf{W}_b^* := \mathsf{W}_b$
    end if
  end for
until pairwise-optimal partition for P, W is reached

In summary, it can be noted that Algorithm 2 requires less computational power ($\mathcal{O}(kn^2)$ per iteartion, where k is the average number of quantized models per dimension) but provides only local optima in the form of Centroidal Configurations. Algorithm 3 on the other side converges to Pairwise-Optimal Configurations, which forms a subset of the Centroidal Configurations with reduced values of the coverage function. This reduction comes at the cost of increased computational complexity ($\mathcal{O}(k^3n^4)$). A combination of both algorithms allows to exploit their advantages while attenuating their disadvantages. In the combined algorithm, apply the gradient descent of Algorithm 2 until a Centroidal Configuration is reached. Then apply Algorithm 3 to validate whether the obtained solution is pairwise-optimal. If not, continue the gradient descent on the intermediate result returned by Algorithm 3. Repeat these steps until Algorithm 3 confirms pairwise optimality of the final configuration. While the curse of dimensionality may not be avoided in the presented algorithms, the computational load may be kept low by carefully increasing the number of quantized models k per dimension of the uncertainty set.

8.4.6 Sources of Suboptimality

In this section, we have presented two algorithms to optimize the distribution of candidate model-controller pairs in MMAC. While both algorithms are based on the rigorous coverage function from 8.4.1, various pragmatic considerations lead to problem approximations,which ultimately render the obtained solutions only near optimal. In order to make these sources of suboptimality transparent, a short summary is in order.

The central source for suboptimality is quantization, which immediately restricts the search space to suboptimal configurations. By making the quantization finer, however, the corresponding error can be reduced. Also measuring distances on the graph by summing edge weights yields two sources of suboptimality. First, there is an error due to the triangle inequality, which again may be reduced through finer quantization. Second, the obtained

distance is only accurate under the assumption that the actual ν-gap is monotonically increasing along the shortest path between two models. If this assumption is violated, the partitions $\mathcal{V}_i$ turn out to be unconnected. The presented algorithms would then assign a different partition V_j to each of the connected subsets, which results in a suboptimal number of models. This problem may not be overcome with finer quantization. Overall, the quantization step was made to trade-off computational load and accuracy of the obtained solution. And while quantization enabled the derivation of a systematic approach for optimized model distribution in MMAC, the trade-off subsists in terms of quantization fineness and computational load against suboptimality.

Furthermore, both the gradient descent and the pairwise optimization only guarantee convergence to local minima. Hence, if the algorithms are not evaluated starting from a sufficiently large number of initial configurations, the obtained solution may only correspond to a local minimum in the coverage function.

8.5 Simulation Studies

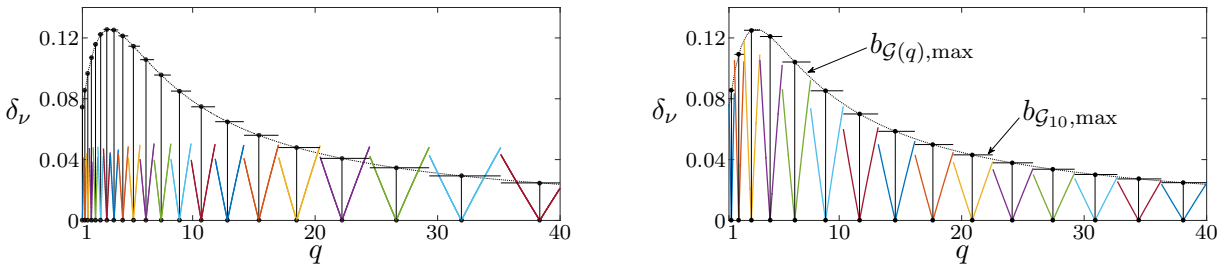
Next, the hybrid algorithm obtained by combining Algorithm 2 and Algorithm 3 is tested in two benchmark examples.

8.5.1 SISO System

To validate the proposed algorithms, first consider the benchmark example studied by Anderson [2] and Morse [169]. For the uncertain linear time invariant transfer function

$$\mathcal{G}(s) = q\frac{s-1}{(s+1)(s-2)}, \qquad q \in [1, 40], \tag{8.41}$$

it was found that a total of 21 nominal model-controller pairs are sufficient to stabilize all configurations from the entire uncertainty set. Note that the scalar multiplicative uncertainty in this example allows to find an infinite continuum of controllers to stabilize the system. Hence, the retrieval of a finite controller covering in this case is (as in [2, 169]) simply motivated by illustrative purposes.



(a) Configuration in [2] with 21 systems.

(b) Configuration with 15 systems obtained by proposed algorithms.

Fig. 8.5: Comparison of nominal model distribution by (a) trail and error and (b) the proposed coverage formulation.

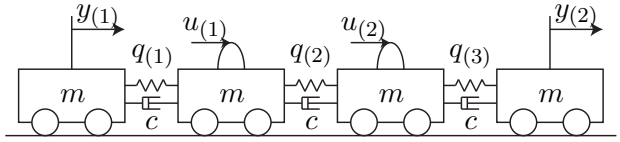


Fig. 8.6: Four-cart system with uncertain spring constants q.

Figure 8.5a analyzes the configurations given in Table 2 of [2] in more detail. The dotted line visualizes the achievable stability margin $b_{\mathcal{G},\max}$ as a function of the uncertain parameter q in (8.41). Next, dots on the q-axis denote the chosen nominal model configurations, which were found in [169] by tedious trial and error. Horizontal lines represent the maximal stability margins of theses nominal models. The colored segments indicate for each configuration q the distance $\min_{p_i} \delta_\nu(\mathcal{G}(q), \mathcal{G}(p_i))$. Stability can be concluded with the ν-gap metric if the colored segments remain underneath the horizontal lines, i.e., $\delta_\nu(\mathcal{G}(q), \mathcal{G}(p_i)) < b_{\mathcal{G}(p_i),\max}$. As the five models to the right of Fig. 8.5a violate this requirement, Anderson *et al.* had to resort to frequency dependent considerations to ensure stability, resulting in additional complexity. Furthermore, Fig. 8.5a shows that the trial and error result does not exploit the maximum stability margin in the interval $q \in [1, 10]$, leading to the overly conservative number of 21 nominal systems.

In order to apply the proposed algorithms, a graph was initialized with equally spaced vertices separated by $\Delta q = 0.01$. With design parameter $\Gamma = 30$ for the inverse controller performance in (8.23), it is possible to determine a finite controller covering with only 15 nominal systems, which are shown in Fig. 8.5b. As $\min_{p_i} \delta_\nu(\mathcal{G}(q), \mathcal{G}(p_i)) < b_{\mathcal{G}(p_i),\max}, \forall q$, it is guaranteed that there exists a stabilizing controller for every possible system configuration. The figure furthermore shows that the larger stability margin in the interval $q \in [1, 10]$ is successfully exploited.

Overall, the proposed algorithm reduces the number of nominal model configurations from 21 to 15 and at the same time removes the need for frequency dependent stability considerations.

8.5.2 Four-Cart System

Next consider the four-cart system in Fig. 8.6, which was previously studied by Baldi *et al.* in [18, 19]. By applying forces $u_{(1)}$ and $u_{(2)}$ to the inner two carts, the positions $y_{(1)}$ and $y_{(2)}$ of the outer two carts are to be controlled. Let the masses $m = 1\,\mathrm{kg}$ and the damping coefficients $c = 0.6\,\mathrm{Ns/m}$ be known. Only the spring constants $q_{(1)}, q_{(2)}, q_{(3)}$ are assumed to be uncertain.

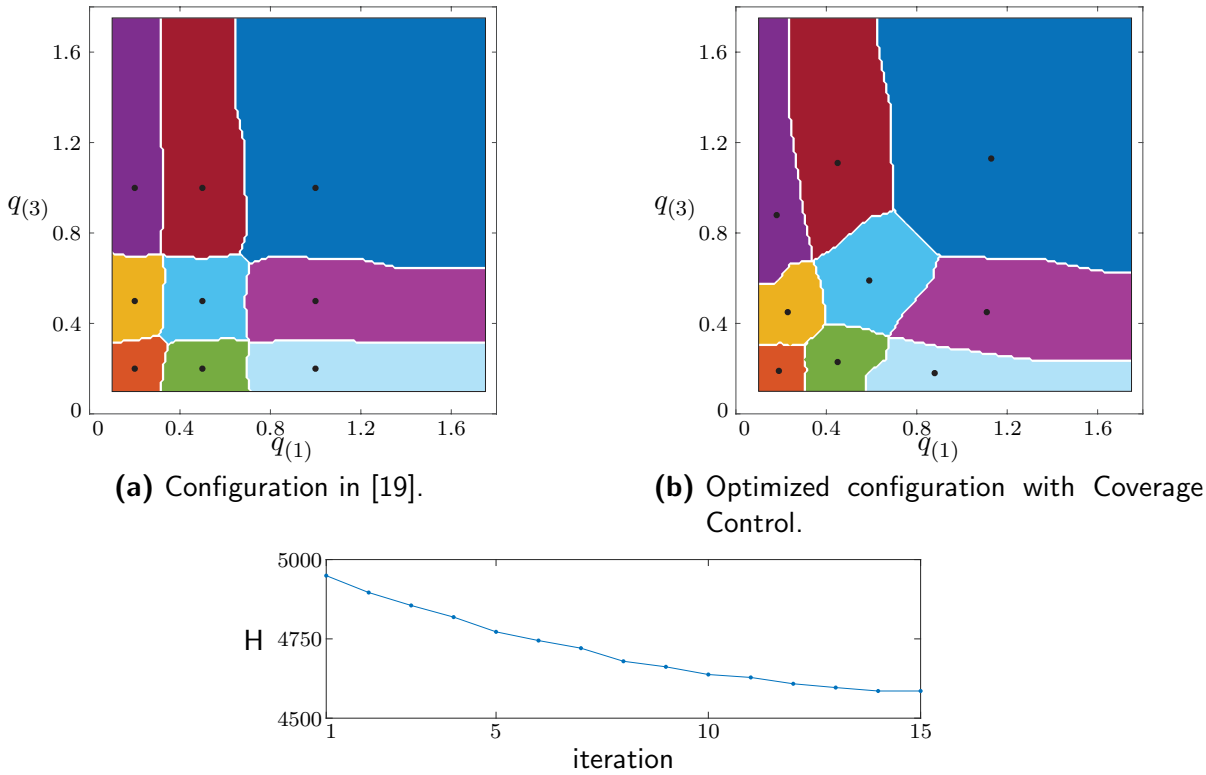
For uncertain outer springs $q_{(1)}, q_{(3)} \in [0.1, 1.75]$ and known inner spring $q_{(2)} = 0.7\,\mathrm{N/m}$, Baldi *et al.* chose the configuration shown in Fig. 8.7a consisting of 9 nominal model-controller pairs. Let the uncertainty set be quantized by a fine grid with equal spacing of $\Delta q_{(i)} = 0.01$. Initializing the proposed optimization algorithm with this configuration and $\Gamma = 10$ shows that further improvements are possible, leading to the optimized configuration shown in Fig. 8.7b. The corresponding evolution of the coverage functional throughout the gradient descent is shown in Fig. 8.7c.

It can furthermore be shown that the distribution of 9 model-controller pairs is too

conservative. Optimization studies reveal that, even if N is reduced to 5 nominal systems, the proposed algorithms converge to a stabilizing controller distribution, which is shown in Fig. 8.8. Even though the initial configuration in [19] did not claim optimality, this example gives a good impression of how much benefit may be generated with the proposed algorithms.

Note that reducing the number of nominal controllers does not compromise stability. This is shown in Fig. 8.9, where reference signal tracking is achieved under time varying spring constants, i.e. $q_{(1)}(t) = 0.2{+}1.6\frac{t}{300}$ and $q_{(3)}(t) = 0.2{+}1.2\sin(0.02t)^2$. The controllers are synthesized with the above introduced combination of $\mathcal{H}_\infty$-controller synthesis and loop shaping [226, Sec. 9.4.2]. Other design constants are $\alpha = \beta = 1$, $\lambda = 0.5$ and $h = 0.2$. Figure 8.9 furthermore shows that fewer nominal models result in less switching, which in turn reduces transient effects in the system outputs due to controller reinitialization [206]. Finally, the visualization of true parameters q and estimated parameters $\hat{q}$, i.e. the constant parameters associated with the currently active model-controller pair, suggests that a greater number of nominal systems enables a more precise tracking of the actual parameters. Nonetheless, a satisfactory tracking is maintained for five optimized models.

Finally, the systematic nature of the proposed algorithms makes the consideration of additional uncertain parameters straightforward. To demonstrate this, assume that also the inner spring constant $q_{(2)}$ takes values from an uncertainty set ranging from 0.7 N/m



(a) Configuration in [19].

(b) Optimized configuration with Coverage Control.

(c) Cost evolution during Coverage Control.

Fig. 8.7: Beginning from the suboptimal configuration in (a), the coverage control methodology yields the optimized configuration shown in (b). The evolution of the coverage functional during optimization is given in (c).

Tab. 8.1: Obtained covering for three uncertain parameters $q_{(1)}, q_{(2)}, q_{(3)}$.

	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}	p_{11}	p_{12}	p_{13}	p_{14}	p_{15}
$p_{i(1)}$	0.2	0.2	0.2	0.3	0.3	0.5	0.6	0.6	0.7	1.0	1.1	1.2	1.3	1.3	1.3
$p_{i(2)}$	0.8	1.3	1.2	1.4	0.9	0.9	1.5	1.2	1.4	0.9	0.8	1.4	1.4	0.9	1.4
$p_{i(3)}$	0.8	1.2	0.2	0.5	0.3	1.0	1.3	0.1	0.6	0.2	0.5	0.2	0.5	1.3	1.3

to 1.7 N/m. Note that this uncertainty set is smaller than before to keep the number of required models reasonable. For a grid with equal spacing of $\Delta q_{(i)} = 0.1$ and design parameter $\Gamma = 10$, it is sufficient to distribute a total of 15 model-controller pairs inside the uncertainty set as shown in Fig. 8.10. The obtained values are given in Table 8.1.

8.6 Summary

The problem in Multiple-Models Adaptive Control (MMAC) is to stabilize a system with parameter uncertainties too large to be handled by a single controller. Therefore, a finite set of nominal model-controller pairs is needed to cover the entire uncertainty set, i.e. ensure that each configuration of the uncertainty set is stabilized by at least one controller. At run-time, MMAC algorithms evaluate all models in parallel and switch the corresponding controller into feedback which is capable of stabilizing the current system parameters. While most research effort focuses on improving the switching logic in MMAC, this chapter derives the first systematic approach to optimize the distribution of nominal model-controller pairs in the uncertainty set.

Covering the uncertainty set by a finite set of model-controller pairs is phrased as a coverage control problem, which is well-known in the multi-agent systems literature. The underlying coverage function evaluates for each configuration the expected control performance of the closest nominal controller and integrates these individual costs to obtain the

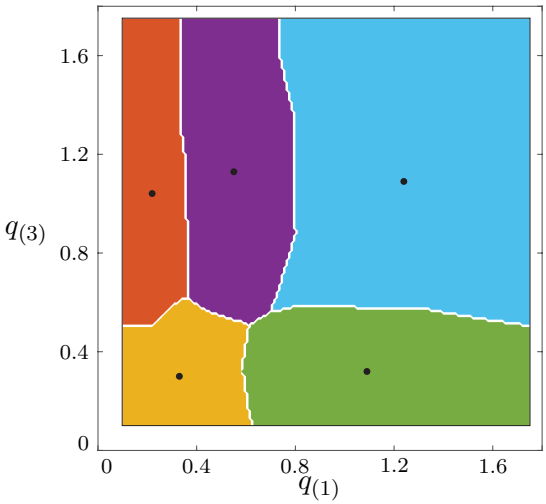
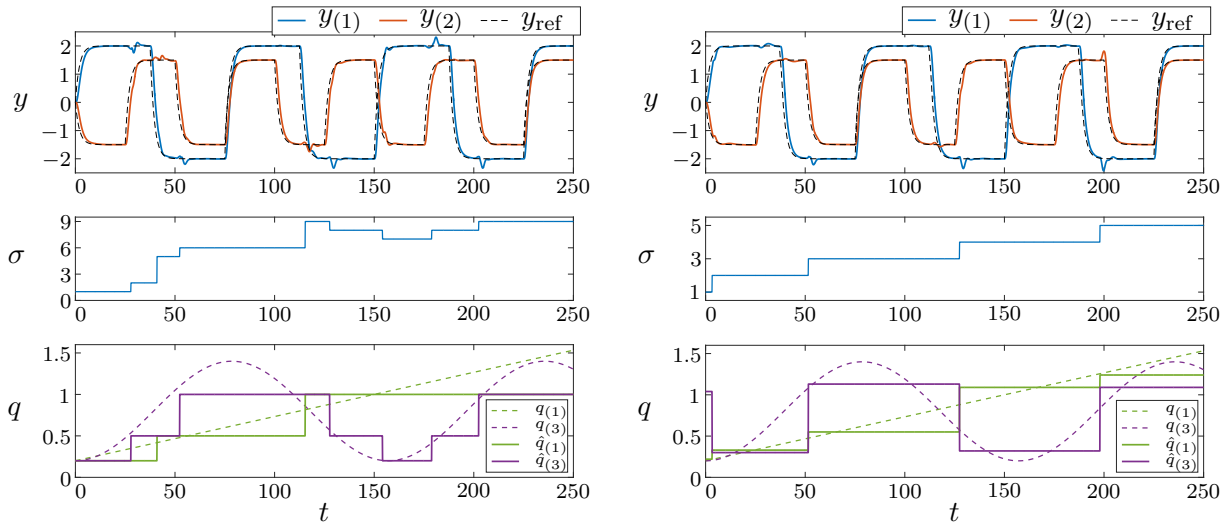


Fig. 8.8: Optimized covering of the two-cart uncertainty set with 5 model-controller pairs.



(a) Nine model-controller pairs from Fig. 8.7(a) **(b)** Five model-controller pairs from Fig. 8.8

Fig. 8.9: Comparison of control performance with (a) the original nine and (b) the five optimized model-controller pairs.

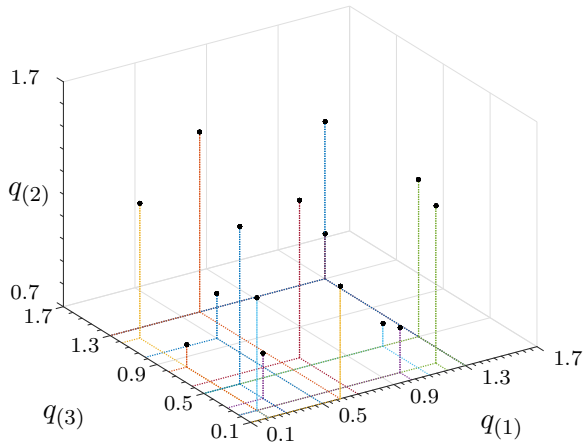


Fig. 8.10: Optimized model distribution for the four-cart system with uncertain parameters $q_{(1)}, q_{(3)} \in [0.1, 1.7]$ and $q_{(2)} \in [0.7, 1.7]$.

overall coverage. Distances are thereby measured in terms of the ν-gap metric, which, in combination with the maximum generalized stability margin, enables stability considerations needed to ensure that each system is stabilizable by at least one nominal controller. As evaluating the coverage function is computationally involved due to the ν-gap metric, a discretization is proposed, which reduces the optimization to an efficient graph-search. For the minimization of the discretized coverage functional, two algorithms are presented. First, a gradient descent drives initially suboptimal configurations to centroidal configurations, which constitute local minima. Second, a pairwise optimization strategy yields pairwise optimal configurations at the cost of increased computational load. Finally, com-

bining the two algorithms yields an algorithm which combines the advantage of the gradient descent in form of reduced computational load with the pairwise optimal solutions obtained by the second algorithm.

The considered class of systems in this chapter is very general as both the ν-gap metric and the generalized stability margin do not pose critical restrictions. Regarding system uncertainties, the algorithms are so far limited to structured uncertainty in form of the uncertainty set. Future work should therefore be directed towards including unstructured uncertainties. While the studies in this chapter were motivated by finding an optimal coverage of the uncertainty set, there are a few sources for suboptimality. First, the discretization, which was introduced to reduce computational complexity, immediately restricts the search-space to a finite set and causes errors in distance evaluation due to triangle inequality. Therefore, finding an efficient solution to the continuous problem formulation could remove the discretization and the corresponding trade-off between accuracy and elegance of the continuous representation against computational efficiency of the discrete graph. Another source of suboptimality is the convergence to local minima, which requires evaluating a large number of initial conditions to increase chances of finding a global optimum.

A central assumption in the presented distribution algorithms is that the switching logic at run-time selects controllers based on the partitioning of the uncertainty set used during distribution. As the actual system parameters at run-time are however unknown, this assumption may not be satisfied by every switching logic. Deriving conditions for switching, which ensure this assumption is left for future work. Another interesting direction for future work is to generalize the presented ideas to closely related fields, where multiple models or controllers need to cover a certain parameter space. This is for instance the case in modeling with linear parameter-varying systems or gain scheduling.

