UNIVERSITÀ DEGLI STUDI DI SIENA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

DOTTORATO DI RICERCA IN INGEGNERIA DELL'INFORMAZIONE

– CICLO XVI –

IDENTIFICATION OF PIECEWISE AFFINE MODELS

Simone Paoletti

Advisor: Prof. A. Vicino

Draft version 1.0

Anno Accademico 2002/2003

“Dicono che c’è un tempo per seminare
e uno più lungo per aspettare
Io dico che c’era un tempo sognato
che bisognava sognare”
I. Fossati

A zio Nedo

Contents

Notation

The following lists of symbols, acronyms, etc. are intended to gather notations that are used in this thesis. Occurrence of the same symbol for different purposes will be always notified.

Symbols, Operators and Functions

$\in$	belongs to
$\triangleq$	equal by definition
$\mathbb{R}$	the field of real numbers
$x \in \mathbb{R}$	real number x
$\mathbf{v} \in \mathbb{R}^n$	n-dimensional real column vector $\mathbf{v}$
v_i	the i-th element of vector $\mathbf{v}$
$A \in \mathbb{R}^{m \times n}$	real matrix A with m rows and n columns
A_i	the i-th row of matrix A
A_{ij}	the element in the i-th row and the j-th column of matrix A
$\mathbf{0}$	column vector with all elements equal to zero
$\mathbf{1}$	column vector with all elements equal to one
$\mathrm{rank}(A)$	rank of the matrix A
$\prec, \preceq, \succ, \succeq$	componentwise inequalities (for vectors)
max, min	componentwise maximum or minimum (for vectors)
$\arg\max$, $\arg\min$	maximizing or minimizing argument

$\mathscr{A} \subseteq \mathbb{R}^n$	subset $\mathscr{A}$ of $\mathbb{R}^n$
$\#\mathscr{A}$	cardinality of the set $\mathscr{A}$, if $\mathscr{A}$ is finite
$\lvert x \rvert$	absolute value of $x \in \mathbb{R}$
$\lVert \mathbf{x} \rVert_p$	ℓ_p-norm of $\mathbf{x} \in \mathbb{R}^n$: $\lVert \mathbf{x} \rVert_p = \left(\sum_{i=1}^{n} \lvert x_i \rvert^p \right)^{\frac{1}{p}}$
$\lVert \mathbf{x} \rVert_\infty$	ℓ_∞-norm of $\mathbf{x} \in \mathbb{R}^n$: $\lVert \mathbf{x} \rVert_\infty = \max_{i=1,\ldots,n} \lvert x_i \rvert$
$\binom{n}{k}$	the binomial coefficient, $\binom{n}{k} = \frac{n!}{k!(n-k)!}$
$P(A \mid B)$	probability of the event A given that the event B has occurred
$\mathscr{N}(m, \sigma^2)$	normal distribution with mean m and variance σ^2
$E[X]$	expected value of the random variable X

Acronyms

ARX	Autoregressive Exogenous
ELC	Extended Linear Complementarity
HH	Hinging Hyperplane
HHARX	Hinging Hyperplane Autoregressive Exogenous
H-PWARX	Hammerstein Piecewise Affine Autoregressive Exogenous
JL	Jump Linear
JML	Jump-Markov Linear
LC	Linear Complementarity
MAX FS	Maximum Feasible Subsystem
MIN PFS	Minimum Partition into Feasible Subsystems
MLD	Mixed Logical Dynamical
MMPS	Max-Min-Plus-Scaling
M-RLP	Multicategory Robust Linear Programming
M-SVM	Multicategory Support Vector Machine
OE	Output Error
PWA	Piecewise Affine

PWARX	Piecewise Affine Autoregressive Exogenous
PWL	Piecewise Linear
RLP	Robust Linear Programming
SVM	Support Vector Machine
UBB	Unknown But Bounded
W-PWARX	Wiener Piecewise Affine Autoregressive Exogenous