9 Conclusions and Future Work

This thesis was motivated by the increasing complexity of dynamical systems and the resulting challenges for control engineers. More precisely, switching behavior and uncertain parameters are of central importance in this work. The class of piecewise affine (PWA) systems was shown to be suitable for describing such switching phenomena and at the same time maintain an almost linear characteristic, which makes them appealing for control engineers in industry. In PWA systems, the continuous state-input space is partitioned into various regions and each region is assigned some local, linear dynamics. These linear dynamics govern the evolution of the continuous state of the system. The discrete state of the system on the other side is the index of the region that comprises the current state and input of the system. The list of applications, where PWA systems can be used for modeling is long due their universal approximation capabilities. Under the assumption of uncertain parameters, the first two challenges considered in this thesis are the recursive identification (Chapter 3 to 6) and the adaptive control (Chapter 7) of PWA systems.

Besides causing challenges for control engineers, switching may also be the solution to deal with other problems in control. This is the case in Multiple Models Adaptive Control (MMAC), where a switching logic switches between a set of nominal controllers in order to cope with parameter uncertainties too large to be covered by a single, fixed controller. While most research in the field is directed towards developing new switching strategies, the third challenge (Chapter 8) considered in this thesis is the question of how to distribute the set of nominal controllers in MMAC.

9.1 Conclusions

The literature review on state-of-the-art identification algorithms for PWA systems in Chapter 2 and Appendix B reveals a lack of provably converging adaptive identification algorithms for continuous-time PWA systems in state-space form. This lack is filled in Chapter 3 to Chapter 6. As identifying PWA system requires estimating both the state-space partitions and the subsystem dynamics, these two tasks are first investigated separately before a combined algorithm is developed.

Beginning in Chapter 3, it is pointed out that the state-of-the-art support-vector machine based separation to reconstruct the state-space partitions of PWA systems yields unacceptable hyperplanes in the case of continuous-time systems due to an additional delay in the estimation process. To overcome this limitation, Chapter 3 presents an alternative approach to reconstruct the switching hyperplanes more precisely. The approach consists of three distinct steps. First, a hybrid observer estimates the unknown switching signal. As the estimated switching signal is characterized by an inevitable delay, the second step removes this delay by solving a simple optimization problem to obtain the exact switching times. Finally, after collecting a series of switches, the estimated hyperplanes

are fitted to the obtained switches by total least squares. The benefits of the presented approach are the greater precision and the ability to realize it recursively and thus track time-varying hyperplanes. A limiting factor for the hyperplane estimation are dwell-time constraints, which are imposed in order to correctly reconstruct the switching signal.

The recursive estimation of subsystem parameters is subject in Chapter 4. Based on the assumption of a known switching signal, it is shown that parameter identifiers from the adaptive control community can be extended from linear to piecewise linear and piecewise affine systems. In parameter identifiers, time-varying estimates of the unknown subsystem parameters are introduced and updated according to the mismatch between the measured system state and a predicted state. The presented extension realizes a separate identifier for each subsystem of the switched system. The adaptation of each identifier only takes place while the corresponding subsystem is active. In order to ensure stability of the estimation process, the predicted state must be reset at every switch. Stability guarantees are derived based on multiple Lyapunov functions. Besides stability of the resulting hybrid system, the parameter convergence is of fundamental importance. To this end, a condition for persistent excitation of affine subsystems is derived. Given that the excitation of the system is sufficiently rich and the system matrices are invertible, the presented update laws provide converging estimates of all subsystem parameters. Due to the algorithm's recursive nature, the presented identifiers also enable tracking of time-varying parameters, which makes them applicable for monitoring purposes. The largest drawback is certainly the assumption of a known switching signal, which is frequently not available in practice. In that case, a fictitious partitioning by the user may be used as an approximation.

The requirement of sufficiently rich reference signals is common in adaptive systems where parameter convergence is desired. Nonetheless, such a requirement constitutes a limitation as large excitation over extended periods are generally undesirable in practice. To overcome this limitation, Chapter 5 adds concurrent learning to the proposed parameter identifiers. In concurrent learning, adaptation makes concurrent use of instantaneous and recorded data. For this purpose a finite set of measurements of control inputs, states and state derivatives are stored in so-called history stacks. In case, state derivatives are not available, an estimate obtained through fixed point smoothing can be sufficient. The benefit of concurrent learning is that persistent excitation is replaced by the more easily accomplished and verified requirement of linearly independent data in the history stack. Furthermore, exponential convergence of all parameter estimates is achieved independent of switching. The introduction of memory, however, also introduces drawbacks. That is, erroneous or outdated history stack data causes convergence to wrong estimates and generally limits the tracking ability. In turn, Chapter 5 also presents data management strategies to restore convergence and tracking abilities by removing erroneous or outdated history stack data.

After solving the individual problems of hyperplane estimation and subsystem identification, Chapter 6 is devoted to the joint estimation problem. The two proposed identification algorithms make use of the performance signals in Multiple Models Adaptive Control. These performance signals are essentially integrals over prediction errors with an exponential decay. It is proven that the performance signal in Multiple Models Adaptive Control is equivalent to a weighted vector norm of the difference between true and

estimated subsystem parameters. The first swarm-like algorithm applies the performance signals to repeatedly redistribute a set of parameter estimates in a goal-oriented way, which causes the estimated parameters to provably converge to the true parameters. Limitations are an introduced dwell-time constraint as well as the number of models needed to form the swarm, which grows exponentially with the parameter-vector dimension. For the second limitation, it was shown that the incorporation of previous knowledge, e.g. in the form of known parameters, may reduce the number of required models. Chapter 6 also presents a second algorithm, which incorporates the performance signals in parameter identifiers. Instead of updating only the currently active subsystem, a performance signal-based weighting term is used to update all subsystems in parallel depending on their ability to describe the current system behavior. This enables estimation of PWA systems with unknown switching signals. As a proof of convergence is currently not available, the conditions under which convergence is guaranteed are a topic for future work.

Closely related to the adaptive identification of PWA systems is adaptive control for PWA systems. In Chapter 7, a direct and an indirect algorithm for Model Reference Adaptive Control (MRAC) are derived. Both algorithms allow the user to specify desired closed-loop dynamics in form of a possibly switched reference model. The unknown control gains are then adaptively adjusted until the state of the controlled system tracks the state of the reference model. By choosing a linear time-invariant reference system, it is thus possible to hide the switching behavior of the unknown PWA system. In direct MRAC, the gains are directly tuned based on the tracking error. Indirect MRAC on the other side includes an intermediate step of estimating the system parameters, based on which the control gains are dynamically adjusted. In both cases, asymptotic tracking of the reference system is shown. Furthermore, convergence of all parameter errors and tracking errors to zero is guaranteed for sufficiently rich reference signals. As for parameter identifiers, the biggest limitation is the assumption of known switching signals.

Finally, Chapter 8 deals with Multiple Models Adaptive Control, where the switching between a set of candidate controllers is the solution to cope with large parameter uncertainties. A long standing problem in the field is the optimal distribution of nominal model-controller pairs, which guarantees that each system configuration in the uncertainty set is stabilized by at least one controller from the set of candidate controllers. The proposed coverage control methodology constitutes the first systematic approach to address this issue. Each nominal model and its corresponding controller are thereby responsible to stabilize all configurations from a subset of the uncertainty set. The ν-gap metric and the maximum generalized stability margin of the nominal model-controller pair enable the partitioning and allow stability considerations. Based on the introduced stability distances, an optimal partitioning of the uncertainty set for a given configuration of nominal models is obtained. The coverage functional describes the expected controller performance and is essentially an integral over the stability distances in the subsets. For the minimization of the coverage functional, two algorithms are presented. First, a gradient descent drives initially suboptimal configurations to centroidal configurations, which constitute local minima. Second, a pairwise optimization strategy yields pairwise optimal configurations at the cost of increased computational load. Finally, combining the two algorithms yields an algorithm which combines the advantage of the gradient descent, in form of reduced

computational load, with the pairwise optimal solutions obtained by the second algorithm.

In summary, given the increased complexity of engineered systems in form of switching behavior and parameter uncertainties, the ideas, approaches and algorithms established in this thesis advance the state-of-the-art in identification and control of PWA systems and thus pave the way to handle complex systems with almost linear formulations. Furthermore, with the optimal model distribution in MMAC, it is shown that switching is not only a source of complexity but may also be an enabling factor in dealing with large parameter uncertainties.

9.2 Directions for Future Research

Various future research directions emerge from this work and range from further exploitation of the presented concepts to new and exciting fields. The most promising directions are listed below:

- **Advanced hybrid observers:** The recursive algorithm for hyperplane estimation of PWA systems, which is presented in Chapter 3, enables learning of the underlying switching mechanism in hybrid systems. This switching mechanism can be used to improve the tracking performance of hybrid system observers, which are so far limited to reacting to unknown switches. By predicting switches rather than reacting to them, the observer error can be substantially reduced.
- **Parameter identifiers with performance-based weighting:** In this thesis, parameter identifiers were extended to PWA systems and proven to yield converging estimates of the unknown subsystem parameters under the assumption of known switching signals. As this constitutes a restrictive assumption in practice, an extension for unknown switching signals was proposed in form of performance based weighting. Unfortunately a formal proof of convergence is still missing for this approach. Hence, future work should focus on deriving convergence guarantees for parameter identifiers for PWA systems with unknown state-space partitions.
- **Combining indirect MRAC with concurrent learning:** The concurrent learning concept, which is applied for subsystem identification in Chapter 5, was originally developed for the task of adaptive control. As shown in [56], however, it is limited to adaptive control of systems with known input matrices. Combining the indirect MRAC structure presented in Chapter 7 with the concurrent learning-based parameter identifiers enables the application of concurrent learning also for adaptive control of systems with unknown input matrix.
- **Optimal model distribution in Gain Scheduling and LPV modeling:** Chapter 8 discusses the optimal model distribution in the context of MMAC. Similar, long-standing problems exist in other approaches which make use of multiple models. Concepts like Gain Scheduling may benefit from the robustness properties readily included in the coverage-control algorithm presented in this thesis. Furthermore, LPV modeling faces the same question of how to optimally place the linear models. Hence, future work should be directed towards extending the proposed coverage-control approach to these closely related fields.

A Background on Stability of Switched Systems and Adaptive Systems

This appendix provides necessary backgrounds on the stability of switch systems as well as basics on adaptive systems, which are needed throughout the manuscript. Important stability notions for switched and hybrid systems are summarized in Section A.1, while basics of adaptive control are subject in Section A.2.

A.1 Stability of Switched Systems

This section revises the essential stability conditions for switched systems and thus sets the foundation for various stability considerations throughout the thesis. This summary is based on the book [149], which gives an extensive introduction to switching in systems and control.

Consider the family of nonlinear functions $g_i(x)$ with the index set $\mathcal{I}$. The switching signal σ is piecewise constant and takes values from this set, i.e. $\sigma(t) : \mathbb{R}^+ \to \mathcal{I}$. The switched system is then given by

$$\dot{x} = g_\sigma(x). \tag{A.1}$$

A common assumption is that σ only has a finite number of discontinuities on every bounded time interval. These discontinuities are called switches and the time instances at which σ changes its value are called switching times. The switching sequence and the switching times might depend on time t, or state x and control input u, or both.

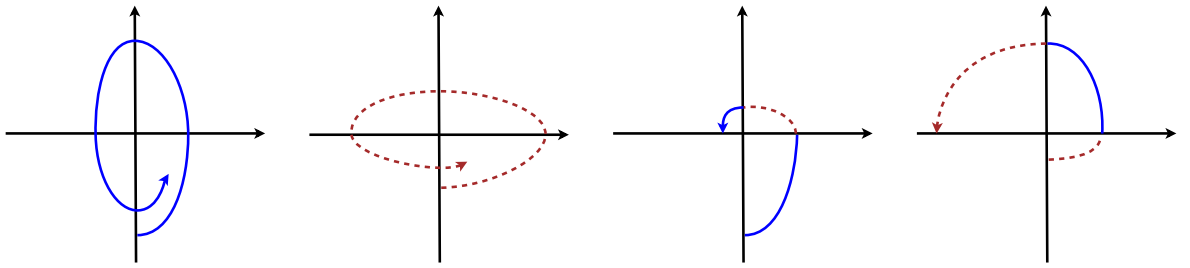
The reason why switched systems are at the same time interesting and challenging is that properties of the individual subsystems

$$\dot{x} = g_i(x), \quad i \in \mathcal{I} \tag{A.2}$$

in general do not carry over to the switched system (A.1). In other words, properties can be gained or lost as a result from switching. While the same holds for observability and controllability, the focus of this section is on stability of switched systems.

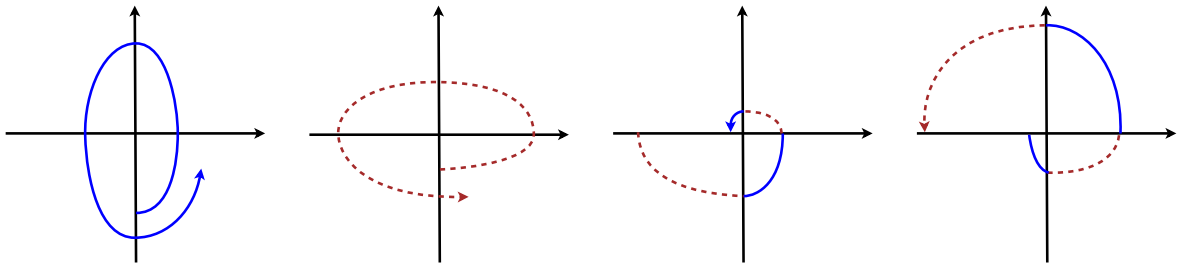
In order to stress the fact that stability of all subsystems does not necessarily imply stability of the switched system, consider the case where $\mathcal{I} = \{1, 2\}$ and $x \in \mathbb{R}^2$. Hence, there are two subsystems with trajectories evolving in the plane. First, let the two subsystems be asymptotically stable with trajectories as shown in the left two plots of Fig. A.1. The right two plots of the same figure show that the switched system can be asymptotically stable or unstable for different choices of σ. Hence, poor switching may cause instability even if all subsystems are individually stable.

Figure A.2 shows a similar effect for two unstable subsystems. As before, the switching signal determines whether the resulting switched system is asymptotically stable or unsta-



(a) stable system 1 **(b)** stable system 2 **(c)** stable switching **(d)** unstable switching

Fig. A.1: Poor switching between stable subsystems can cause instability.



(a) unstable system 1 **(b)** unstable system 2 **(c)** stable switching **(d)** unstable switching

Fig. A.2: Proper switching between unstable subsystems can result in stable trajectories.

ble. It follows that a suitably constrained switching signal may stabilize a switched system despite instability of individual subsystems.

This simple scenario motivates the need for stability tests in switched systems. In the following, Lyapunov's stability theory is extended to switched systems (Section A.1.1). With the discussed extensions, stability under arbitrary switching will be ensured by means of common Lyapunov functions in Section A.1.2. In case arbitrary switching is not feasible, the next step is to put constraints on the switching signal. Constraining the switching signal enables the use of Multiple Lyapunov functions in Section A.1.3 and dwell-time considerations in Section A.1.4.

A.1.1 Lyapunov's Stability Theory

The concept of Lyapunov stability constitutes a fundamental tool in the analysis of nonlinear systems. In this thesis, we assume knowledge of the basic Lyapunov stability concepts and point those readers who are unfamiliar with it to the standard textbook by Khalil [127]. The stability definitions for switched systems presented in [149] are straightforward modifications of the standard stability concepts in [127].

Let $x = 0$ be an equilibrium point of the switched system (A.1). Then, the system (A.1) is said to be *stable* in the sense of Lyapunov if, for each $\epsilon > 0$, there exists a $\delta = \delta(\epsilon) > 0$ such that

$$\|x(0)\| \leq \delta \quad \Rightarrow \quad \|x(t)\| \leq \epsilon \qquad \forall t \geq 0.$$

The system (A.1) is said to be *globally asymptotically stable* if it is stable and furthermore

there exists $T(\epsilon, \delta) > 0$ for arbitrary positive numbers ϵ and δ such that

$$\|x(0)\| \leq \delta \quad \Rightarrow \quad \|x(t)\| \leq \epsilon \qquad \forall t \geq T.$$

Furthermore, the system (A.1) is called *globally exponentially stable* if there exist constants $c > 0$ and $\lambda > 0$ such that the solution of (A.1) for all initial conditions satisfies the inequality

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

Next, note 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 pointed out above, the stability of switched systems heavily depends on the switching signal. This condition yields two levels of stability. First, showing *stability under arbitrary switching* amounts to finding conditions that guarantee asymptotic stability of a switched system for all possible switching signals σ. This approach becomes necessary if the switching mechanism is unconstrained, unknown, or too complicated to be modeled. A necessary condition is that all individual subsystems must be asymptotically stable. Note that stability under arbitrary switching is highly desirable as it implies that no care must be given to the switching. At the same time it is very hard to show and only few classes of systems allow stability under arbitrary switching.

If a switched system is not asymptotically stable for arbitrary switching, the next step is to characterize the switching signals under which the 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 [149] for more details on the later case. For the remainder of this work, we assume only stable subsystems and determine additional requirements on the subsystems g_i or switching signal σ that guarantee stability of the switched system.

A.1.2 Common Lyapunov Functions - Arbitrary Switching

Stability under arbitrary switching is closely related to the existence of a function that qualifies as a Lyapunov function for all subsystems (A.2). Such a Lyapunov function is called a *common Lyapunov function* for the family of subsystems. More precisely, a common Lyapunov function for (A.2) is a positive definite ($V(x) > 0, \forall x \neq 0$) and continuously differentiable function $V : \mathbb{R}^n \to \mathbb{R}$, if there exists another positive definite, continuous

function $W : \mathbb{R}^n \to \mathbb{R}$, such that

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

The asymptotic stability of a switched system for which there exists a common Lyapunov function is guaranteed by the following theorem.

Theorem A.1 (Common Lyapunov Function) "If all subsystems g_i in the family (A.2) share a radially unbounded common Lyapunov function, then the switched system (A.1) is globally uniformly asymptotically stable (GUAS)." [149, 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.

While the search for a common Lyapunov function is very difficult in the general nonlinear case, it can be reduced to a set of linear matrix inequalities (LMIs) in the case of linear switched systems. Consider therefore the switched linear system

$$\dot{x} = A_\sigma x,$$

for which it is usual to consider quadratic Lyapunov functions of the form

$$V(x) = x^\top P x.$$

In order to qualify as a *common quadratic Lyapunov function*, the matrix $P \in \mathbb{R}^{n \times n}$ must be symmetric, positive definite ($P \succ 0$) and satisfy

$$A_i^\top P + P A_i \prec -Q \quad , \forall i \in \mathcal{I} \tag{A.3}$$

for a symmetric, positive definite matrix Q. Note that (A.3) is a set of LMIs for which there exist efficient solvers such as the Matlab toolbox YALMIP [154, 155]. The ability to obtain such simple LMI-conditions is also the central motivation for quadratic common Lyapunov functions in switched linear systems.

Besides the discussed derivation of a common Lyapunov function, there exist other approaches to proof stability under arbitrary switching. One such approach is to analyze the commutation relations among the subsystems. Another way is to exploit special structures in the system such as Hurwitz subsystems in triangular form.

A.1.3 Multiple Lyapunov Functions - Constrained Switching

As pointed out before, it might not be possible to prove asymptotic stability of a switched system under arbitrary switching. If a switched system is not asymptotically stable under arbitrary switching, then there does not exist a common Lyapunov function. In that case, one can proceed by constraining the switching signal and work with *multiple Lyapunov functions*.

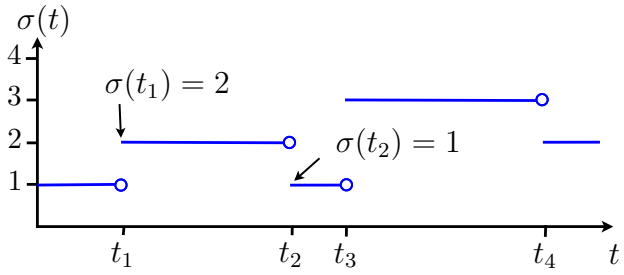


Fig. A.3: Sketch of the switching signal notation.

Suppose that all systems $\dot{x} = g_i(x)$ are globally asymptotically stable and an individual Lyapunov function V_i is known for each subsystem $i \in \mathcal{I}$. The goal is then to prove stability for a constrained switching signal σ based on the switched Lyapunov function V_σ. For the corresponding analysis a more precise notation for σ is needed. Let us denote with $t_j,\ j = 1, 2, \ldots$ the switching times at which the switching signal is discontinuous. Between two consecutive switching times the switching signal is constant, i.e. $\sigma(t) = \sigma(t_j)$ for the time interval $t_j \leq t < t_{j+1}$. Figure A.3 sketches this notation.

It is possible that the values of the Lyapunov functions V_i coincide at all switching times t_j, i.e. $V_{\sigma(t_{j-1})}(t_j) = V_{\sigma(t_j)}(t_j)$. In that case, V_σ is a continuous Lyapunov function for the switched systems. The stability of the switched system directly follows from such a continuous Lyapunov function. Figure A.4a 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. It is more common that V_σ is discontinuous at switching times. This is usually due to an antagonistic behavior of the individual Lyapunov functions V_i. Often, 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 $\sigma(t) \neq j$. Figure A.4b gives an example for such discontinuous Lyapunov functions.

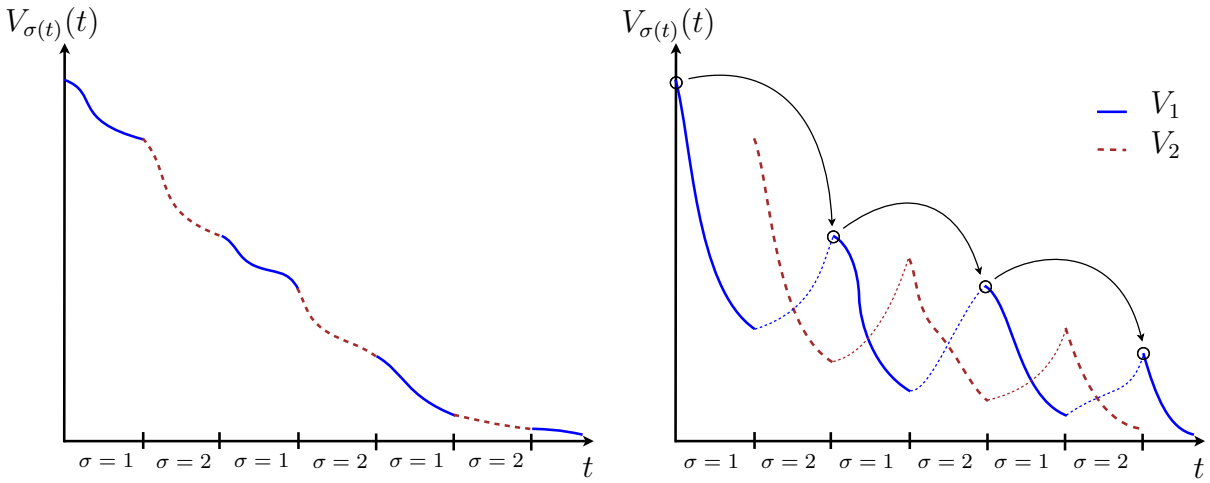
Despite the possible increase of V_σ at every switch, multiple Lyapunov functions still enable a proof of asymptotic stability. The idea is to analyze the values of each Lyapunov function V_i individually at the beginning of each interval in which the corresponding subsystem is active, i.e. in which $\sigma(t) = i$. 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 A.2 (Multiple Lyapunov Functions) "Let (A.2) 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 $(t_j, t_l), j < l$ such that $\sigma(t_j) = \sigma(t_l) = i \in \mathcal{I}$ and $\sigma(t_k) \neq i$ for $t_j < t_k < t_l$, we have

$$V_i(x(t_l)) - V_i(x(t_j) \leq -W_i(x(t_j)).$$

Then the switched system (A.1) is globally asymptotically stable." [149, Th. 3.1]

The continuous case of multiple Lyapunov functions was attributed to state-dependent



(a) Continuous Lyapunov function **(b)** Discontinuous Lyapunov function

Fig. A.4: Multiple Lyapunov functions V_i form a switched Lyapunov function V_σ.

switching above. The discontinuous case covered by Theorem A.2 on the other side is common in *time-dependent switching*. With the knowledge of multiple Lyapunov functions it is thus possible derive constraints on the switching signal in the time domain.

A.1.4 Dwell-Time Assumptions

Assuming that all subsystems are asymptotically stable, it is easy to agree that instability of the switched system is primarily caused by frequent switching. If the system would switch sufficiently slow, the transient effects would have enough time to dissipate. Conditions for when a system switches sufficiently slow can be formulated based on the notion of multiple Lyapunov functions derived above.

Slow switching is commonly defined by a minimum time interval that the switched system has to reside in each mode. Therefore, one defines a positive time $\tau_d > 0$ called the *dwell time* and restricts the admissible switching signals such that each subsystem remains active for at least τ_d. It follows that two consecutive switching times t_j and t_{j+1} must always satisfy the inequality $t_{j+1} - t_j \geq \tau_d$.

We can calculate a lower bound on τ_d from the decay of the individual subsystems. Assume that the Lyapunov functions V_i of all subsystems are known and there exist positive constants a_i, b_i and c_i with

$$a_i\|x\|^2 \leq V_i(x) \leq b_i\|x\|^2$$
$$\frac{\partial V_i(s)}{\partial x} g_i(x) \leq -c_i\|x\|^2.$$

Then, a lower bound for τ_d is given in terms of these constants. For just two subsystems for instance, the lower bound on the dwell time is

$$\tau_d > \left(\frac{c_1}{b_1} + \frac{c_2}{b_2}\right)^{-1} \log \frac{b_1 b_2}{a_1 a_2}.$$

Note that specifying a fixed dwell time might be too restrictive in practice. In such

cases, a relaxed version of the dwell-time assumption is required. The motivation behind relaxed dwell-time assumptions is to switch fast when necessary and then compensate by sufficiently slow switching when possible. By this compromise, the dwell-time assumption is only satisfied on average. Let $N_\sigma(T,t)$ be the number of switches of σ on the time interval $[t,T]$. The switching signal σ is then said to have an *average dwell time* τ_a if there exist $N_0 > 0$ and $\tau_a > 0$ such that:

$$N_\sigma(T,t) \leq N_0 + \frac{T-t}{\tau_a} \quad ,\forall T \geq t \geq 0.$$

With the definitions, theorems and assumptions revised in this section, the presented introduction to stability of switched systems remains concise but sufficiently detailed for the reader to follow the considerations in this thesis.

A.2 Preliminaries for Adaptive Systems

As the focus of this thesis is on the adaptive identification and control of switched systems, a few preliminaries from the adaptive control literature are required. This section repeats important definitions and frequently applied lemmas from [114] which are needed for various convergence proofs throughout the thesis. Besides [114], the interested reader is referred to [6, 144, 175, 240] for alternative introductions to the field of adaptive control.

Adaptive control emerged as a framework to deal with dynamical systems whose parameters are uncertain or time varying. One way to handle parameter uncertainties in control loops is through the robust control methodology, where it is assumed that a constant controller exists and provides good performance for all possible parameter configuration. As a matter of course the performance of a robust controller deteriorates with increasing uncertainty. For very large uncertainty there might not even exist a single, constant controller to stabilize the system with the required performance. This is where adaptive controllers come into play.

In adaptive control, the formerly constant control gains are replaced by time-varying gains initialized with some estimates. Adaptive mechanisms then adjust these gains in order to match the unknown parameters of the system. The resulting closed-loop system is highly nonlinear due to the adaptation. Early update mechanisms relied on some clever heuristics that worked well most of the times. Under some conditions, however, they led to serious accidents such as the crash of NASA's X-15-3 in 1967. Accidents like that shattered the trust in adaptive control which took many years to regain. Lyapunov's stability theory enabled stability proofs for adaptive control algorithms and helped to mature the field during the last half century. With the general awareness for stability today, it is possible to revise the X-15 crash and prevent similar events in the future [78].

Adaptive algorithms adjust the time-varying control gains in such a way that the controlled system responds to reference signals in a specified way. Note however that in most algorithms the control gains do not actually converge to the ideal parameters. That means adaptation is only present until the estimated gains deliver the desired system output. What sounds like a problem is actually none as long as the system does not get excited in all its modes by the reference signal.

A.2.1 Persistent Excitation

For some applications, such as parameter estimation, an incomplete convergence is unsatisfactory. A lot of research effort went into the definition of conditions under which adaptive update laws converge to the true parameters [37, 174, 269]. Today it is well known that the system must be excited in all its continuous modes for all times in order to ensure convergence. The adaptive control community first introduced a qualitative notion for *persistent excitation* (PE) to describe input signals that excite all modes of the plant in such a way that perfect identification is possible. The first formal definition goes back to Åström *et al.* in 1966 [5]. The convergence results presented in this thesis are based on the following definition.

Definition A.3 (Persistent Excitation) "A piecewise continuous signal vector $z : \mathbb{R}^+ \to \mathbb{R}^n$ is persistently excited in $\mathbb{R}^n$ with a level of excitation $\alpha_0 > 0$ if there exist constants $\alpha_1, T_0 > 0$ such that

$$\alpha_1 I_n \geq \frac{1}{T_0} \int_t^{t+T_0} z(\tau) z^\top(\tau) d\tau \geq \alpha_0 I_n, \tag{A.4}$$

$\forall t \geq 0$." [114, Def. 4.3.1]

The next natural question is how to ensure PE in an adaptive system? For linear controllable systems (which we assume), it can be shown that sufficiently many frequencies in the input signal yield sufficiently many frequencies in the state of the system and therefore satisfies (A.4). More precisely, a controllable system is PE if the control inputs or reference signals are *sufficiently rich* (of suitable order) with the following definition.

Definition A.4 (Sufficiently rich signals) "A stationary signal $u : \mathbb{R}^+ \to \mathbb{R}$ is called sufficiently rich of order n, if the support of the spectral measure $S_u(\omega)$ of u contains at least n points." [114, Def. 5.2.3]

In later chapters it will be utilized that a signal u is sufficiently rich of order $2n$ if it contains at least n sinusoidal signals. Furthermore, the following lemma will be used to ensure PE in Lemma 4.4 and Lemma 7.9.

Lemma A.5 (PE and autocovariance) "If the autocovariance of a function $z : \mathbb{R}^+ \to \mathbb{R}^n$ defined as

$$R_z(t) := \lim_{T \to \infty} \frac{1}{T} \int_{t_0}^{t_0+T} z(\tau) z^\top(t+\tau) d\tau$$

exists and is uniform with respect to t_0, then z is persistently exciting if and only if $R_z(0)$ is positive definite." [114, Lem. 5.6.1]

Note that PE is a very complex property and we attempted to introduce here only its main features needed in subsequent chapters. For a more elaborate discussion, refer to [175, Chapter 6]. Recently, a first definition for PE in switched systems was given by Petrecky and Bako [199], where besides the continuous modes also the discrete modes need to be excited in a persistent way.

A.2.2 Convergence and Stability

Above it was pointed out that adaptive systems introduce time-varying estimates of control gains or system parameters, and apply update laws to drive these estimates to their true values. The estimates can now be interpreted as additional states of the closed-loop system. The following lemma describes the convergence of a system augmented in this way, which frequently arises in adaptive systems.

Lemma A.6 (Convergence in adaptive systems) "Consider the system described by

$$\begin{bmatrix}\dot{\xi}_1\\ \dot{\xi}_2\end{bmatrix} = \begin{bmatrix}\mathcal{A} & \Psi^\top(t)\\ -P_1\Psi(t)P_2 & 0\end{bmatrix}\begin{bmatrix}\xi_1\\ \xi_2\end{bmatrix} \tag{A.5}$$

where $\xi_1 \in \mathbb{R}^n$, $\xi_2 \in \mathbb{R}^{n_z n}$ for some integer $n_z, n \geq 1$, and $\mathcal{A}, P_1, P_2$ are constant matrices and $\Psi(t)$ is of the form

$$\Psi(t) = \begin{bmatrix} z_1(t)I_n \\ z_2(t)I_n \\ \vdots \\ z_{n_z}(t)I_n \end{bmatrix} \in \mathbb{R}^{n_z n \times n},$$

where $z_i, i = 1, 2, \ldots, n_z$ are the elements of the vector $z \in \mathbb{R}^{n_z}$. Suppose that z is persistently excited and there exists a matrix $P_0 > 0$ such that $\dot{P}_0 + A_0^\top P_0 + P_0 A_0 + C_0 C_0^\top \leq 0$, where

$$A_0 = \begin{bmatrix}\mathcal{A} & \Psi^\top(t)\\ -P_1\Psi(t)P_2 & 0\end{bmatrix}, \quad C_0^\top = [I_n, 0]\,.$$

Then, the equilibrium $\xi_{1e} = 0$, $\xi_{2e} = 0$ of (A.5) is exponentially stable in the large." [114, Lem. 5.6.3]

In subsequent chapters, we will encounter various adaptive systems of the form (A.5). In those instances ξ_1 is either the tracking error (in an adaptive control setting) or the prediction error (in an adaptive identification setting). The vector ξ_2 will usually describe the current error in the estimated parameters (controller or system parameters). Lemma A.6 therefore serves as a helpful tool to show global (equivalent to in the large) convergence of all errors to zero. This at the same time corresponds to a successful estimation of the true parameters.

In the considered settings, the auxiliary vector z in Lemma A.6 consists of states and control inputs or reference signals. In that context, the following lemma proves itself useful to verify the PE condition on z.

Lemma A.7 (PE of internal signals – linear systems) Consider a linear subsystem $\dot{x} = Ax + Bu$ with state $x \in \mathbb{R}^n$ and controllable pair (A, B). If the input signals in u are sufficiently rich of order $n + 1$ with distinct frequencies, then the vector $z = [x^\top, u^\top]^\top$ is persistently excited.

The proof of Lemma A.7 is given in Appendix D in order to highlight the challenge in extending parameter identifiers from linear to affine systems. More precisely, the affine input vector corresponds to an additional input signal that does not satisfy the assumption of being sufficiently rich. The derivation of suitable countermeasures in Section 4.2.3 is one of the contributions of this work.

The above definitions and lemmas can characterize the amount of information contained in signals required for convergence. For adaptive control, additional definitions may be needed to analyze input-output properties. For this purpose it is conventional to use $\mathcal{L}^p$ spaces [175, Sec. 2.7]. Let the $\mathcal{L}^2$ norm of a signal vector $z : \mathbb{R}^+ \to \mathbb{R}^n$ be

$$\|z\|_2 := \left(\int_0^\infty |z(\tau)|^2 d\tau \right)^{1/2}$$

where $|\cdot|$ denotes any norm in $\mathbb{R}^n$. Furthermore, let the $\mathcal{L}^\infty$ norm be

$$\|z\|_\infty := \sup_{t \geq 0} |z(t)|.$$

We say $z \in \mathcal{L}^2$ or $z \in \mathcal{L}^\infty$ if $\|x\|_2$ or $\|x\|_\infty$ exists (i.e. is finite), respectively. As $z \in \mathcal{L}^\infty$ implies boundedness of $z(t)$ and vice versa, we use the two statements synonymously.

This concludes the required preliminaries for adaptive systems.

B Existing Identification Algorithms for PWA Systems

This appendix provides a short description of the identification algorithms for switched or hybrid systems listed in table 2.1. The list extends previous surveys on the matter [96, 121, 193] with more recent results. Table 2.1 categorizes existing algorithms into four types. Optimization-based methods seek a relaxation of the mixed-integer optimization problem or find an alternative formulation. Algebraic methods are based on the idea that multiple ARX models can be identified (independent of the switching sequence) as a single, lifted ARX model. Clustering techniques are applied at different stages of the identification process in clustering-based methods. Recursive methods estimate the subsystem parameters in a recursive way.

B.1 Identification of SARX Systems

B.1.1 Algebraic methods

Original algebraic method [250, 251]: All algebraic methods for the identification of SARX models originate from the Vidal's early idea in [251] and its comprehensive discussion in [250]. The key idea in algebraic methods is to decouple the identification of the model parameters from the inference of the discrete mode of the system. This can be achieved by viewing the set of ARX models as a single, lifted dynamical model with increased complexity. The lifted model is obtained through polynomial embedding of the input-output data. The parameters of the lifted model can be identified independent of the switching mechanism. In [250] it is further exploited that the model is linear in the so-called hybrid model parameters, which allows to apply standard recursive identifiers to estimated the lifted model with the embedded input-output data. Finally, the parameters of the ARX submodels are obtained from the lifted model through the derivatives of the polynomial at different regressors.

Multi-output systems [17]: The algebraic approach was originally developed for single-output systems with an unknown number of modes and unknown, possibly different orders. Extending the algebraic approach to multi-output SARX systems poses a few challenges. Instead of finding the coefficients of one vanishing polynomial in the single-output case, multiple polynomials are needed in the multiple-output case. This causes an exponential growth in the number of parameters. In [17], the authors propose a projection of the input-output data onto a low-dimensional linear subspace to cope with this problem.

Total Least Squares [181]: This work also build upon the algebraic method by Vidal [251], which provides the exact solution to the identification task only in the noise-free case. In order to improve the performance for noisy measurements, an improved estimation

algorithm for the highest order of the subsystem as well as a Total Least Squares-based identification of the subsystems is proposed. The obtained closed form solution, however, is restricted to bi-modal systems.

Matrix-wise [182]: Nazari *et al.* carry on with their motivation from [181] to reduce the sensitivity to noise in algebraic methods through the use of matrix differential calculus and the Kronecker product. This approach is termed Matrix-Wise. The obtained nonlinear algebraic equation for calculating the paremters of the SARX model cannot be solved analytically. Hence, a solution is found numerically by combining the Newton iterative method and particle swarm optimization. A proof of convergence remains an open issue. Further discussions on the approaches described in [181] and [182] can be found in Nazari's Doctoral Thesis [180].

B.1.2 Optimization-based methods

Continuous optimization [140, 141]: These works propose a framework in which the mixed-integer optimization in (2.13) is avoided. The original combinatorial optimization problem, is recast into the minimization of a product of loss functions, i.e. a reformulation as a continuous, nonlinear and non-convex optimization problem. In the obtained optimization problem, only the subsystem parameters appear as unknown variables. The complexity scales linearly with the available number of data points and standard solvers can be applied to find the optimal solution. Furthermore, the proposed framework is argued to contain the algebraic approach [251] and the support vector regression [138, 139] as special cases.

Semi-definite optimization [186, 187]: This paper builds upon the algebraic method [251], which is known to find the optimal solution only in the noise-free case. For noisy measurements, however, the algebraic procedure results in a challenging, non-convex polynomial optimization problem. The contribution of [186, 187] is a relaxation of the algebraic method to the rank minimization of a matrix. The relaxed problem is shown to be a semi-definite optimization problem that can be solved efficiently.

Sparse optimization [188, 189]: Based on some a priori information about the system, the idea in [188] is to identify a suitable set of ARX models that explains the available input-output data and at the same time optimizes a performance index in terms of minimum number of switches or subsystems. This behavior is incorporated by analyzing the difference between two consecutive parameter vectors. The sparsity of the obtained vector sequence must be maximized to achieve fewer switches. Although this maximization problem is known to be NP-hard, the paper shows that a solution can be found efficiently by employing recent results on sparse signal recovery. In [189], the same ideas are extended by additional remarks about the identifiability of switches. Furthermore, the greedy algorithm in [188] is executed on the available data in two directions: forward and backward. Also, it is noted that the set membership definition of the measurement noise plays a crucial role. It is shown that the exact number of switches can be found in case the noise is specified in terms of its ℓ_∞-norm. Finally, the difference between identifying the minimum number of switches and identifying the minimum number subsystems is highlighted and the consequences are discussed.

Sparse optimization [10]: Also Bako formulated the identification of SARX systems as a sparsification problem, an idea inspired by recent developments in the compressed sensing community. The combinatorial ℓ_0 optimization problem (2.13) is relaxed into a convex ℓ_1-norm minimization problem. The paper provides sufficient conditions under which this relaxation is exact.

Sum-of-norms regularization [185]: The work presented in [185] is concerned with the field of segmentation, i.e. segmenting time-varying systems into ARX models with piecewise constant parameters. The proposed solution to this problem is formulated as a least-squares problem with sum-of-norms regularization. Prediction errors are penalized by the least-squares terms, while too frequent switching is penalized by the sum-of-norms regularization. The authors do not claim to outperform other approaches but highlight the simplicity of the proposed algorithm. Its only tuning parameter is the regularization constant which is used to trade-off model fit against the number of switches. The discussed principles were later applied in the estimation PWARX models [183].

Particle swarm optimization [165]: Maruta *et al.* also recognize that the identification of SARX systems according to the optimization in (2.13) is ill-conditioned and non-convex and thus gives rise to many local minima. The authors propose a particle swarm-based optimization algorithm in which a swarm of particles is distributed in the parameter space. Each particle represents an ARX model and is updated iteratively. It is shown that standard particle swarm optimization algorithms are prone to converge to local minima. Hence, the authors propose a distributed particle swarm optimization in which particles are updated only based on information from a subset of neighboring particles. While a formal proof of convergence is not available, simulation results indicate the applicability of the approach.

Subspace clustering [11]: Although the work presented in [11] does not directly address the problem of SARX system identification, it is nevertheless closely related. Unlabeled data from a finite number of linear subspaces are clustered by formulating a sparse optimization problem. In the context of SARX systems, the subspaces correspond to the ARX subsystems.

B.1.3 Clustering-based methods

Bounded-switching clustering [223]: Similarly to the optimization-based algorithms, the approach in [223] converts the identification task into a binary integer optimization problem with least square. The available input-output data is optimally divided into clusters that can be associated with parameters which are piecewise constant in time. The unknown number and order of the ARX subsystems as well as the number of switches are automatically selected in [223] with the help of the Akaike Information Criterion.