1

Introduction

System identification deals with the problem of building mathematical models of dynamical systems based on measured data. The most important choice in system identification concerns the set of candidate models within which a model will be fitted to the data. Most common models are linear difference equations descriptions, such as ARX and ARMAX models, as well as linear state-space models. A wide range of linear identification techniques are available; see, *e.g.*, (Ljung, 1999) and references therein. When linear models are not sufficient for describing accurately the dynamics of a system, nonlinear identification must be employed. A large number of nonlinear model structures have been considered and their properties investigated; see, *e.g.*, the survey papers (Sjöberg *et al.*, 1995; Juditsky *et al.*, 1995) and references therein. This thesis is dedicated to the problem of identifying piecewise affine (PWA) models of discrete-time nonlinear systems from input-output data. Piecewise affine systems are obtained by partitioning the state+input set into a finite number of polyhedral regions, and by considering linear/affine systems sharing the same continuous state in each region (Sontag, 1981). Such systems are sufficiently expressive to model a large number of physical processes, and can approximate nonlinear dynamics with arbitrary accuracy. In addition, given the equivalence between PWA systems and several classes of hybrid systems (Bemporad *et al.*, 2000*b*; Heemels *et al.*, 2001), PWA system identification techniques can be used

to obtain hybrid models.

The identification of PWA models is a challenging problem. It involves the estimation of both the parameters of the affine submodels, and the coefficients of the hyperplanes defining the partition of the state+input set (or the regressor set, for models in regression form). This issue clearly underlies a classification problem such that each data point is associated with one submodel. Depending on how the partitioning into regions is done, two alternative approaches can be distinguished: either the partition is defined a priori, or it is estimated along with the submodels. In the first case, data classification is very simple, and estimation of the submodels can be accomplished by resorting to standard linear identification techniques. On the other hand, the number of regions needed to give enough flexibility in the model structure might be very large. In the second approach, the regions are shaped according to the data, thus allowing for fewer regions and parameters. The main difficulty in this case is that the three issues of data classification, parameter estimation and region estimation, being closely related, should be carried out simultaneously. The problem is even more complicated when also the number of submodels must be estimated. A number of approaches resulting in PWA models of nonlinear dynamical systems can be found in different fields, *e.g.*, neural networks, electrical networks, time-series analysis, function approximation; see (Roll, 2003) for a nice overview and classification of different approaches to PWA system identification. Recently, novel contributions to this topic have been proposed in both the hybrid systems and the nonlinear identification community. Roll *et al.* (2004) formulate the identification problem for two subclasses of PWA models, namely Hinging Hyperplane ARX (HHARX) and Wiener PWARX (W-PWARX) models, that lead to mixed-integer linear or quadratic programs. Ferrari-Trecate *et al.* (2003) consider PieceWise affine ARX (PWARX) models and exploit the combined use of clustering, linear identification, and pattern recognition techniques in order to identify both the affine submodels and the polyhedral partition of the regressor set. In (Vidal *et al.*, 2003*b*) an algebraic geometric solution to the identification of PieceWise Linear (PWL) models is proposed. It establishes a connection between PWL system identification, polynomial factorization, and hyperplane clustering. Ragot *et al.* (2003) describe an iterative algorithm, allowing for sequential estimation of the model parameters and data classification through the use of adapted weights.

In this thesis, a different approach, inspired by ideas from set-membership identification, is proposed. The main feature of this approach is the selection of a bound δ on the fitting error (*i.e.*, the difference between the measured output of the system and the predicted output of the model). This enables to address the estimation of the number of submodels, the data classification and the parameter estimation simultaneously, by partitioning a suitable set of linear complementary inequalities derived from data into a minimum number of feasible subsystems (MIN PFS problem). A refinement procedure is also applied in order to reduce misclassifications, and to improve parameter estimates. Region estimation is lastly performed via two-class (Bennett and Mangasarian, 1992; Cortes and Vapnik, 1995) or multi-class (Bennett and Mangasarian, 1994; Bredensteiner and Bennett, 1999) linear separation techniques. The bound δ can be used as a tuning knob to trade off between quality of fit and model complexity. The identified PWA model associates to each submodel a set of feasible parameters, thus allowing for evaluation of the related parametric uncertainty (Milanese and Vicino, 1991). In this thesis, the greedy algorithm (Amaldi and Mattavelli, 2002) for solving the MIN PFS problem with complementary inequalities is also modified in order to obtain improved solutions. The performance of the identification procedure is tested on experimental data from an electronic component placement process in a pick-and-place machine (Juloski *et al.*, 2003).

This chapter, where some general concepts and results are briefly introduced, is structured as follows. The interest in piecewise affine models of dynamical systems will be motivated in Section 1.1, where the connections between piecewise affine and hybrid systems will be also pointed out. In Section 1.2 the general system identification problem is introduced, together with a commonly used family of parametric identification methods, namely the prediction error methods. Some basic ideas of set-membership identification (*i.e.*, identification under the unknown-but-

bounded error description) are recalled in Section 1.3, and techniques for model validation are the subject of Section 1.4. The organization and the main contributions of the thesis, as well as a list of the related publications, are found in Sections 1.5 and 1.6.