B.1.4 Recursive methods

Non-supervised classification [9]: The main goal in [9] is to estimate the number of subsystem. This goal is achieved in two steps. First, the statistical features, which are needed to discriminate the subsystems, are determined. Then, a non-supervised classification method, called Fuzzy Pattern Matching, is proposed to determine online the number

of subsystems as well as their parameters.

Two-step [12]: The recursive identification algorithm proposed by Bako *et al.* in [12] alternates between two steps: data assignment to submodels and parameter update. The algorithm begins with an initial set of parameter vectors. At each time instance, the system output is predicted for each parameter vector. The input-output data is assigned to the model/ parameter vector that best predicted the actual system output. After this assignment step, the input-output data is applied in a recursive least square (RLS) scheme to update the parameter vector of the corresponding subsystem. It should be noted that a detailed analysis of convergence is not available. Furthermore, the algorithm does not converge for poorly chosen initial parameter vectors. A similar algorithm was later proposed in [104].

Mode mismatch reset [257, 258]: RLS algorithms, such as Bako's two-step approach [12], are the starting point for work presented in [258]. The authors point out that errors during the mode detection keep spreading and deteriorate the identification performance. After an analysis of the influences of errors and their origin, a reset strategy is proposed that avoids spreading of the error and in turn improves convergence.

Recursive algebraic [250]: As pointed out earlier, a recursive version of the originally bath-type algebraic method by Vidal [251] is presented in [250].

Multiple models [81, 82]: An algorithm inspired by the Multiple Models Adaptive Control concept [176, 177, 179] was proposed by Feiler and Narendra in order to deal with rapidly time varying systems. The parameters of all subsystems are updated simultaneously with each measurement through a projection algorithm. Each subsystems update is weighted by a factor which depends on the subsystem's ability to explain the current measurement. While simulation studies indicate convergence under sufficient excitation, a formal proof of converges could not be established.

B.1.5 Other methods

The following method does not fall into one of the above mentioned categories.

Data set partitioning [205]: Porreca and Ferrari-Trecate presented a procedure to partition data-sets into subsets characterized by the same parameter values. Their procedure is applied in order to determine which measurements have been generated by the same ARX subsystem. The procedure relies on a fine-to-coarse principle which avoids testing all possible partitions.

B.2 Identification of PWARX Systems

B.2.1 Optimization-based methods

Mixed-integer programming [210]: One of the first optimization-based approaches to the identification of PWARX models is given in [210]. The presented algorithms are based on mixed-integer linear or quadratic programming and guarantee convergence to a global optimum. Due to the computational complexity, however, this is only achievable for small problems. Hence, for the special case of rare switching, a change detection approach

is proposed to trade off optimality and complexity.

BPWARX models [263]: Wen *et al.* proposed Piecwise Affine Basis Function Autoregressive eXogenous (BPWARX) models for the identification of nonlinear black-box models. BPWARX modes are characterized by a weighted sum of PWA basis functions. A modified Gauss-Newton algorithm is investigated to reconstruct a model from input-output data.

PrARX models [236, 237]: A Probability weighted ARX (PrARX) model consists of multiple ARX models which are weighted by the probabilistic function 'softmax'. The authors formulate a single optimization problem in terms of subsystem parameters and probabilities, which is solved by a steepest descent algorithm. The identified PrARX model is then easily transformed into a PWARX model by analyzing the estimated probability weighting. A detailed discussion of convergence guarantees or local minima is missing.

Expectation-maximization [118]: Another approach is to solve the identification problem within the EM (expectation-maximization) algorithm framework. Jin *et al.* propose a revised EM algorithm which utilizes contaminated Gaussian distributions for the construction of the objective function. An interesting solution is proposed for the classification of un-decidable data points. The final step of region estimation is carried out with a modified MRLP algorithm in which the data points are weighted according to their probability of being an outlier.

Weighted least-squares [131–134]: Lai *et al.* propose a different formulation of the optimization problem in identifying PWA systems. In the formulation in (2.13), each data point is assigned to one subsystem through the indicator function, which leads to a mixed integer optimization problem. In the proposed algorithm, this assignment is replaced by the geometrical mean or harmonic mean of the squared errors at each time instance. The estimated subsystem parameters are then updated by weighted least-squares using multiple models, i.e. according to their ability to predict the current measurement. While the simulation results show convergence under persistent excitation and sufficiently frequent switching, there exists no formal proof of convergence.

Sum-of-norms regularization [183, 184]: The non-convex identification problem is approximated by a least-squares problem on the fit of the data and a sum-of-norms regularization term. The regularization depends on the difference between two consecutive parameter vectors and serves as a trade off between fit and complexity (number of models). The algorithm is initialized with a different parameter vector for each measurement. Since this corresponds to an over-parametrization, similar parameter vectors are merged if this does not increase the error too much.

Nonlinear hybrid systems [145, 246]: These works deals with switched systems, in which the subsystems are nonlinear. The algorithms build on previous works on sparse learning of PWARX models. Iteratively, the parameters of one model after another are estimated by maximizing the sparsity in the obtained error.

Compression with ℓ_1 optimization [162–164]: The work by Maruta *et al.* considers non-parametric, data-based representations of PWA systems. The proposed algorithms compress the data, which is required for this mapping, with ℓ_1 optimization techniques. This enables the application of the proposed method to large data sets. The trade-off between complexity and accuracy of the model is easily adjusted by a single parameter.

The revisited version in [162] provides a novel derivation of the algorithm which shows that under certain conditions, the algorithm computes the PWA function that passes through the training data and that has the least amount of broken hinges.

B.2.2 Clustering-based methods

K-means [88]: One of the early algorithms to exploit clustering, linear identification, and pattern recognition techniques is defined in [88]. The input-output data is divided into local data sets for which local parameter vectors are identified. A feature space is created from the local data sets and the estimated local parameter vectors. A modified K-means algorithm is applied for clustering in the feature space and the subsystem parameters are then reconstructed from the feature space clusters.

Statistical clustering [173]: Nakada *et al.* propose a statistical clustering method which is based on a Gaussian mixture model. Once the data is clustered, the classification is achieved through SVMs and the subsystem parameters are retrieved by least squares estimation.

Greedy approach [29]: The algorithm proposed by Bemporad *et al.* partitions an infeasible system of linear inequalities into a minimum number of feasible subsystems in a greedy way. The greedy approach provides an initial clustering, which is repeatedly updated by a refinement procedure in order to improve data classification and parameter estimation.

k-plane clustering [235]: This algorithm identifies PWARX models in three steps. First, a modified version of the k-plane clustering algorithm [39] provides an initial classification of available input-output data and the corresponding subsystem parameter estimates. In the second step, the refinement algorithm discussed in [29] improves both data classification and parameter estimation. The final step is the common SVM-based reconstruction of the regressor partitions.

Competitive learning [97]: In this competitive learning-inspired approach, the submodels are obtained by means of an algorithm which rewards subsystems that best fit the data in each region of the regression space. Then, fuzzy clustering is used to choose representatives from the original data set.

Dempster-Shafer theory [36]: The fact that the data obtained from PWARX systems is locally linear is exploited in [36] through an evidential procedure, which is based on Dempster-Shafer theory. It simultaneously minimizes the error between measured output and each subsystem's output on the one hand side and the distance between data belonging to the same subsystem on the other side.

Split and merge clustering [23]: After the available input-output data is divided into subsets, a new split and merge clustering algorithm is used to estimate the correct number of subsystems as well as the subsystem parameters. The benefit of the proposed algorithm is that there is only a single tuning parameter and that the algorithm is insensitive to the initialization step.

Correlation clustering [116]: A very nice overview of different clustering algorithms for arbitrary data is given in [116]. The authors employ correlation clustering algorithms to identify a PWARX model of a combustion engine.

Density-based spatial clustering of applications with noise (DBSCAN) [137]: The DBSCAN algorithm is used for clustering, which is motivated by the fact that the number of models is automatically determined. Furthermore, the algorithm is claimed to be robust against noisy data and outliers.

B.2.3 Recursive methods

Recursive two-step [12]: We discussed the two-step method by Bako *et al.* already as a recursive method for the identification of SARX systems. As shown in [12], the algorithm can be easily extended to PWARX systems by adjusting the data assignment criterion. While data assignment in the SARX case depends only on the prediction of each subsystem, the spacial information is added in the PWARX case. This is achieved by estimating and updating the center of each polytope in the regressor space. As for the SARX case, this heuristic fails for poorly initialized estimates.

Recursive k-plane clustering [234]: The work presented in [235] is extended to a recursive formulation. The submodel parameters are estimated via recursive k-plane clustering and the regions in the regressor space are estimated by incremental proximal support vector machines.

Piecewise Affine Regression via Recursive Multiple Least Squares [27, 42]: The algorithm proposed by Bemporad *et al.* is similar to the recursive two-step algorithm by Bako. New regressor vectors are sequentially assigned by clustering to one of the subsystems and are subsequently applied to update both the corresponding subsystem parameters and the centroid. The approach relies on a computationally efficient algorithm to solve recursive least squares problems with inverse QR decomposition.

B.2.4 Other methods

Bayesian approach [122]: In this work, the unknown parameters are considered as random variables characterized by their probability density functions. The identification problem thus transforms to computing the posterior probability density function of the model parameters, which is subsequently relaxed to a particle filtering method.

Bounded error [30, 233]: In [30], Bemporad *et al.* define the identification of PWARX systems in terms of the MIN PFS problem, which stands for partition into a minimum number of feasible subsystems. That means, given a bound of the desired error, the algorithm finds a minimum number of regions and corresponding parameter vectors that satisfy the bounded error. Such a bound is a useful tuning parameter to trade off between quality of fit and model complexity. The work in [233] follows the same idea and replaces the greedy randomized relaxation algorithm in [30] by a greedy clustering-based method.

Adaptive hinging hyperplanes [266]: A model of adaptive hinging hyperplanes (AHH) is developed in [266] and combines the model of multivariate adaptive regression splines (MARS) and generalized hinging hyperplanes (GHH) such that the positive aspects of both are retained. The universal approximation ability of the AHH model is proved and an identification procedure is derived.

B.3 Identification of State-Space Systems

B.3.1 Optimization-based methods

Mixed integer-free optimization [172]: Two approaches for the identification of discrete-time PWA systems in state-space form are presented in [172]. Starting from a mixed-integer optimization problem, similar to the one formulated in (2.13) for SARX systems, the paper discusses two relaxations that circumvent the mixed integer optimization but still yield the same optimal solution as the original problem. The first approach results in a smooth constrained optimization problem and the second approach yields a non smooth unconstrained optimization problem. The alternative formulations, however, introduce various local minimal due to wrongly assigned data. A graph-based method is proposed to reduce the risk of converging to these local minima.

Expectation maximization [34, 99]: In [34], the authors show that standard expectation maximization is not suitable for hybrid systems due to the combination of discrete mode and continuous states. Hence, an approximate expectation maximization (EM) approach is proposed, where the approximation is incorporated by tracking a sequence of discrete modes in the expectation step. The convergence to local maxima constitutes the main limitation of the approximate EM algorithm. Therefore, work in [99] provides an extension of [34] which avoids being trapped in local maxima. This is achieved by learning Gaussian representations of the local maxima. The search for optimal initial conditions can thus be focused on areas away from these local optima and hence convergence to more accurate models is provided.

Sparse optimization [16]: This paper follows the sparse optimization approach presented in [10] for the identification of SARX models. The approach is extended to the identification of discrete time, switched linear state-space systems in which the full continuous state is assumed to be measurable. Since the sparsification is computationally intractable, it is relaxed to a convex optimization problem. The paper provides some conditions under which the recovered solution obtained with the relaxed optimization is exact.

Convex formulation [80]: A convex relaxation of the identification problem in (2.13) for state-space systems is considered in [80]. Elhamifar *et al.* propose to divide the sequence of input-output data into a large number of subsets and estimate a state-space system with common basis for each subset. By relaxing the indicator functions to continuous weights and by introducing an upper bound on the maximum number of activated subsystems, one obtains a constrained but convex optimization problem. One drawback of this approach is that the estimated parameters are not part of the optimization problem anymore.

Guarded Transitions [220]: This paper presents an algorithm on unsupervised learning of guarded Probabilistic Hybrid Automata (PHA) models. The central difference to PWA state-space systems are potential state jumps.

B.3.2 Clustering-based methods

Change detection [198]: The central element of this algorithm is the estimation of switching times by change detection. The obtained switching times enable the estimation

of subsystem parameters by standard subspace identification techniques. Since the combination of change detection and subspace identification provides very similar local models, a merging algorithm is added in order to combine similar models. The two steps of parameter estimation and merging are performed repeatedly until the number of models and subsystem parameters converge.

Online clustering [197]: This method estimates the Markov parameters matrix of the linear submodels through online clustering.

Switching detection [35]: This paper constitutes an alternative approach to the estimation of switching signals. The stream of input-output data is divided in smaller batches and a projected subspace of the input-output data is obtained for each batch. In this setting, rank variations correspond to switches of the system.

Clustering with a priori knowledge [171]: This work considers the same optimization problem as in [172] but proposes a clustering-based approach to find an approximate solution. The five steps of the algorithm are the analysis of a priori knowledge, the identification of the input net, the scaling and clustering of measured data, the triangulation of the obtained lattice, and finally the optimization. The proposed algorithm allows to incorporate previous knowledge about the system in form of known subsystem dynamics or known switching hyperplanes at various steps in the algorithm. The paper concludes with a validation of the algorithm on a hybrid two tank systems.

Subset clustering [156]: This paper – similar to ideas presented in [193] – combines subspace identification and robust clustering techniques for the identification of switched linear state-space systems. The proposed method first divides the original data into smaller data sets. Afterwards, a different subsystem is identified for each set through standard subspace identification techniques. Then, a robust clustering algorithm merges the large number of subsystems. In a final iteration, the original data set is divided again and measurements are assigned to the obtained parameter clusters. Finally, applying the standard subspace identification methods on these data sets delivers more precise subsystems. The major drawback of this algorithm is the required dwell time. Only for slowly switching systems does the initial data division lead to distinct clusters in the parameter space.

B.3.3 Recursive methods

On-line two-step clustering [196]: This algorithm consist of two stages: estimation and classification. First, the subsystem parameters are estimated with least squares based on measurements taken from a sliding window. In the second stage, a dynamic clustering algorithm classifies the parameters.

On-line switching detection [13, 14]: These algorithms build upon recursive parameter estimation, on-line switching time detection and on-line order estimation. It is noted that most subspace identification algorithms are based on Singular Value Decomposition (SVD). The application of SVD is, however, computationally prohibitive in recursive applications. Instead, the authors propose a structured subspace identification strategy, also known as propagator method, which can be implemented on-line. Besides estimating subsystem parameters, the propagator method allows to decide at runtime whether a new subsystem is detected. New subsystems are characterized by an increased estimated system order due to inconsistent data from two subsystems. Also it is proposed that the Euclidean

distance can be used to reason about the similarity to previously estimated subsystems. This avoids creation of additional models for each occurrence of the same subsystem. In order to achieve good performance with the proposed algorithms, the system must reside in each discrete mode for a certain time, hence the switching signal must satisfy a minimum dwell-time.

Continuous state estimation [15]: An identification algorithm for switched linear state-space systems without minimum dwell-time assumptions is given in [15]. At the same time, the authors propose a solution to circumvent one of the dominant problems in the identification of state-space systems, which is the unknown, continuous state of the system. Based on some observability assumptions, the state-space model is converted into an input-output realization with an increased number of subsystems. The algorithm then alternates between data classification and parameter update and provides estimates of the input-output models with recursive least squares. In the final step of the algorithm, the original state-space models are recovered from the input-output models. One problem of this algorithm is the randomized initialization, which leads to non deterministic results. Hence, satisfactory results might only be obtained after several trials.

Sparse optimization [50]: The presented ideas in this paper make use of recent developments in the field of sparse optimization. It is shown that a multi-variable model, which is subject to a sparse error sequence can be recursively identified. In the next step, this result is applied to identify jump linear Markov systems, i.e. switched linear state-space systems in which the discrete mode evolution is given in terms of a probabilistic transition matrix. In this setting, the sparse errors are caused by erroneous data assignment.

Bimodal and Trimodal PWA Systems [242]: In this work, the continuous bimodal or trimodal PWA systems are represented in max-form and based on a suitable cost function, a recursive Gauss-Newton algorithm yields adaptive laws to estimate both the subsystem matrices and centers online. The fact that these centers are used to describe the state-space partitions limits the algorithm to PWA systems with only two or three subsystems.

B.3.4 Other methods

Subspace-identification [249]: Verdult *et al.* discusses the identification of discrete-time PWL systems under the assumption of known switching. The submodels are identified using subspace identification methods, which requires the switching to be slower than the block size used to build the Hankel matrices. The obtained models are equal to the true models up to a linear state transformation. In a final step, they must therefore be transformed to the same state basis.

C Assumptions for Hybrid Observer in Region Estimation

The hybrid switching observer in Chapter 3 is tied to three assumptions, which were originally presented in [146]. For completeness and readability of the main text, they are summarized here.

A general premise for mode detection in switched systems is related to the following assumption on joint observability.

Assumption C.1 (Joint Observability) "The joint observability Gramian

$$W_{i,j}(t) := \int_0^t \begin{bmatrix} \phi_i^\top(\tau) \\ \phi_j^\top(\tau) \end{bmatrix} \begin{bmatrix} \phi_i(\tau) & \phi_j(\tau) \end{bmatrix} d\tau,$$

of two subsystems i and j with $\phi_i(t) := C_i e^{A_i t}$ is non-singular for any $t \neq 0$ and $i \neq j$." [146, Ass. 1]

Let $\omega_{\min}(t) = \min_{i \neq j} \lambda_{\min}(W_{i,j}(t))$ be the minimum eigenvalue of the observability Gramian for all subsystem combinations and a time period t. Furthermore, introduce the approximations $\check{\mu}, \hat{\mu} \geq 1$ and $\check{\lambda}, \hat{\lambda} \geq 0$ such that $\|e^{A_i t}\| \leq \check{\mu} e^{\check{\lambda} t}$ and $\|e^{(A_i - L_i C_i)t}\| \leq \hat{\mu} e^{-\hat{\lambda} t}$, $\forall t \geq 0, \forall i \in \mathcal{N}$. Finally, let $L_{\max} = \max_{i \in \mathcal{N}} \|L_i\|$ and $C_{\max} = \max_{i \in \mathcal{N}} \|C_i\|$. Then the intervals T_δ and T_Δ can be related to the input and noise of the PWA system by the following assumption.

Assumption C.2 (Definitions T_δ, T_Δ, $\mathcal{X}_\triangle$, $\mathcal{S}_\triangle$) "The input u and the measurement noise v_y are uniformly bounded, i.e., $\|u(t)\| \leq u_{\max}$, $\|v_y(t)\| \leq v_{y,\max}, \forall t \geq 0$. Moreover, there exist positive constants T_δ and T_Δ, such that

$$\omega_{\min}(T_\delta)\|x(t)\|^2 > \left(u_{\max}\sqrt{N_u(T_\delta)} + 2v_{y,\max}\sqrt{T_\delta}\right)^2,$$
$$\omega_{\min}(T_\Delta)\|x(t)\|^2 > \frac{1}{2}\left(u_{\max}\sqrt{N_u(T_\Delta)} + 2v_{y,\max}\sqrt{T_\Delta} + 2v_{y,\max}C_{\max}(\check{\mu}e^{\check{\lambda}T_\Delta} + 1)^2 \hat{E}(T_\delta)\sqrt{T_\Delta}\right)^2,$$

for all $t \geq 0$, where $h_i(t) := C_i e^{A_i t} B_i = \phi_i(t) B_i$ and $U_i(t) := \int_0^t \phi_i^\top(\tau)\phi_i(\tau) d\tau$ and

$$N_u(T_\delta) := \max_{i,j \in \mathcal{N}, i \neq j} \int_0^{T_\delta} \left(\int_0^\tau \|h_i(s) - h_j(s)\| ds \right)^2 d\tau,$$
$$M_{\max}(T_\delta) := \max_{i \in \mathcal{N}} M_i(T_\delta) := \max_{i \in \mathcal{N}} \int_0^{T_\delta} \|U_i^{-1}(T_\delta)\phi_i^\top(\tau)\| d\tau,$$
$$\hat{E}(T_\delta) := \max\left\{ M_{\max}(T_\delta), \frac{L_{\max}}{\hat{\lambda}} \right\}." \text{ [146, Ass. 2]}$$

The threshold $\mathcal{N}_u(T_\delta)$ resembles a maximum distance between any two subsystem parameterizations. In [275], we have investigated various different approaches to quantify distances between dynamical systems. Finally define the following thresholds

$$\mathcal{X}_\triangle := v_{y,\max}(\check{\mu}e^{\check{\lambda}T_\Delta} + 1)\hat{E}(T_\delta),$$
$$\mathcal{S}_\triangle := v_{y,\max}^2 \left(1 + C_{\max}(\check{\mu}e^{\check{\lambda}T_\Delta} + 2)\hat{E}(T_\delta)\right)^2 T_\Delta.$$

For the dwell time of the PWA system, the following assumption arises.

Assumption C.3 (Dwell Time) "The switching signal σ has a dwell time of $T_\Delta + T_\delta$ and there is no switch in the initial time interval $[0, T_\delta]$." [146, Ass. 3]

D Proofs related to Persistent Excitation

Throughout this thesis, it was found at various occasions that parameter convergence requires an internal signal vector z to be persistently excited (PE). For the linear case, one could resort to the standard result in form of Lemma A.7 given in the preliminaries Section A.2. As the same does not hold for affine systems, Lemma 4.4 was proposed in Section 4.2.3 and enabled the application of parameter identifiers for PWA systems. A similar extension is given by Lemma 7.9 in Section 7.3 for parameter convergence in indirect MRAC for PWA systems. In the following, the somewhat technical proofs of these lemmas are given.

D.1 Proof of Lemma A.7 – PE linear systems

Proof: The proof of Lemma A.7 presented here constitutes a modification of [114, pp.306-308] and is essentially based on Lemma A.5 in Section A.2 and a frequency domain relationship between the internal signal vector $z = [x^\top, u^\top]^\top$ and the inputs u.

First, define the transfer function $H_k(s)$ which relates z to the k-th input u_k in the frequency domain. Overall, the Laplace transform $Z(s)$ of the state $z(t)$ relates to the Laplace transform $U(s)$ of the inputs $u(t)$ by

$$Z(s) = \sum_{k=1}^{p} \underbrace{\begin{bmatrix} (sI_n - A)^{-1} b_k \\ \ell_k \end{bmatrix}}_{H_k(s)} U_k(s), \tag{D.1}$$

where b_k denotes the k-th column of B and $\ell_k \in \mathbb{R}^p$ denotes the k-th column of the identity matrix $I_p \in \mathbb{R}^{p\times p}$, i.e.:

$$\ell_k = [\, 0 \quad \cdots \quad 0 \quad \underbrace{1}_{k\text{-th element}} \quad 0 \quad \cdots \quad 0]^\top.$$

Under the assumption of distinct input frequencies, the auto covariance of z is given by

$$R_{z1}(0) = \frac{1}{2\pi} \sum_{k=1}^{p} \int_{-\infty}^{\infty} H_k(-j\omega) S_{u_k}(\omega) H_k^\top(j\omega) d\omega, \tag{D.2}$$

where $S_{u_k}(\omega)$ is the spectral distribution of the input u_k. Since u_k is sufficiently rich of order $n+1$, its spectrum has $n+1$ distinct peaks $\mathcal{F}_{u_k}(\omega_{kl})$ at frequencies ω_{kl}:

$$S_{u_k}(\omega) = \sum_{l=1}^{n+1} \mathcal{F}_{u_k}(\omega_{kl}) \delta(\omega - \omega_{kl}), \tag{D.3}$$

where $\delta(\omega) = 1$ for $\omega = 0$ and $\delta(\omega) = 0$ otherwise. With (D.3), the integrals in (D.2) are replaced by summations:

$$R_{z1}(0) = \frac{1}{2\pi} \sum_{k=1}^{p} \sum_{l=1}^{n+1} \mathcal{F}_{u_k}(\omega_{kl}) H_k(-j\omega_{kl}) H_k^\top(j\omega_{kl}). \tag{D.4}$$

According to Lemma A.5, the vector z is persistently exciting if (D.4) is positive definite, i.e., the quadratic equation

$$\chi^\top R_{z1}(0)\chi = 0, \quad \chi \in \mathbb{R}^{n+p} \tag{D.5}$$

can only have a single solution at $\chi = 0_{n+p}$. Note that the outer products $H_k(-j\omega_{kl})H_k^\top(j\omega_{kl})$ of the column vectors $H_k(j\omega_{kl})$ are positive semi-definite. Therefore, each summand in (D.4) is positive semi-definite. As all the summands of $R_{z1}(0)$ are positive semi-definite, equality holds if and only if

$$\chi^\top H_k(-j\omega_{kl}) H_k^\top(j\omega_{kl})\chi = 0 \qquad \begin{matrix} k = 1, 2, \ldots, p, \\ l = 1, 2, \ldots, n+1, \end{matrix}$$

or equivalently

$$H_k^\top(j\omega_{kl})\chi = 0 \qquad \begin{matrix} k = 1, 2, \ldots, p, \\ l = 1, 2, \ldots, n+1. \end{matrix} \tag{D.6}$$

The transfer function $H_k(s)$ can also be written as

$$H_k(s) = \frac{1}{a(s)} \begin{bmatrix} \operatorname{adj}(sI_n - A)b_k \\ a(s)\ell_k \end{bmatrix} = \frac{1}{a(s)} \bar{H}_k(s), \tag{D.7}$$

where $a(s) = \det(sI_n - A)$. Therefore, the equalities (D.6) are also equivalent to

$$\bar{H}_k^\top(j\omega_{kl})\chi = 0 \qquad \begin{matrix} k = 1, 2, \ldots, p, \\ l = 1, 2, \ldots, n+1. \end{matrix} \tag{D.8}$$

Each element of $\bar{H}_k(j\omega_{kl})$ is a polynomial in ω_{kl} of maximal order n (see equation (D.7)). So, after multiplying with χ, $g_k(s) := \bar{H}_k^\top(s)\chi$ is also a polynomial in s of maximal order n. For a given k, equation (D.8) states that $g_k(s)$ is zero at $n+1$ frequencies or values of s. As $g_k(s)$ is only of order n, it follows that $g_k(s) = 0,\ \forall s \in \mathbb{C},\ k = 1, \ldots, p$. Combining the p equalities of (D.8) in matrix form gives

$$\begin{bmatrix} \operatorname{adj}(sI_n - A)B \\ a(s)I_p \end{bmatrix}^\top \chi = 0_p, \tag{D.9}$$

where $0_p \in \mathbb{R}^p$ is a p-dimensional zero vector. Next, the vector χ is expressed as $\chi = [\mathcal{X}^\top, \mathcal{Y}^\top]^\top$, with $\mathcal{X} = [\chi_1, \ldots, \chi_n]^\top \in \mathbb{R}^n$ and $\mathcal{Y} = [\chi_{n+1}, \ldots, \chi_{n+p}]^\top \in \mathbb{R}^p$. This leads to

$$\left(\operatorname{adj}(sI_n - A)B\right)^\top \mathcal{X} + a(s)\mathcal{Y} = 0_p. \tag{D.10}$$

The following expressions for $\text{adj}(sI_n - A)B$ and $a(s)$ are applied to equation (D.10)

$$\begin{aligned} \text{adj}(sI - A)B =& Bs^{n-1} + (AB + a_{n-1}B)s^{n-2} + \\ & (A^2B + a_{n-1}AB + a_{n-2}B)s^{n-3} + \ldots + \\ & (A^{n-1}B + a_{n-1}A^{n-2}B + \ldots + a_1 B), \\ a(s) =& s^n + a_{n-1}s^{n-1} + \ldots + a_1 s + a_0, \end{aligned} \tag{D.11}$$

which results in a polynomial in s of order n. All coefficients of the resulting polynomial must be zero, so that the polynomial is zero independent of s. This leads to n equalities that can be written in a compact form:

$$\mathcal{Y} = 0_p \quad \text{and} \quad [B, AB, \ldots, A^{n-1}B]^\top \mathcal{X} = 0_{np}.$$

With (A, B) controllable, it follows that the matrix $[B, AB, \ldots, A^{n-1}B]$ is of full rank. Therefore, the equality (D.10) holds only for $\mathcal{X} = 0_n$ and $\mathcal{Y} = 0_p$. Consequently, $R_{z1}(0)$ is indeed positive definite and according to Lemma A.5 z is PE, which completes the proof of Lemma A.7. □

D.2 Proof of Lemma 4.4 – PE affine systems

Proof: The proof of Lemma 4.4 is very similar to the one of Lemma A.7 above, with some additional complications caused by the affine input vector.

As highlighted in (4.22), the vector f can be seen as an additional column of the input matrix associated to an additional constant input 1. Therefore, let z be related to the constant input via a transfer functions $H_f(s)$. Furthermore, let the transfer function $H_k(s)$ relate z to the k-th input u_k. Overall, z is related to the inputs by

$$Z(s) = \underbrace{\begin{bmatrix} (sI_n - A)^{-1} f \\ \ell_{p+1} \end{bmatrix}}_{H_f(s)} 1(s) + \sum_{k=1}^{p} \underbrace{\begin{bmatrix} (sI_n - A)^{-1} b_k \\ \ell_k \end{bmatrix}}_{H_k(s)} U_k(s), \tag{D.12}$$

where b_k denotes the k-th column of B, and $\ell_k,\ \ell_{p+1} \in \mathbb{R}^{p+1}$ denote the k-th and $(p+1)$-th column of the identity matrix $I_{p+1} \in \mathbb{R}^{(p+1)\times(p+1)}$, i.e.:

$$\begin{aligned} \ell_k &= [\, 0 \quad \cdots \quad 0 \quad \underbrace{1}_{k\text{-th element}} \quad 0 \quad \cdots \quad 0 \quad 0]^\top, \\ \ell_{p+1} &= [\, 0 \quad \cdots \quad 0 \quad 0 \quad 0 \quad \cdots \quad 0 \quad 1]^\top. \end{aligned}$$

Under the assumption of distinct input frequencies, the auto-covariance of z is given by

$$R_z(0) = \frac{1}{2\pi} \int_{-\infty}^{\infty} H_f(-j\omega) S_1(\omega) H_f^\top(j\omega) d\omega + \frac{1}{2\pi} \sum_{k=1}^{p} \int_{-\infty}^{\infty} H_k(-j\omega) S_{u_k}(\omega) H_k^\top(j\omega) d\omega \tag{D.13}$$

where $S_1(\omega)$ and $S_{u_k}(\omega)$ are the spectral distributions of the constant input 1 and the

inputs u_k, respectively. Since u_k is sufficiently rich of order $n+1$, its spectrum has $n+1$ distinct peaks $\mathcal{F}_{u_k}(\omega_{kl})$ at frequencies ω_{kl}:

$$S_{u_k}(\omega) = \sum_{l=1}^{n+1} \mathcal{F}_{u_k}(\omega_{kl})\delta(\omega - \omega_{kl}), \tag{D.14}$$

where $\delta(\omega) = 1$ for $\omega = 0$ and $\delta(\omega) = 0$ otherwise. The affine term f results in a spectral peak at zero:

$$S_1(\omega) = \delta(\omega). \tag{D.15}$$

With (D.14) and (D.15), the integrals in (D.13) are replaced by summations:

$$R_z(0) = \frac{1}{2\pi} H_f(-j0) H_f^\top(j0) + \frac{1}{2\pi} \sum_{k=1}^{p} \sum_{l=1}^{n+1} \mathcal{F}_{u_k}(\omega_{kl}) H_k(-j\omega_{kl}) H_k^\top(j\omega_{kl}). \tag{D.16}$$

According to Lemma A.5, the vector z is persistently excited if (D.16) is positive definite, i.e., the quadratic equation

$$\chi^\top R_z(0)\chi = 0, \quad \chi \in \mathbb{R}^{n+p+1} \tag{D.17}$$

can only have a single solution at $\chi = 0_{n+p+1}$. Be aware that the dimension of χ increased by one as compared to the linear case in the previous proof (compare (D.5)). Now, note that the outer products $H_k(-j\omega_{kl})H_k^\top(j\omega_{kl})$ of the column vectors $H_k(j\omega_{kl})$ are positive semi-definite. Therefore, each summand in (D.16) is positive semi-definite and (D.17) is equivalent to

$$\chi^\top R_z(0)\chi = \underbrace{\chi^\top R_{z1}(0)\chi}_{\geq 0} + \underbrace{\chi^\top R_{z2}(0)\chi}_{\geq 0} = 0, \tag{D.18}$$

where

$$R_{z1}(0) := \frac{1}{2\pi} \sum_{k=1}^{p} \sum_{l=1}^{n+1} \mathcal{F}_{u_k}(\omega_{kl}) H_k(-j\omega_{kl}) H_k^\top(j\omega_{kl}), \qquad R_{z2}(0) := \frac{1}{2\pi} H_f(-j0) H_f^\top(j0).$$

Here, one needs to be cautious if one were to believe that positive definiteness of $R_{z1}(0)$ follows due to the same arguments as in the proof of Lemma A.7 above. Note that the additional input associated with f increases the dimension of $R_z(0)$ by one. While $\ell \in \mathbb{R}^{n+p}$ in case of a linear system, we have $\ell \in \mathbb{R}^{n+p+1}$ for affine systems (compare (D.1) and (D.12)).

Following the same arguments as before (compare (D.9)), rewrite the equality $\chi^\top R_{z1}(0)\chi = 0$ as

$$\begin{bmatrix} \operatorname{adj}(sI_n - A)B \\ a(s)I_p \\ 0_p^\top \end{bmatrix}^\top \chi = 0_p, \tag{D.19}$$

where $0_p \in \mathbb{R}^p$ is a p-dimensional zero vector. Next, the vector χ is expressed as $\chi = [\mathcal{X}^\top, \mathcal{Y}^\top, \mathcal{Z}]^\top$, with $\mathcal{X} = [\chi_1, \ldots, \chi_n]^\top \in \mathbb{R}^n$, $\mathcal{Y} = [\chi_{n+1}, \ldots, \chi_{n+p}]^\top \in \mathbb{R}^p$ and $\mathcal{Z} = \chi_{n+p+1} \in \mathbb{R}^1$. This leads to

$$(\mathrm{adj}(sI_n - A)B)^\top \mathcal{X} + a(s)\mathcal{Y} + 0_p\mathcal{Z} = 0_p. \tag{D.20}$$

Applying the expressions for $\mathrm{adj}(sI_n - A)B$ and $a(s)$ in (D.11) to equation (D.20) results in a polynomial in s of order n. All coefficients of the resulting polynomial must be zero, so that the polynomial is zero independent of s. This leads to n equalities that can be written in a compact form:

$$\mathcal{Y} = 0_p \quad \text{and} \quad [B, AB, \ldots, A^{n-1}B]^\top \mathcal{X} = 0_{np}.$$

With (A, B) controllable, it follows that the matrix $[B, AB, \ldots, A^{n-1}B]$ is of full rank. Therefore, the equality (D.20) holds for $\mathcal{X} = 0_n$, $\mathcal{Y} = 0_p$ and an arbitrary $\mathcal{Z} \in \mathbb{R}$. Consequently, $R_{z1}(0)$ is only positive semi-definite. Therefore, $R_{z2}(0)$ in (D.18) must restrict the choice of $\mathcal{Z}$ further, i.e. $\chi^\top R_{z2}(0)\chi = 0$ for $\mathcal{X} = 0_n$, $\mathcal{Y} = 0_p$ and $\mathcal{Z} = 0$. Applying the same arguments as above to the term with $R_{z2}(0)$ in (D.18) leads to

$$\bar{H}_f^\top(j0)\chi = \begin{bmatrix} \mathrm{adj}(-A)f \\ 0_p \\ a(j0) \end{bmatrix}^\top \chi = 0,$$

and with $\chi = [\mathcal{X}^\top, \mathcal{Y}^\top, \mathcal{Z}]^\top$ results in

$$(\mathrm{adj}(-A)f)^\top \mathcal{X} + a(j0)\mathcal{Z} = 0. \tag{D.21}$$

Since $R_{z1}(0)$ and $R_{z2}(0)$ are both positive semi-definite, one needs to find those χ for which $\chi^\top R_{z1}(0)\chi = 0$ and $\chi^\top R_{z2}(0)\chi = 0$. In order to ensure $\chi^\top R_{z1}(0)\chi = 0$, it is thus necessary to assume $\mathcal{X} = 0_n$ and $\mathcal{Y} = 0_p$ in (D.21). With $a(j0) = a_0$ equation (D.21) reduces to $a_0 Z = 0$. Since A is assumed to be invertible, we have $a_0 \neq 0$. Therefore, $\mathcal{Z} \overset{!}{=} 0$ in order to ensure $H_f^\top(j0)\chi = 0$ with $\mathcal{X} = 0_n$ and $\mathcal{Y} = 0_p$. This shows that $\chi^\top R_z(0)\chi = 0$ only holds for $\chi = 0_{n+p+1}$ and thus proves positive definiteness of $R_z(0)$ and according to Lemma A.5 makes z persistently exciting, which completes the proof of Lemma 4.4. □

D.3 Proof of Lemma 7.9 – PE in indirect MRAC

The following proof is copied from the journal article [280].

Proof: The proof of Lemma 7.9 follows ideas presented in [114, pp.306–308] and relates the PE condition of z_m to the spectral properties of r. Consider the two representations

$$\dot{x}_\mathrm{m} = A_\mathrm{m}x_\mathrm{m} + B_\mathrm{m}r_\mathrm{m} + f_\mathrm{m} \qquad \text{and} \qquad \begin{aligned} \dot{x}_\mathrm{m} &= Ax_\mathrm{m} + Bu_\mathrm{m} + f \\ u_\mathrm{m} &= K_x^* x_\mathrm{m} + K_r^* r + k_f^*. \end{aligned}$$

Note that the affine terms f_m and k_f^* can be understood to relate x_m and u_m to a constant input of 1. Therefore, let $z_\mathrm{m} = [x_\mathrm{m}^\top, u_\mathrm{m}^\top, 1]^\top$ be related to the constant input via the transfer function $H_f(s)$. Furthermore, let the transfer function $H_k(s)$ relate z_m to the k-th element of the reference signal vector $r = [r_1, \ldots, r_p]^\top$. Hence, $z_\mathrm{m}(s)$ is given by

$$z_\mathrm{m}(s) = \begin{bmatrix} x_\mathrm{m}(s) \\ u_\mathrm{m}(s) \\ 1(s) \end{bmatrix} = \underbrace{\begin{bmatrix} (sI - A_\mathrm{m})^{-1} f_\mathrm{m} \\ k_f^* \\ 1 \end{bmatrix}}_{=:H_f(s)} 1(s) + \sum_{k=1}^{p} \underbrace{\begin{bmatrix} (sI - A_\mathrm{m})^{-1} b_{\mathrm{m},k} \\ k_{r,k}^* + K_x^* (sI - A_\mathrm{m})^{-1} b_{\mathrm{m},k} \\ 0 \end{bmatrix}}_{=:H_k(s)} r_k(s), \tag{D.22}$$

where I is the identity matrix, and $b_{\mathrm{m},k}$ and $k_{r,k}^*$ denote the k-th column of B_m and K_r^*, respectively.

Under the assumption of distinct frequencies in the reference signals, the auto covariance of z_m is given by

$$R_{z_\mathrm{m}}(0) = \frac{1}{2\pi} \int_{-\infty}^{\infty} H_f(-j\omega) S_1(\omega) H_f^\top(j\omega) d\omega + \frac{1}{2\pi} \sum_{k=1}^{p} \int_{-\infty}^{\infty} H_k(-j\omega) S_{r_k}(\omega) H_k^\top(j\omega) d\omega, \tag{D.23}$$

where $S_1(\omega)$ and $S_{r_k}(\omega)$ are the spectral distributions of the constant input 1 and the reference signals r_k, respectively. Since r_k is sufficiently rich of order $n+1$, its spectrum has $n+1$ distinct peaks $\mathcal{F}_{r_k}(\omega_{kl})$ at frequencies $\omega_{kl}, l = 1, \ldots n+1$. The constant input $1(s)$ results in a single spectral peak at zero. The spectral distributions are

$$S_1(\omega) = \delta(\omega), \quad S_{r_k}(\omega) = \sum_{l=1}^{n+1} \mathcal{F}_{r_k}(\omega_{kl}) \delta(\omega - \omega_{kl}), \tag{D.24}$$

where $\delta(\omega = 0) = 1$ and $\delta(\omega \neq 0) = 0$. With (D.24), the integrals in (D.23) are replaced by summations:

$$R_{z_\mathrm{m}}(0) = \frac{1}{2\pi} H_f(-j0) H_f^\top(j0) + \frac{1}{2\pi} \sum_{k=1}^{p} \sum_{l=1}^{n+1} \mathcal{F}_{r_k}(\omega_{kl}) H_k(-j\omega_{kl}) H_k^\top(j\omega_{kl}). \tag{D.25}$$

According to [114, Lem. 5.6.1], the vector signal z_m is PE if $R_{z_\mathrm{m}}(0)$ is positive definite, i.e. the quadratic equation

$$\chi^\top R_{z_\mathrm{m}}(0) \chi = 0, \quad \chi \in \mathbb{R}^{n+p+1} \tag{D.26}$$

can only have a single solution at $\chi = 0_{n+p+1}$. Note that the outer products $H_k(-j\omega_{kl}) H_k(j\omega_{kl})^\top$ of the column vectors $H_k(j\omega_{kl})$ are positive semidefinite. Therefore, each summand in (D.25) is positive semidefinite and (D.26) is equivalent to

$$\chi^\top R_{z_\mathrm{m}}(0) \chi = \underbrace{\chi^\top R_{z1}(0) \chi}_{\geq 0} + \underbrace{\chi^\top R_{z2}(0) \chi}_{\geq 0} = 0, \tag{D.27}$$

where

$$R_{z1}(0) := \frac{1}{2\pi}\sum_{k=1}^{p}\sum_{l=1}^{n+1}\mathcal{F}_{r_k}(\omega_{kl})H_k(-j\omega_{kl})H_k^T(j\omega_{kl}), \qquad R_{z2}(0) := \frac{1}{2\pi}H_f(-j0)H_f^T(j0).$$

First, $\chi^\top R_{z1}(0)\chi = 0$ is analyzed. Since all summands in $R_{z1}(0)$ are positive semidefinite, the equality holds, if and only if

$$\chi^\top H_k(-j\omega_{kl})H_k^\top(j\omega_{kl})\chi = 0 \qquad \begin{matrix}\forall k = 1,\dots,p,\\ \forall l = 1,\dots,n+1,\end{matrix}$$

or equivalently

$$H_k^\top(j\omega_{kl})\chi = 0 \qquad \begin{matrix}\forall k = 1,\dots,p,\\ \forall l = 1,\dots,n+1.\end{matrix} \tag{D.28}$$

Next, rewrite the transfer function $H_k(s)$ with $a(s) = \det(sI - A_\mathrm{m})$ as

$$H_k(s) = \frac{1}{a(s)}\begin{bmatrix}\mathrm{adj}(sI - A_\mathrm{m})b_{\mathrm{m},k}\\ a(s)k_{r,k}^* + K_x^*\mathrm{adj}(sI - A_\mathrm{m})b_{\mathrm{m},k}\\ 0\end{bmatrix} = \frac{1}{a(s)}\bar{H}_k(s), \tag{D.29}$$

and note that the equalities (D.28) are also equivalent to

$$\bar{H}_k^\top(j\omega_{kl})\chi = 0 \qquad \begin{matrix}\forall k = 1,\dots,p,\\ \forall l = 1,\dots,n+1.\end{matrix} \tag{D.30}$$

From (D.29), it can be seen that each element of the vector $\bar{H}_k(j\omega_{kl})$ is a polynomial in ω_{kl} of maximal order n. Hence, multiplying with χ makes $g_k(s) := \bar{H}_k^\top(s)\chi$ also a polynomial in s of maximal order n. Now note that the requirement $\forall l$ in (D.30) implies that $g_k(s)$ vanishes at all $n+1$ frequencies ω_{kl} or corresponding values of s. As, however, $g_k(s)$ is only of order n, it follows that $g_k(s) = 0,\ \forall s \in \mathbb{C},\ \forall k$. Combining the p equalities obtained for $k = 1,\dots,p$ in matrix form yields

$$\begin{bmatrix}\mathrm{adj}(sI - A_\mathrm{m})B_\mathrm{m}\\ a(s)K_r^* + K_x^*\mathrm{adj}(sI - A_\mathrm{m})B_\mathrm{m}\\ 0_p^\top\end{bmatrix}^\top \chi = 0_p, \tag{D.31}$$

where $0_p \in \mathbb{R}^p$ is a zero vector. Next, express the vector χ as $\chi = [\mathcal{X}^\top, \mathcal{Y}^\top, \mathcal{Z}]^\top$, with $\mathcal{X} = [\chi_1,\dots,\chi_n]^\top \in \mathbb{R}^n$, $\mathcal{Y} = [\chi_{n+1},\dots,\chi_{n+p}]^\top \in \mathbb{R}^p$ and $\mathcal{Z} = \chi_{n+p+1} \in \mathbb{R}$, which leads to

$$\left(\mathrm{adj}(sI - A_\mathrm{m})B_\mathrm{m}\right)^\top \mathcal{X} + \left(a(s)K_r^* + K_x^*\mathrm{adj}(sI - A_\mathrm{m})B_\mathrm{m}\right)^\top \mathcal{Y} + 0_p\mathcal{Z} = 0_p. \tag{D.32}$$

The following expressions for $\mathrm{adj}(sI - A)B$ and $a(s)$ are applied to equation (D.32)

$$\begin{aligned}\mathrm{adj}(sI - A)B = {} & Bs^{n-1} + (AB + a_{n-1}B)s^{n-2} + (A^2B + a_{n-1}AB + a_{n-2}B)s^{n-3}\\ & + \ldots + (A^{n-1}B + a_{n-1}A^{n-2}B + \ldots + a_1B),\end{aligned}$$

$$a(s) = s^n + a_{n-1}s^{n-1} + \ldots + a_1 s + a_0,$$

which results in a polynomial of order n. All coefficients of the resulting polynomial must be zero in order for the polynomial to be zero independent of s. For the coefficient of s^n, one obtains

$$\mathcal{Y}^\top K_r^* = 0, \tag{D.33}$$

which by multiplication of SM_s from the right yields $\mathcal{Y} = 0_p$. Note that the assumption of full column ranks in B and B_m ensures that both S and M_s have empty null spaces such that their multiplication does not affect the solution of (D.33).

For $\mathcal{Y} = 0_p$, the n equalities related to the coefficients from s^0 to s^{n-1} can be combined in the compact form

$$\begin{bmatrix} B_\mathrm{m} & A_\mathrm{m}B_\mathrm{m} & \ldots & A_\mathrm{m}^{n-1}B_\mathrm{m} \end{bmatrix}^\top \mathcal{X} = 0_{np},$$

which due to controllable $(A_\mathrm{m}, B_\mathrm{m})$ leads to $\mathcal{X} = 0_n$.

This shows that (D.32) holds for $\mathcal{X} = 0_n$, $\mathcal{Y} = 0_p$ and an arbitrary $\mathcal{Z} \in \mathbb{R}$. Consequently, $R_{z1}(0)$ is only positive semidefinite. Therefore, $R_{z2}(0)$ in (D.27) must restrict the choice of $\mathcal{Z}$ further. In other words, for $\mathcal{X} = 0_n$ and $\mathcal{Y} = 0_p$ the equality $\chi^\top R_{z2}(0)\chi = 0$ must only hold for $\mathcal{Z} = 0$. The same arguments as above lead to

$$\bar{H}_f^\top(j0)\chi = \begin{bmatrix} \mathrm{adj}(-A_\mathrm{m})f_\mathrm{m} \\ a(j0)k_f^* \\ a(j0) \end{bmatrix}^\top \chi = 0,$$

which with $\chi = [0_p^\top, 0_n^\top, \mathcal{Z}]^\top$ and $a(j0) = a_0$ implies $a_0\mathcal{Z} = 0$. Since A_m is assumed to be invertible, we have $a_0 \neq 0$. Therefore, $\mathcal{Z} \overset{!}{=} 0$ in order to ensure $\bar{H}_f^\top(j0)\chi = 0$ with $\mathcal{X} = 0_n$ and $\mathcal{Y} = 0_p$. This proves positive definiteness of $R_{z_\mathrm{m}}(0)$ and according to [114, Lem. 5.6.1] makes z_m PE. □

Bibliography

[1] K. Alexis, C. Huerzeler, and R. Siegwart. Hybrid modeling and control of a coaxial unmanned rotorcraft interacting with its environment through contact. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5417–5424, 2013.

[2] B. Anderson, T. S. Brinsmead, F. De Bruyne, J. P. Hespanha, D. Liberzon, and A. Morse. Multiple Model Adaptive Control, Part 1: Finite Controller Coverings. *International Journal of Robust and Nonlinear Control*, 10(11-12):909–929, 2000.

[3] B. Anderson, T. S. Brinsmead, D. Liberzon, and A. Morse. Multiple model adaptive control with safe switching. *International Journal of Adaptive Control and Signal Processing*, 15(5):445–470, 2001.

[4] V. Antonio, B. Michele, W. Eric, and V. Dirk. Hybrid System Modeling and Identification of Cell Biology Systems: Perspectives and Challenges. In *Proceedings of the 15th IFAC Symposium on System Identification*, pages 227–232, 2009.

[5] K.-J. Åström and T. Bohlin. Numerical Identification of Linear Dynamic Systems from Normal Operating Records. In *Theory of Self-Adaptive Control Systems*, pages 96–111.

[6] K. J. Åström and B. Wittenmark. *Adaptative control*. Addison Wesley, Reading (Mass.) and New York, 2nd edition, 1995.

[7] F. Aurenhammer. Power Diagrams: Properties, Algorithms and Applications. *SIAM Journal on Computing*, 16(1):78–96, 1987.

[8] H. Axelsson, Y. Wardi, M. Egerstedt, and E. Verriest. A Provably Convergent Algorithm for Transition-Time Optimization in Switched Systems. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 1397–1402, 2005.

[9] O. Ayad, M. Sayed Mouchweh, and P. Billaudel. Switched Hybrid Dynamic Systems identification based on pattern recognition approach. In *Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ)*, pages 1–7, 2010.

[10] L. Bako. Identification of switched linear systems via sparse optimization. *Automatica*, 47(4):668–677, 2011.

[11] L. Bako. Subspace Clustering Through Parametric Representation and Sparse Optimization. *IEEE Signal Processing Letters*, 21(3):356–360, 2014.

[12] L. Bako, K. Boukharouba, E. Duviella, and S. Lecoeuche. A recursive identification algorithm for switched linear/affine models. *Nonlinear Analysis: Hybrid Systems*, 5(2):242–253, 2011.

[13] L. Bako, G. Mercère, and S. Lecoeuche. On-line structured identification of switching systems with possibly varying orders. In *Proceedings of the European Control Conference*, pages 4065–4072, 2007.

[14] L. Bako, G. Mercère, and S. Lecoeuche. On-line structured subspace identification with application to switched linear systems. *International Journal of Control*, 82(8):1496–1515, 2009.

[15] L. Bako, G. Mercère, R. Vidal, and S. Lecoeuche. Identification of switched linear state space models without minimum dwell time. In *Proceedings of the 15th IFAC Symposium on System Identification*, pages 569–574, 2009.

[16] L. Bako, L. van Luong, F. Lauer, and G. Bloch. Identification of MIMO switched state-space models. In *Proceedings of the American Control Conference (ACC)*, pages 71–76, 2013.

[17] L. Bako and R. Vidal. Algebraic Identification of MIMO SARX Models. In *Hybrid Systems: Computation and Control*, pages 43–57. Springer Berlin Heidelberg, 2008.

[18] S. Baldi, G. Battistelli, D. Mari, E. Mosca, and P. Tesi. Multiple-Model Adaptive Switching Control for Uncertain Multivariable Systems. In *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, pages 6153–6158, 2011.

[19] S. Baldi, G. Battistelli, D. Mari, E. Mosca, and P. Tesi. Multi-model unfalsified switching control of uncertain multivariable systems. *International Journal of Adaptive Control and Signal Processing*, 26(8):705–722, 2012.

[20] S. Baldi, G. Battistelli, E. Mosca, and P. Tesi. Multi-model unfalsified adaptive switching supervisory control. *Automatica*, 46(2):249–259, 2010.

[21] S. Baldi and P. Ioannou. Stability Margins in Adaptive Mixing Control via a Lyapunov-based Switching Criterion. *Automatic Control, IEEE Transactions on*, 61(5):1194–1207, 2016.

[22] A. Balluchi, L. Benvenuti, M. D. Di Benedetto, and A. Sangiovanni-Vincentelli. The design of dynamical observers for hybrid systems: Theory and application to an automotive control problem. *Automatica*, 49(4):915–925, 2013.

[23] R. S. Baptista, J. Y. Ishihara, and G. A. Borges. Split and merge algorithm for identification of Piecewise Affine systems. In *Proceedings of the American Control Conference (ACC)*, pages 2018–2023, 2011.

[24] G. Battistelli. On adaptive stabilization of mode-observable switching linear systems. In *Proceedings of the 18th IFAC World Congress*, pages 356–361, 2011.

[25] G. Battistelli, E. Mosca, M. Safonov, and P. Tesi. Stability of Unfalsified Adaptive Switching Control in Noisy Environments. *Automatic Control, IEEE Transactions on*, 55(10):2424–2429, 2010.

[26] F. Bejarano and A. Pisano. Switched Observers for Switched Linear Systems With Unknown Inputs. *Automatic Control, IEEE Transactions on*, 56(3):681–686, 2011.

[27] A. Bemporad, V. Breschi, and D. Piga. Piecewise Affine Regression via Recursive Multiple Least Squares and Multicategory Discrimination. *Automatica*, accepted for publication, 2016.

[28] A. Bemporad, G. Ferrari-Trecate, and M. Morari. Observability and controllability of piecewise affine and hybrid systems. *Automatic Control, IEEE Transactions on*, 45(10):1864–1876, 2000.

[29] A. Bemporad, A. Garulli, S. Paoletti, and A. Vicino. A Greedy Approach to Identification of Piecewise Affine Models. In *Hybrid Systems: Computation and Control*, pages 97–112, 2003.

[30] A. Bemporad, A. Garulli, S. Paoletti, and A. Vicino. A bounded-error approach to piecewise affine system identification. *Automatic Control, IEEE Transactions on*, 50(10):1567–1580, 2005.

[31] K. P. Bennett and O. L. Mangasarian. Robust linear programming discrimination of two linearly inseparable sets. *Optimization Methods and Software*, 1(1):23–34, 1992.

[32] K. P. Bennett and O. L. Mangasarian. Multicategory discrimination via linear programming. *Optimization Methods and Software*, 3(1-3):27–39, 1994.

[33] S. Bhattacharya, R. Ghrist, and V. Kumar. Multi-robot coverage and exploration on Riemannian manifolds with boundaries. *The International Journal of Robotics Research*, 33(1):113–137, 2014.

[34] L. Blackmore, S. Gil, C. Seung, and B. Williams. Model learning for switching linear systems with autonomous mode transitions. In *Proceedings of the 46th IEEE Conference on Decision and Control*, pages 4648–4655, 2007.

[35] J. Borges, V. Verdult, M. Verhaegen, and M. A. Botto. A switching detection method based on projected subspace classification. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 344–349, 2005.

[36] K. Boukharouba, L. Bako, and S. Lecoeuche. Identification of Piecewise Affine Systems Based on Dempster-Shafer Theory. In *Proceedings of the 15th IFAC Symposium on System Identification*, pages 1662–1667, 2009.

[37] S. Boyd and S. Sastry. On parameter convergence in adaptive control. *Systems & Control Letters*, 3:311–319, 1983.

[38] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge Univ. Press, Cambridge, 2004.

[39] P. Bradley and O. Mangasarian. k-Plane Clustering. *Journal of Global Optimization*, 16(1):23–32, 2000.

[40] M. S. Branicky. Multiple Lyapunov functions and other analysis tools for switched and hybrid systems. *Automatic Control, IEEE Transactions on*, 43(4):475–482, 1998.

[41] E. J. Bredensteiner and K. P. Bennett. Multicategory Classification by Support Vector Machines. In *Computational Optimization*, pages 53–79. Springer US, 1999.

[42] V. Breschi, A. Bemporad, and D. Piga. Identification of Hybrid and Linear Parameter Varying Models via Recursive Piecewise Affine Regression and Discrimination. In *Proceedings of the European Control Conference*, pages 2632–2637, 2016.

[43] D. Buchstaller and M. French. Robust Stability for Multiple Model Adaptive Control: Part I - The Framework. *Automatic Control, IEEE Transactions on*, 61(3):677–692, 2016.

[44] D. Buchstaller and M. French. Robust Stability for Multiple Model Adaptive Control: Part II - Gain Bounds. *Automatic Control, IEEE Transactions on*, 61(3):693–708, 2016.

[45] M. Buss, M. Glocker, M. Hardt, O. v. Stryk, R. Bulirsch, and G. Schmidt. Nonlinear hybrid dynamical systems: Modeling, optimal control, and applications. *Lecture Notes in Control and Information Science*, 279:311–335, 2002.

[46] G. Cauwenberghs and T. Poggio. Incremental and decremental support vector machine learning. *Advances in neural information processing systems*, 13:409–415, 2001.

[47] V. Cerone and D. Regruto. Bounding the parameters of linear systems with input backlash. *Automatic Control, IEEE Transactions on*, 52(3):531–536, 2007.

[48] S. Chaib, D. Boutat, A. Benali, and J. P. Barbot. Observability of the discrete state for dynamical piecewise hybrid systems. *Nonlinear Analysis: Theory, Methods & Applications*, 63(3):423–438, 2005.

[49] M. H. Chang and E. J. Davison. Adaptive switching control of LTI MIMO systems using a family of controllers approach. *Automatica*, 35(3):453–465, 1999.

[50] D. Chen, L. Bako, and S. Lecoeuche. A Recursive Sparse Learning Method: Application to Jump Markov Linear Systems. In *Proceedings of the 18th IFAC World Congress*, pages 3198–3203, 2011.

[51] S.-Y. Cheong and M. G. Safonov. Slow-Fast Controller Decomposition Bumpless Transfer for Adaptive Switching Control. *Automatic Control, IEEE Transactions on*, 57(3):721–726, 2012.

[52] G. Chowdhary. *Concurrent Learning for convergence in adaptive control without persistency of excitation*. PhD thesis, Georgia Institute of Technology, 2010.

[53] G. Chowdhary and E. Johnson. Concurrent learning for convergence in adaptive control without persistency of excitation. In *Proceedings of the 49th IEEE Conference on Decision and Control (CDC)*, pages 3674–3679, 2010.

[54] G. Chowdhary and E. Johnson. A singular value maximizing data recording algorithm for concurrent learning. In *Proceedings of the American Control Conference (ACC)*, pages 3547–3552, 2011.

[55] G. Chowdhary, M. Mühlegg, J. How, and F. Holzapfel. A Concurrent Learning Adaptive-Optimal Control Architecture for Nonlinear Systems. In *Proceedings of the 52nd IEEE Conference on Decision and Control*, pages 868–873, 2013.

[56] 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*, 27:280–301, 2013.

[57] L. O. Chua and A.-C. Deng. Canonical piecewise-linear representation. *Circuits and Systems, IEEE Transactions on*, 35(1):101–111, 1988.

[58] L. O. Chua, C. A. Desoer, and E. S. Kuh. *Linear and Nonlinear Circuits*. McGraw-Hill and New York, 1987.

[59] T. H. Cormen. *Introduction to algorithms*. MIT Press, Cambridge and Mass, 3rd edition, 2009.

[60] C. Cortes and C. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273, 1995.

[61] J. Cortes, S. Martinez, T. Karatas, and F. Bullo. Coverage Control for Mobile Sensing Networks. *Robotics and Automation, IEEE Transactions on*, 20(2):243–255, 2004.

[62] C. E. Davila. An efficient recursive total least squares algorithm for FIR adaptive filtering. *Signal Processing, IEEE Transactions on*, 42(2):268–280, 1994.

[63] G. De La Torre, G. Chowdhary, and E. N. Johnson. Concurrent learning adaptive control for linear switched systems. In *Proceedings of the American Control Conference (ACC)*, pages 854–859, 2013.

[64] M. Di Bernardo, C. I. Hoyos Velasco, U. Montanaro, and S. Santini. Experimental implementation and validation of a novel minimal control synthesis adaptive controller for continuous bimodal piecewise affine systems. *Control Engineering Practice*, 20:269–281, 2012.

[65] M. Di Bernardo, U. Montanaro, R. Ortega, and S. Santini. Extended hybrid model reference adaptive control of piecewise affine systems. *Nonlinear Analysis: Hybrid Systems*, 21:11–21, 2016.

[66] M. Di Bernardo, U. Montanaro, and S. Santini. Novel switched model reference adaptive control for continuous piecewise affine systems. In *Proceedings of the 47th IEEE Conference on Decision and Control*, pages 1925–1930, 2008.

[67] M. Di Bernardo, U. Montanaro, and S. Santini. Hybrid Minimal Control Synthesis identification of continuous piecewise linear systems. In *Proceedings of the 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, pages 3188–3193, 2009.

[68] M. Di Bernardo, U. Montanaro, and S. Santini. Minimal Control Synthesis Adaptive Control of Continuous Bimodal Piecewise Affine Systems. *SIAM Journal on Control and Optimization*, 48(7):4242–4261, 2010.

[69] M. Di Bernardo, U. Montanaro, and S. Santini. Minimal control synthesis adaptive control of continuous bimodal piecewise affine systems. In *51st IEEE Conference on Decision and Control*, pages 2637–2642, 2012.

[70] M. Di Bernardo, U. Montanaro, and S. Santini. Hybrid Model Reference Adaptive Control of Piecewise Affine Systems. *Automatic Control, IEEE Transactions on*, 58(2):304–316, 2013.

[71] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, 1959.

[72] X. C. Ding, Y. Wardi, and M. Egerstedt. On-Line Optimization of Switched-Mode Dynamical Systems. *Automatic Control, IEEE Transactions on*, 54(9):2266–2271, 2009.

[73] L. dos Santos Coelho and B. M. Herrera. Fuzzy Identification Based on a Chaotic Particle Swarm Optimization Approach Applied to a Nonlinear Yo-yo Motion System. *Industrial Electronics, IEEE Transactions on*, 54(6):3234–3245, 2007.

[74] Q. Du, V. Faber, and M. Gunzburger. Centroidal Voronoi Tessellations: Applications and Algorithms. *SIAM Review*, 41(4):637–676, 1999.

[75] M. Duarte and K. Narendra. Combined direct and indirect approach to adaptive control. *Automatic Control, IEEE Transactions on*, 34(10):1071–1075, 1989.

[76] M. Duarte and K. Narendra. Indirect Model Reference Adaptive Control with Dynamic Adjustment of Parameters. *International Journal of Adaptive Control and Signal Processing*, 10:603–621, 1996.

[77] J. W. Durham, R. Carli, P. Frasca, and F. Bullo. Discrete Partitioning and Coverage Control for Gossiping Robots. *Robotics and Automation, IEEE Transactions on*, 28(2):364–378, 2012.

[78] Z. Dydek, A. Annaswamy, and E. Lavretsky. Adaptive Control and the NASA X-15-3 Flight Revisited. *IEEE Control Systems Magazine*, 30(3):32–48, 2010.

[79] M. Egerstedt, Y. Wardi, and H. Axelsson. Transition-time optimization for switched-mode dynamical systems. *Automatic Control, IEEE Transactions on*, 51(1):110–115, 2006.

[80] E. Elhamifar, Burden, S., A., and S. Sastry. Adaptive Piecewise-Affine Inverse Modeling of Hybrid Dynamical Systems. In *Proceedings of the 19th IFAC World Congress*, pages 10844–10849, 2014.

[81] M. J. Feiler and K. Narendra. Simultaneous Identification and Control of Time-Varying Systems. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 1093–1098, 2006.

[82] M. J. Feiler and K. Narendra. Identification and Control Using Multiple Models. In *Proceedings of the 14th Yale Workshop on Adaptive and Learning Systems*, 2008.

[83] S. Fekri, M. Athans, and A. Pascoal. Issues, progress and new results in robust adaptive control. *International Journal of Adaptive Control and Signal Processing*, 20(10):519–579, 2006.

[84] D.-Z. Feng, X.-D. Zhang, D.-X. Chang, and W. X. Zheng. A Fast Recursive Total Least Squares Algorithm for Adaptive FIR Filtering. *Signal Processing, IEEE Transactions on*, 52(10):2729–2737, 2004.

[85] G. Feng. Stability analysis of piecewise discrete-time linear systems. *Automatic Control, IEEE Transactions on*, 47(7):1108–1112, 2002.

[86] G. Feng. Controller synthesis of fuzzy dynamic systems based on piecewise lyapunov functions. *Fuzzy Systems, IEEE Transactions on*, 11(5):605–612, 2003.

[87] G. Feng. H_infty Controller Design of Fuzzy Dynamic Systems Based on Piecewise Lyapunov Functions. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, 34(1):283–292, 2004.

[88] G. Ferrari-Trecate, M. Muselli, D. Liberati, and M. Morari. A clustering technique for the identification of piecewise affine systems. *Automatica*, 39(2):205–217, 2003.

[89] A. F. Filippov. *Differential Equations with Discontinuous Righthand Sides*, volume 18. Springer Netherlands, Dordrecht, 1988.

[90] M. Fliess, C. Join, and W. Perruquetti. Real-time estimation for switched linear systems. In *Proceedings of the 47th IEEE Conference on Decision and Control*, pages 941–946, 2008.

[91] M. Fliess, C. Join, and W. Perruquetti. Real-time estimation of the switching signal for perturbed switched linear systems. *Proceedings of the 3rd IFAC Conference on Analysis and Design of Hybrid Systems ADHS*, 2009.

[92] G. E. Forsythe, M. A. Malcolm, and C. B. Moler. *Computer methods for mathematical computations*. Prentice-Hall series in automatic computation. Prentice-Hall, Englewood Cliffs and N.J, 1977.

[93] G. Fung and O. L. Mangasarian. Incremental Support Vector Machine Classification. In *Proceedings of the Second SIAM International Conference on Data Mining*, pages 247–260. SIAM, 2002.

[94] F. Gao, S. E. Li, D. Kum, and H. Zhang. Synthesis of multiple model switching controllers using H_inf theory for systems with large uncertainties. *Neurocomputing*, 157:118–124, 2015.

[95] H. Garnier. Direct continuous-time approaches to system identification. Overview and benefits for practical applications. *European Journal of Control*, 24:50–62, 2015.

[96] A. Garulli, S. Paoletti, and A. Vicino. A Survey on Switched and Piecewise Affine System Identification. In *Proceedings of the 16th IFAC Symposium on System Identification*, pages 344–355, 2012.

[97] M. E. Gegúndez, J. Aroba, and J. M. Bravo. Identification of piecewise affine systems by means of fuzzy clustering and competitive learning. *Engineering Applications of Artificial Intelligence*, 21(8):1321–1329, 2008.

[98] T. Georgiou and M. Smith. Optimal robustness in the gap metric. *Automatic Control, IEEE Transactions on*, 35(6):673–686, 1990.

[99] S. Gil and B. Williams. Beyond local optimality: An improved approach to hybrid model learning. In *Proceedings of the 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, pages 3938–3945, 2009.

[100] L. Giovanini, G. Sanchez, and M. Benosman. Observer-based adaptive control using multiple-models switching and tuning. *IET Control Theory & Applications*, 8(4):235–247, 2014.

[101] K. Glover and D. McFarlane. Robust stabilization of normalized coprime factor plant descriptions with h-infinity-bounded uncertainty. *Automatic Control, IEEE Transactions on*, 34(8):821–830, 1989.

[102] R. Goebel, R. Sanfelice, and A. Teel. Hybrid dynamical systems. *IEEE Control Systems Magazine*, 29(2):28–93, 2009.

[103] D. Gómez-Gutiérrez, S. Čelikovský, A. Ramírez-Treviño, and B. Castillo-Toledo. On the observer design problem for continuous-time switched linear systems with unknown switchings. *Journal of the Franklin Institute*, 352(4):1595–1612, 2015.

[104] A. Goudjil, M. Pouliquen, E. Pigeon, and O. Gehan. A Real-Time Identification Algorithm for Switched linear Systems with Bounded Noise. In *Proceedings of the European Control Conference*, pages 2626–2631, 2016.

[105] J. W. Grizzle, C. Chevallereau, A. D. Ames, and R. W. Sinnet. 3d bipedal robotic walking: Models, feedback control, and open problems. *IFAC Proceedings Volumes*, 43(14):505–532, 2010.

[106] M. Halimi, G. Millerioux, and J. Daafouz. Model-Based Modes Detection and Discernibility for Switched Affine Discrete-Time Systems. *Automatic Control, IEEE Transactions on*, 60(6):1501–1514, 2015.

[107] Z. Han and K. Narendra. New Concepts in Adaptive Control Using Multiple Models. *Automatic Control, IEEE Transactions on*, 57(1):78–89, 2012.

[108] S. Hecker and H. Pfifer. Affine LPV-modeling for the ADDSAFE benchmark. *Control Engineering Practice*, 31:126–134, 2014.

[109] W. P. M. H. Heemels, B. d. Schutter, and A. Bemporad. Equivalence of hybrid dynamical models. *Automatica*, 37(7):1085–1091, 2001.

[110] J. P. Hespanha, D. Liberzon, and A. Morse. Hysteresis-based switching algorithms for supervisory control of uncertain systems. *Automatica*, 39(2):236–272, 2003.

[111] J. P. Hespanha, D. Liberzon, and A. Morse. Overcoming the limitations of adaptive control by means of logic-based switching. *Systems & Control Letters*, 49(1):49–65, 2003.

[112] J. P. Hespanha, D. Liberzon, A. Morse, B. Anderson, T. S. Brinsmead, and F. De Bruyne. Multiple Model Adaptive Control. Part 2: Switching. *International Journal of Robust and Nonlinear Control*, 11(5):479–496, 2001.

[113] J. Imura. Classification and stabilizability analysis of bimodal piecewise affine systems. *International Journal of Robust and Nonlinear Control*, 12(10):897–926, 2002.

[114] P. Ioannou and J. Sun. *Robust adaptive control*. Prentice-Hall, Upper Saddle River and NJ, 1996.

[115] A. Isidori. *Nonlinear control systems*. Communications and control engineering series. Springer Science & Business Media, 3rd edition, 1995.

[116] A. M. Ivanescu, T. Albin, D. Abel, and T. Seidl. Employing correlation clustering for the identification of piecewise affine models. In *Proceedings of the 2011 Workshop on Knowledge Discovery, Modeling and Simulation*, pages 7–14, 2011.

[117] A. H. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, New York, 1970.

[118] X. Jin and B. Huang. Robust identification of piecewise/switching autoregressive exogenous process. *AIChE Journal*, 56(7):1829–1844, 2010.

[119] M. Johansson. *Piecewise linear control systems: A computational approach*. Springer, Berlin, 2003.

[120] M. Johansson and A. Rantzer. Computation of piecewise quadratic Lyapunov functions for hybrid systems. *Automatic Control, IEEE Transactions on*, 43(4):555–559, 1998.

[121] A. L. Juloski, W. P. M. H. Heemels, G. Ferrari-Trecate, R. Vidal, S. Paoletti, and J. Niessen. Comparison of four procedures for the indentification of hybrid systems. In *Hybrid systems: computation and control*, pages 354–369. Springer, 2005.

[122] A. L. Juloski, S. Weiland, and W. Heemels. A Bayesian approach to identification of hybrid systems. *Automatic Control, IEEE Transactions on*, 50(10):1520–1533, 2005.

[123] S. M. Jung and P. Parks. Variable Forgetting Factor Recursive Total Least Squares Algorithm for FIR Adaptive filtering. In *International Conference on Electronics Engineering and Informatics (ICEEI)*, pages 170–174, 2012.

[124] 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*, 73(6):383–390, 1955.

[125] R. Kamalapurkar, B. Reish, G. Chowdhary, and W. E. Dixon. Concurrent learning for parameter estimation using dynamic state-derivative estimators. *arXiv:1507.08903*, 2015.

[126] R. Kamalapurkar, P. Walters, and W. Dixon. Concurrent learning-based approximate optimal regulation. In *Proceedings of the 52nd IEEE Conference on Decision and Control*, pages 6256–6261, 2013.

[127] H. K. Khalil. *Nonlinear systems*. Prentice Hall, Upper Saddle River, 3rd edition, 2002.

[128] J. Kivinen, A. Smola, and R. Williamson. Online Learning with Kernels. *Signal Processing, IEEE Transactions on*, 52(8):2165–2176, 2004.

[129] J. Klotz, R. Kamalapurkar, and W. E. Dixon. Concurrent learning-based network synchronization. In *Proceedings of the American Control Conference (ACC)*, pages 796–801, 2014.

[130] M. Kuipers and P. Ioannou. Multiple Model Adaptive Control With Mixing. *Automatic Control, IEEE Transactions on*, 55(8):1822–1836, 2010.

[131] C. Y. Lai. *Identification and Control of Nonlinear Systems using Multiple Models*. PhD thesis, National University of Singapore, Singapore, 2011.

[132] C. Y. Lai, C. Xiang, and T. H. H. Lee. Identification and control of nonlinear systems using piecewise affine models. In *Proceedings of the 11th International Conference on Control Automation Robotics & Vision (ICARCV)*, pages 2259–2265, 2010.

[133] C. Y. Lai, C. Xiang, and T. H. H. Lee. Identification and Control of Nonlinear Systems via Piecewise Affine Approximation. In *Proceedings of the 49th IEEE Conference on Decision and Control (CDC)*, pages 6395–6402, 2010.

[134] C. Y. Lai, C. Xiang, and T. H. H. Lee. Identification of Piecewise Affine Systems and Nonlinear Systems using Multiple Models. In *Proceedings of the 8th IEEE International Conference on Control and Automation (ICCA)*, pages 2005–2012, 2010.

[135] F. Lamnabhi-Lagarrigue and J. Lunze. *Handbook of hybrid systems control: Theory, tools, applications*. Cambridge University Press, Cambridge [etc.], 2009.

[136] P. Laskov, C. Gehl, S. Krüger, and K.-R. Müller. Incremental Support Vector Learning: Analysis, Implementation and Applications. *Journal of Machine Learning Research*, 7:1909–1936, 2006.

[137] Z. Lassoued and K. Abderrahim. New Results on PWARX Model Identification Based on Clustering Approach. *International Journal of Automation and Computing*, 11(2):180–188, 2014.

[138] F. Lauer and G. Bloch. A new hybrid system identification algorithm with automatic tuning. In *Proceedings of the 17th IFAC World Congress*, pages 10207–10212, 2008.

[139] F. Lauer and G. Bloch. Switched and PieceWise Nonlinear Hybrid System Identification. In *Hybrid Systems: Computation and Control*, volume 4981, pages 330–343. Springer Berlin Heidelberg, 2008.

[140] F. Lauer, G. Bloch, and R. Vidal. A continuous optimization framework for hybrid system identification. *Automatica*, 47(3):608–613, 2011.

[141] F. Lauer, R. Vidal, and G. Bloch. A product-of-errors framework for linear hybrid system identification. In *Proceedings of the 15th IFAC Symposium on System Identification*, pages 563–568, 2009.

[142] E. Lavretsky. Combined/Composite Model Reference Adaptive Control. *Automatic Control, IEEE Transactions on*, 54(11):2692–2697, 2009.

[143] E. Lavretsky, Gibson, T., E., and A. M. Annaswamy. Projection Operator in Adaptive Systems. arXiv preprint arXiv:1112.4232, 2011.

[144] E. Lavretsky and K. A. Wise. *Robust and Adaptive Control*. Springer London, London, 2013.

[145] V. L. Le, F. Lauer, L. Bako, and G. Bloch. Learning nonlinear hybrid systems: from sparse optimization to support vector regression. In *Proceedings of the 16th International Conference on Hybrid Systems: Computation and Control (HSCCC '13)*, pages 33–42, 2013.

[146] C. Lee, Z. Ping, and H. Shim. On-line Switching Signal Estimation of Switched Linear Systems with Measurement Noise. In *Proceedings of the European Control Conference*, pages 2180–2185, 2013.

[147] H. Lehouche, H. Guéguen, and B. Mendil. Supervisory control based on closed-loop adaptive control approach of nonlinear systems. *Asian Journal of Control*, 14(1):258–270, 2012.

[148] X. Li, Z. Zhanlue, and B. L. Xiao. General model-set design methods for multiple-model approach. *Automatic Control, IEEE Transactions on*, 50(9):1260–1276, 2005.

[149] D. Liberzon. *Switching in systems and control*. Birkhauser, Boston and MA, 2003.

[150] J.-N. Lin and R. Unbehauen. Canonical piecewise-linear approximations. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 39(8):697–699, 1992.

[151] J.-N. Lin and R. Unbehauen. Canonical representation: from piecewise-linear function to piecewise-smooth functions. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 40(7):461–468, 1993.

[152] L. Ljung. *System identification: Theory for the user*. Prentice-Hall, Englewood Cliffs and NJ, 1987.

[153] S. Lloyd. Least squares quantization in PCM. *Information Theory, IEEE Transactions on*, 28(2):129–137, 1982.

[154] J. Löfberg. YALMIP: a toolbox for modeling and optimization in MATLAB. In *IEEE International Symposium on Computer Aided Control Systems Design*, pages 284–289, 2004.

[155] J. Löfberg. Modeling and solving uncertain optimization problems in YALMIP. In *Proceedings of the 17th IFAC World Congress*, pages 1337–1341, 2008.

[156] L. R. Lopes, G. Araujo Borges, and J. Yoshiyuki Ishihara. New algorithm for identification of discrete-time switched linear systems. In *Proceedings of the American Control Conference (ACC)*, pages 6219–6224, 2013.

[157] A. d. Luca and R. Mattone. Sensorless robot collision detection and hybrid force/motion control. In *IEEE International Conference on Robotics and Automation*, pages 999–1004, 2005.

[158] B. Luitel and G. K. Venayagamoorthy. Particle swarm optimization with quantum infusion for system identification. *Engineering Applications of Artificial Intelligence*, 23(5):635–649, 2010.

[159] H. Mahdianfar, S. Ozgoli, and H. R. Momeni. Robust multiple model adaptive control: Modified using v-gap metric. *International Journal of Robust and Nonlinear Control*, 21(18):2027–2063, 2011.

[160] I. Markovsky and S. van Huffel. Overview of total least-squares methods. *Signal Processing*, 87(10):2283–2302, 2007.

[161] R. Martin. A metric for ARMA processes. *Signal Processing, IEEE Transactions on*, 48(4):1164–1170, 2000.

[162] I. Maruta and H. Ohlsson. Compression Based Identification of PWA Systems. In *Proceedings of the 19th IFAC World Congress*, pages 4985–4992, 2014.

[163] I. Maruta and T. Sugie. Identification of PWA models via data compression based on l1 optimization. In *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, pages 2800–2805, 2011.

[164] I. Maruta and T. Sugie. Identification of PWA Models Via Optimal Data Compression. In *Proceedings of the 16th IFAC Symposium on System Identification*, pages 368–373, 2012.

[165] I. Maruta, T. Sugie, and T. Kim. Identification of multiple mode models via Distributed Particle Swarm Optimization. In *Proceedings of the 18th IFAC World Congress*, pages 7743–7748, 2011.

[166] D. McFarlane and K. Glover. A loop-shaping design procedure using H/sub infinity / synthesis. *Automatic Control, IEEE Transactions on*, 37(6):759–769, 1992.

[167] D. C. McFarlane and K. Glover. *Robust Controller Design Using Normalized Coprime Factor Plant Descriptions*, volume 138. Springer-Verlag, Berlin/Heidelberg, 1990.

[168] G. Michaletzky and L. Gerencser. BIBO stability of linear switching systems. *Automatic Control, IEEE Transactions on*, 47(11):1895–1898, 2002.

[169] A. Morse. Supervisory control of families of linear set-point controllers: Part 1: Exact matching. *Automatic Control, IEEE Transactions on*, 41(10):1413–1431, 1996.

[170] A. Morse. Supervisory control of families of linear set-point controllers. 2. Robustness. *Automatic Control, IEEE Transactions on*, 42(11):1500–1515, 1997.

[171] E. Münz and V. Krebs. Identification of hybrid systems using a priori knowledge. In *Proceedings of the 15th IFAC World Congress*, 2002.

[172] E. Münz and V. Krebs. Continuous optimization approaches to the identification of piecewise affine systems. In *Proceedings of the 16th IFAC World Congress*, 2005.

[173] H. Nakada, K. Takaba, and T. Katayama. Identification of piecewise affine systems based on statistical clustering technique. *Automatica*, 41(5):905–913, 2005.

[174] K. Narendra and A. M. Annaswamy. Persistent excitation in adaptive systems. *International Journal of Control*, 45(1):127–160, 1987.

[175] K. Narendra and A. M. Annaswamy. *Stable adaptive systems*. Dover Publications, Mineola and NY, 2005.

[176] K. Narendra and J. Balakrishnan. Adaptive control using multiple models. *Automatic Control, IEEE Transactions on*, 42(2):171–187, 1997.

[177] K. Narendra, J. Balakrishnan, and M. Ciliz. Adaptation and learning using multiple models, switching, and tuning. *IEEE Control Systems Magazine*, 15(3):37–51, 1995.

[178] K. Narendra, O. Driollet, M. Feiler, and K. George. Adaptive Control Using Multiple Models, Switching and Tuning. *International Journal of Adaptive Control and Signal Processing*, 17:1–16, 2003.

[179] K. Narendra and Z. Han. Location of models in multiple-model based adaptive control for improved performance. In *Proceedings of the American Control Conference (ACC)*, pages 117–122, 2010.

[180] S. Nazari. *Identification of Switching ARX Models for Hybrid Systems*. PhD thesis, University of Alberta, Edmonton and Alberta, 2013.

[181] S. Nazari, Z. Qing, and H. Biao. An improved algebraic geometric solution to the identification of switched ARX models with noise. In *Proceedings of the American Control Conference (ACC)*, pages 1230–1235, 2011.

[182] S. Nazari, Z. Qing, and H. Biao. Matrix-wise approach for identification of multi-mode Switched ARX models with noise. In *Proceedings of the American Control Conference (ACC)*, pages 3402–3407, 2012.

[183] H. Ohlsson and L. Ljung. Identification of Piecwise Affine Systems Using Sum-of-Norms Regularization. In *Proceedings of the 18th IFAC World Congress*, pages 6640–6645, 2011.

[184] H. Ohlsson and L. Ljung. Identification of switched linear regression models using sum-of-norms regularization. *Automatica*, 49(4):1045–1050, 2013.

[185] H. Ohlsson, L. Ljung, and S. Boyd. Segmentation of ARX-models using sum-of-norms regularization. *Automatica*, 46(6):1107–1111, 2010.

[186] N. Ozay, C. Lagoa, and M. Sznaier. Robust identification of switched affine systems via moments-based convex optimization. In *Proceedings of the 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, pages 4686–4691, 2009.

[187] N. Ozay, C. Lagoa, and M. Sznaier. Set membership identification of switched linear systems with known number of subsystems. *Automatica*, 51:180–191, 2015.

[188] N. Ozay, M. Sznaier, C. Lagoa, and O. Camps. A sparsification approach to set membership identification of a class of affine hybrid systems. In *Proceedings of the 47th IEEE Conference on Decision and Control*, pages 123–130, 2008.

[189] N. Ozay, M. Sznaier, C. M. Lagoa, and O. I. Camps. A Sparsification Approach to Set Membership Identification of Switched Affine Systems. *Automatic Control, IEEE Transactions on*, 57(3):634–648, 2012.

[190] F. M. Pait and F. Kassab. On a class of switched, robustly stable, adaptive systems. *International Journal of Adaptive Control and Signal Processing*, 15(3):213–238, 2001.

[191] S. Paoletti. *Identification of piecewise affine models*. PhD thesis, University of Siena, Italy, 2004.

[192] S. Paoletti, A. Garulli, J. Roll, and A. Vicino. A necessary and sufficient condition for input-output realization of switched affine state space models. In *Proceedings of the 47th IEEE Conference on Decision and Control*, pages 935–940, 2008.

[193] S. Paoletti, A. L. Juloski, G. Ferrari-Trecate, and R. Vidal. Identification of hybrid systems: a tutorial. *European Journal of Control*, 13(2-3):242–260, 2007.

[194] S. Paoletti, J. Roll, A. Garulli, and A. Vicino. On the Input-Output Representation of Piecewise Affine State Space Models. *Automatic Control, IEEE Transactions on*, 55(1):60–73, 2010.

[195] M. Papageorgiu, M. Leibold, and M. Buss. *Optimierung: Statische, dynamische, stochastische Verfahren für die Anwendung*. Springer-Verlag, Berlin, 4th edition, 2015.

[196] K. M. Pekpe and S. Lecoeuche. Online Classification of Switching Models Based on Subspace Framework. In *Proceedings of the 2nd IFAC Conference on Analysis and Design of Hybrid Systems*, pages 154–159, 2006.

[197] K. M. Pekpe and S. Lecœuche. Online clustering of switching models based on a subspace framework. *Nonlinear Analysis: Hybrid Systems*, 2(3):735–749, 2008.

[198] K. M. Pekpe, G. Mourot, and J. Ragot. Identification of switching systems using change detection technique in the subspace framework. In *Proceedings of the 43rd IEEE Conference on Decision and Control*, pages 3720–3725, 2004.

[199] M. Petreczky and L. Bako. On the notion of persistence of excitation for linear switched systems. In *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, pages 1840–1847, 2011.

[200] M. Petreczky, L. Bako, and J. H. van Schuppen. Identifiability of discrete-time linear switched systems. In *Proceedings of the 13th ACM international conference on Hybrid Systems: computation and control*, pages 141–150, 2010.

[201] M. Petreczky, L. Bako, and J. H. van Schuppen. Realization theory of discrete-time linear switched systems. *Automatica*, 49(11):3337–3344, 2013.

[202] M. Petreczky, A. Tanwani, and S. Trenn. Observability of Switched Linear Systems. In *Hybrid Dynamical Systems*, volume 457 of *Lecture notes in control and information sciences*, pages 205–240. Springer International Publishing, 2015.

[203] M. Petreczky and J. H. van Schuppen. Realization Theory for Linear Hybrid Systems. *Automatic Control, IEEE Transactions on*, 55(10):2282–2297, 2010.

[204] Pimenta, L., C.A., V. Kumar, Mesquita, R., C., and Pereira, G., A.S. Sensing and Coverage for a Network of Heterogeneous Robots. In *Proceedings of the 47th IEEE Conference on Decision and Control*, pages 3947–3952, 2008.

[205] R. Porreca and G. Ferrari-Trecate. Partitioning datasets based on equalities among parameters. *Automatica*, 46(2):460–465, 2010.

[206] F. R. Pour Safaei, J. P. Hespanha, and G. Stewart. On controller initialization in multivariable switching systems. *Automatica*, 48(12):3157–3165, 2012.

[207] S. Rhode, F. Bleimund, and F. Gauterin. Recursive Generalized Total Least Squares with Noise Covariance Estimation. In *Proceedings of the 19th IFAC World Congress*, pages 4637–4643, 2014.

[208] J. Richter, W. Heemels, N. van de Wouw, and J. Lunze. Reconfigurable control of piecewise affine systems with actuator and sensor faults: Stability and tracking. *Automatica*, 47(4):678–691, 2011.

[209] L. Rodrigues and S. Boyd. Piecewise-affine state feedback for piecewise-affine slab systems using convex optimization. *Systems & Control Letters*, 54:835–853, 2005.

[210] J. Roll, A. Bemporad, and L. Ljung. Identification of piecewise affine systems via mixed-integer programming. *Automatica*, 40:37–50, 2004.

[211] F. Rosenqvist and A. Karlström. Realisation and estimation of piecewise-linear output-error models. *Automatica*, 41(3):545–551, 2005.

[212] S. Sadraddini and C. Belta. Formal methods for adaptive control of dynamical systems. In *arXiv preprint 1703.07704*, 2017.

[213] S. Sadraddini and C. Belta. Formal guarantees in data-driven model identification and control synthesis. In *Proceedings of the 21st International Conference on Hybrid Systems: Computation and Control*, pages 147–156, 2018.

[214] M. Safonov and C. T. Tung. The unfalsified control concept and learning. *Automatic Control, IEEE Transactions on*, 42(6):843–847, 1997.

[215] Q. Sang. *Model Reference Adaptive Control of Piecewise Linear Systems with Applications to Aircraft Flight Control*. PhD thesis, University of Virginia, 2012.

[216] Q. Sang and G. Tao. Adaptive control of piecewise linear systems: The state tracking case. In *Proceedings of the American Control Conference (ACC)*, pages 4040–4045, 2010.

[217] Q. Sang and G. Tao. Adaptive control of piecewise linear systems with applications to NASA GTM. In *Proceedings of the American Control Conference (ACC)*, pages 1157–1162, 2011.

[218] Q. Sang and G. Tao. Adaptive Control of Piecewise Linear Systems: the State Tracking Case. *Automatic Control, IEEE Transactions on*, 57(2):522–528, 2012.

[219] Q. Sang, G. Tao, and J. Guo. In *Proceedings of the American Control Conference (ACC)*.

[220] P. Santana, S. Lane, E. Timmons, B. Williams, and C. Forster. Learning Hybrid Models with Guarded Transitions. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, 2015.

[221] E. d. Santis, M. Di Benedetto, and G. Pola. On observability and detectability of continuous-time linear switching systems. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, pages 5777–5782, 2003.

[222] C. Seatzu, D. Corona, A. Giua, and A. Bemporad. Optimal control of continuous-time switched affine systems. *Automatic Control, IEEE Transactions on*, 51(5):726–741, 2006.

[223] M. G. Sefidmazgi, M. M. Kordmahalleh, A. Homaifar, and A. Karimoddini. Switched Linear System Identification based on Bounded-Switching Clustering. In *Proceedings of the American Control Conference (ACC)*, pages 1806–1811, 2015.

[224] R. Shorten, F. Wirth, O. Mason, K. Wulff, and C. King. Stability Criteria for Switched and Hybrid Systems. *SIAM Review*, 49(4):545–592, 2007.

[225] D. Simon. *Optimal state estimation: Kalman, H [infinity] and nonlinear approaches*. Wiley-Interscience, Hoboken and N.J, 2006.

[226] S. Skogestad and I. Postlethwaite. *Multivariable Feedback Control- Analysis and Design*. Wiley, 2007.

[227] E. D. Sontag. Nonlinear regulation: The piecewise linear approach. *Automatic Control, IEEE Transactions on*, 26(2):346–358, 1981.

[228] E. D. Sontag. Interconnected automata and linear systems: A theoretical framework in discrete-time. In *Hybrid Systems III*, volume 1066 of *Lecture notes in computer science*, pages 436–448. Springer Berlin Heidelberg, 1996.

[229] Y. Stergiopoulos, M. Thanou, and A. Tzes. Distributed Collaborative Coverage-Control Schemes for Non-Convex Domains. *Automatic Control, IEEE Transactions on*, 60(9):2422–2427, 2015.

[230] O. Stursberg, S. Kowalewski, I. Hoffmann, and J. Preußig. Comparing Timed and Hybrid Automata as Approximations of Continuous Systems. In *Hybrid Systems IV*, pages 361–377. Springer, 1997.

[231] Z. Sun and S. Ge. Analysis and synthesis of switched linear control systems. *Automatica*, 41(2):181–195, 2005.

[232] N. Syed, H. Liu, and K. Sung. Incremental learning with support vector machines. In *International Joint Conference on Artificial Intelligence*, 1999.

[233] M. Tabatabaei-Pour, M. Gholami, K. Salahshoor, and H. R. Shaker. A Clustering-Based Bounded-Error Approach for Identification of PWA Hybrid Systems. In *9th International Conference on Control, Automation, Robotics and Vision*, pages 1–6, 2006.

[234] M. Tabatabaei-Pour, M. Gholami, H. Shaker, and B. Moshiri. Recursive Identification of Piecewise Affine Hybrid Systems. In *9th International Conference on Control, Automation, Robotics and Vision*, pages 1–6, 2006.

[235] M. Tabatabaei-Pour, K. Salahshoor, and B. Moshiri. A Modified k-plane Clustering Algorithm for Identification of Hybrid Systems. In *Proceedings of the Sixth World Congress on Intelligent Control and Automation*, volume 1, pages 1333–1337, 2006.

[236] S. Taguchi, T. Suzuki, S. Hayakawa, and S. Inagaki. Identification of hybrid system based on Probability weighted multiple ARX model. In *ICCAS-SICE, 2009*, pages 4837–4842, 2009.

[237] S. Taguchi, T. Suzuki, S. Hayakawa, and S. Inagaki. Identification of Probability weighted multiple ARX models and its application to behavior analysis. In *Proceedings of the 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, pages 3952–3957, 2009.

[238] A. Tanwani, H. Shim, and D. Liberzon. Observability for Switched Linear Systems: Characterization and Observer Design. *Automatic Control, IEEE Transactions on*, 58(4):891–904, 2013.

[239] A. Tanwani, H. Shim, and D. Liberzon. Observer Design for Switched Linear Systems with State Jumps. In *Hybrid Dynamical Systems*, volume 457 of *Lecture notes in control and information sciences*, pages 179–203. Springer International Publishing, 2015.

[240] G. Tao. *Adaptive control design and analysis*. Adaptive and learning systems for signal processing, communications, and control. Wiley-Interscience, Hoboken and N.J, 2003.

[241] G. Tao. Multivariable adaptive control: A survey. *Automatica*, 50(11):2737–2764, 2014.

[242] L. Q. Thuan, T. van den Boom, and S. Baldi. Online Identification of Continuous Bimodal and Trimodal Piecewise Affine Systems. In *Proceedings of the European Control Conference*, pages 1075–1080, 2016.

[243] Y. Tian, T. Floquet, L. Belkoura, and W. Perruquetti. Switching time estimation for linear switched systems: an algebraic approach. In *Proceedings of the 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, pages 3909–3913, 2009.

[244] Y. Tian, T. Floquet, L. Belkoura, and W. Perruquetti. Algebraic switching time identification for a class of linear hybrid systems. *Nonlinear Analysis: Hybrid Systems*, 5.2:233–241, 2011.

[245] F. D. Torrisi and A. Bemporad. HYSDEL-a tool for generating computational hybrid models for analysis and synthesis problems. *Control Systems Technology, IEEE Transactions on*, 12(2):235–249, 2004.

[246] L. van Luong. *Hybrid Dynamical System Identification: gemometry, sparsity, and nonlinearities*. PhD thesis, Université de Lorraine, Lorraine and France, 2013.

[247] V. N. Vapnik. *The nature of statistical learning theory*. Springer, New York, 1995.

[248] V. N. Vapnik. *Statistical learning theory*. Adaptive and learning systems for signal processing, communications, and control. Wiley, New York, 1998.

[249] V. Verdult and M. Verhaegen. Subspace identification of piecewise linear systems. In *Proceedings of the 43rd IEEE Conference on Decision and Control*, pages 3838–3843, 2004.

[250] R. Vidal. Recursive identification of switched ARX systems. *Automatica*, 44(9):2274–2287, 2008.

[251] R. Vidal, S. Soatto, M. Yi, and S. Sastry. An algebraic geometric approach to the identification of a class of linear hybrid systems. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, pages 167–172, 2003.

[252] G. Vinnicombe. Frequency domain uncertainty and the graph topology. *Automatic Control, IEEE Transactions on*, 38(9):1371–1383, 1993.

[253] G. Vinnicombe. A v-gap distance for uncertain and nonlinear systems. In *Proceedings of the 38th IEEE Conference on Decision and Control*, pages 2557–2562, 1999.

[254] S. V. Vishwanathan, A. J. Smola, and R. Vidal. Binet-Cauchy Kernels on Dynamical Systems and its Application to the Analysis of Dynamic Scenes. *International Journal of Computer Vision*, 73(1):95–119, 2007.

[255] D. Vizer and G. Mercère. H_{infty}-Based LPV Model Identification from Local Experiments with a Gap Metric-Based Operating Point Selection. In *Proceedings of the European Control Conference (ECC)*, pages 388–393, 2014.

[256] L. Vu and D. Liberzon. Supervisory Control of Uncertain Linear Time-Varying Systems. *Automatic Control, IEEE Transactions on*, 56(1):27–42, 2011.

[257] J. Wang. *Identification of Switched Linear Systems*. PhD thesis, University of Alberta, Edmonton and Alberta, 2013.

[258] J. Wang and T. Chen. Online identification of switched linear output error models. In *IEEE International Symposium on Computer-Aided Control System Design (CACSD)*, pages 1379–1384, 2011.

[259] Q. Wang, Y. Hou, and C. Dong. Model reference robust adaptive control for a class of uncertain switched linear systems. *International Journal of Robust and Nonlinear Control*, 22(9):1019–1035, 2012.

[260] X. Wang and J. Zhao. Adaptive state tracking of switched systems based on a hyperstability criterion. *International Journal of Adaptive Control and Signal Processing*, 28(1):28–39, 2014.

[261] S. Weiland, A. L. Juloski, and B. Vet. On the equivalence of switched affine models and switched ARX models. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 2614–2618, 2006.

[262] C. Weitian and B. Anderson. A combined multiple model adaptive control scheme and its application to nonlinear systems with nonlinear parameterization. *Automatic Control, IEEE Transactions on*, 57(7):1778–1782, 2012.

[263] C. Wen, S. Wang, X. Jin, and X. Ma. Identification of dynamic systems using Piecewise-Affine basis function models. *Automatica*, 43(10):1824–1831, 2007.

[264] C. Wu, J. Zhao, and X.-M. Sun. Adaptive tracking control for uncertain switched systems under asynchronous switching. *International Journal of Robust and Nonlinear Control*, 25(17):3457–3477, 2015.

[265] J. Xie and J. Zhao. Model reference adaptive control for nonlinear switched systems under asynchronous switching. *International Journal of Adaptive Control and Signal Processing*, 2016.

[266] J. Xu, X. Huang, and S. Wang. Adaptive hinging hyperplanes and its applications in dynamic system identification. *Automatica*, 45(10):2325–2332, 2009.

[267] J. Xu and L. Xie. *Control and estimation of piecewise affine systems*. Woodhead Publishing, Amsterdam and Boston, 2014.

[268] X. Xu and P. J. Antsaklis. Optimal control of switched autonomous systems. In *Proceedings of the 41st IEEE Conference on Decision and Control*, pages 4401–4406, 2002.

[269] J. Yuan and W. Wonham. Probing signals for model reference identification. *Automatic Control, IEEE Transactions on*, 22(4):530–538, 1977.

[270] X. Zhao, H. Liu, J. Zhang, and H. Li. Multiple-Mode Observer Design for a Class of Switched Linear Systems. *Automation Science and Engineering, IEEE Transactions on*, 12(1):272–280, 2015.

Own Publications and Supervised Student Projects

[271] P. Bareiß. Adaptive Control for Piecewise Affine Systems. *Semesterarbeit*, Chair of Automatic Control Engineering, Technical University of Munich (TUM), 2014.

[272] F. Hans. Vergleich zweier Algorithmen zur adaptiven Identifikation stückweise affiner Systeme. *Forschungspraxis*, Chair of Automatic Control Engineering, Technical University of Munich (TUM), 2014.

[273] S. Kersting and M. Buss. Adaptive identification of continuous-time switched linear and piecewise linear systems. In *European Control Conference (ECC)*, pages 31–36, 2014.

[274] S. Kersting and M. Buss. Concurrent learning adaptive identification of piecewise affine systems. In *Proceedings of the 53rd IEEE Conference on Decision and Control (CDC)*, pages 3930–3935, 2014.

[275] S. Kersting and M. Buss. Metrics to compare and control dynamical systems. In *European Control Conference (ECC)*, pages 738–743, 2014.

[276] S. Kersting and M. Buss. Online identification of piecewise affine systems. In *UKACC 10th International Conference on Control*, pages 90–95, 2014.

[277] S. Kersting and M. Buss. An alternative approach to switching hyperplane estimation in PWA systems. In *American Control Conference (ACC)*, pages 4449–4454, 2015.

[278] S. Kersting and M. Buss. Removing erroneous history stack elements in concurrent learning. In *Proceedings of the 54th IEEE Conference on Decision and Control*, pages 7604–7609. 2015.

[279] S. Kersting and M. Buss. How to Systematically Distribute Candidate Models and Robust Controllers in Multiple-Models Adaptive Control: A Coverage Control Approach. *Automatic Control, IEEE Transactions on*, 63(4):1075–1089, Apr. 2018.

[280] S. Kersting and M. Buss. Direct and Indirect Model Reference Adaptive Control for Multivariable Piecewise Affine Systems. *Automatic Control, IEEE Transactions on*, 62(11):5634–5649, Nov. 2017.

[281] S. Kersting and M. Buss. Recursive estimation in piecewise affine systems using parameter identifiers and concurrent learning. *International Journal of Control*, pages 1–18, published online 20. Oct. 2017.

[282] A. Kunz. Learning and Similarity Reasoning of Dynamical Systems. *Diplomarbeit*, Chair of Automatic Control Engineering, Technical University of Munich (TUM), 2012.

[283] M. Lang. Switching Signal Estimation in Piecewise Affine Systems. *Advanced Seminar*, Chair of Automatic Control Engineering, Technical University of Munich (TUM), 2014.

[284] M. Löb. Identification and Control of Piecewise Affine Systems. *Bachelor Thesis*, Chair of Automatic Control Engineering, Technical University of Munich (TUM), 2013.

[285] D. Ossadnik. Optimal Model Distribution in Supervisory Control. *Advanced Seminar*, Chair of Automatic Control Engineering, Technical University of Munich (TUM), 2016.

[286] J. Pfort. Concurrent Learning Adaptive Identification of a Nonlinear Hybrid System. *Bachelor Thesis*, Chair of Automatic Control Engineering, Technical University of Munich (TUM), 2015.

[287] P. Schmitt. Incremental Support Vector Machines. *Advanced Seminar*, Chair of Automatic Control Engineering, Technical University of Munich (TUM), 2014.

[288] P. Schmitt and M. Gaab. Adaptive Control of Piecewise Affine Systems. *Projektpraktikum*, Chair of Automatic Control Engineering, Technical University of Munich (TUM), 2014.

[289] M. J. Stephan. Identifying a Piecewise Affine Model for a Robotic Contact Problem. *Ingenieurpraxis*, Chair of Automatic Control Engineering, Technical University of Munich (TUM), 2015.

[290] H. Stüber. Metrics for Learning and Similarity Reasoning of Dynamical Systems. *Bachelor Thesis*, Chair of Automatic Control Engineering, Technical University of Munich (TUM), 2012.

[291] W. L. Thet. Implementation of Multiple Models Adaptive Control. *Bachelor Thesis*, Chair of Automatic Control Engineering, Technical University of Munich (TUM), 2013.

[292] R. Ursu. On-line identification of regions and subsystem dynamics for continuous-time piecewise affine systems. *Master Thesis*, Chair of Automatic Control Engineering, Technical University of Munich (TUM), 2014.

[293] L. Vergara. Model Reference Adaptive Control of a PWA Approximation for a Nonlinear System. *Ingenieurpraxis*, Chair of Automatic Control Engineering, Technical University of Munich (TUM), 2015.

[294] L. Vergara. Pilot study for novel brain-computer interfaces based on multiple models and p300. *Projektpraktikum*, Chair of Automatic Control Engineering, Technical University of Munich (TUM), 2013.

[295] M. Wagner. Improving the Performance of Concurrent Learning in the Identification of PWA Systems. *Bachelor Thesis*, Chair of Automatic Control Engineering, Technical University of Munich (TUM), 2014.

[296] P. Weiler. Model Reference Adaptive Control for a PWA Approximation of a Nonlinear System. *Ingenieurpraxis*, Chair of Automatic Control Engineering, Technical University of Munich (TUM), 2015.