1.1 Piecewise affine and hybrid systems

Piecewise Affine (PWA) systems form a special class of nonlinear systems whose state and output maps are both piecewise affine, *i.e.*, affine or linear on each of the components of a finite polyhedral partition of the state+input set. Static and dynamical systems described by piecewise affine maps have been considered in many fields, *e.g.*, neural networks (Batruni, 1991), electrical networks (Leenaerts and Van Bokhoven, 1998), time-series analysis (Tong, 1983). PWA systems can be used to describe nonlinear phenomena that are frequent in practical situations, *e.g.*, where there are changes of the dynamics due to physical limits (such as a tank that can get full or empty, or a bouncing ball which alternates between free fall and elastic contact), bounds on the signals, dead-zones, switches and thresholds. Since piecewise affine maps have universal approximation properties, which essentially means that any (sufficiently smooth) nonlinear function can be arbitrarily well approximated by a piecewise affine function (Lin and Unbehauen, 1992; Breiman, 1993), PWA systems can also be used to approximate nonlinear systems that do not themselves exhibit discontinuous or switching behavior. Besides modelling, PWA systems are suitable for analysis (Chua *et al.*, 1982; Chua and Ying, 1983) and control (Sontag, 1981) of classes of nonlinear systems. However, despite the fact that they are just a composition of linear time-invariant systems, their structural properties such as observability, controllability and stability are complex and articulated (Sontag, 1996; Blondel and Tsitsiklis, 1999) as is typical for nonlinear systems.

Recently there has been a growing interest in PWA systems also because of their connections with hybrid systems. PWA systems are indeed a class of hybrid systems, for which the switching rule between different linear/affine dynamics is

given by a polyhedral partition of the state+input set. *Hybrid systems* are dynamical systems whose behavior is determined by interacting continuous and discrete dynamics. Such systems are characterized by both variables or signals that take values from continuous sets, and variables that take values from discrete, typically finite, sets. These continuous or discrete-valued variables or signals may either depend on independent variables such as time, which also may be continuous or discrete, or be driven asynchronously by external or internal discrete events. In the last decade, both the computer science and the control community have been attracted by this class of systems (*e.g.*, Antsaklis and Nerode, eds., 1998; Morse *et al.*, eds., 1999; Van der Schaft and Schumacher, 2000). The interest is mainly motivated by the large variety of practical situations where physical processes interact with digital components. The continuous dynamics of hybrid systems is indeed typically associated with physical systems, whereas the discrete dynamics may come, for instance, from digital controllers, logic devices and rules, or discrete-event systems with finite automaton representations. Several modelling formalisms have been developed for hybrid systems (*e.g.*, Branicky *et al.*, 1998; Heemels *et al.*, 2001), and the issues of stability analysis (*e.g.*, Branicky, 1998; Johansson and Rantzer, 1998; Liberzon and Morse, 1999), observability and controllability (*e.g.*, Bemporad *et al.*, 2000*b*; Sun *et al.*, 2002; Vidal *et al.*, 2003*a*), control (*e.g.*, Branicky *et al.*, 1998; Bemporad and Morari, 1999; Lygeros *et al.*, 1999), verification (*e.g.*, Bemporad *et al.*, 2000*a*; Chutinan and Krogh, 2003), and fault detection (*e.g.*, Bemporad *et al.*, 1999; Lunze, 2000) have been also addressed. The related tools strongly depend on the adopted modelling framework. Equivalence between PWA systems and several other classes of hybrid systems has been shown in (Bemporad *et al.*, 2000*b*; Heemels *et al.*, 2001; Sontag, 1996). The importance of these equivalence results is twofold. First, they allow for transferring theoretical properties and tools from PWA systems (*e.g.*, stability criteria were proposed for PWA systems by Johansson and Rantzer) to the other classes, and vice versa. Second, they enable the use of PWA system identification techniques to obtain general hybrid models.

The discussion in this section motivates the importance of PWA systems, and of the related identification methods. Several different approaches that are applicable, or at least related, to the PWA system identification problem, can be found in the literature. An overview will be given in Section 2.3. A novel approach to PWA system identification is the main contribution of this thesis, and will be described in Chapter 3.

1.2 System identification

The system identification problem is the problem of constructing mathematical models of dynamical systems based on observed data from the system. In most settings, one is interested in finding the relation between the input and the output signals of the system. The main ingredients for the system identification problem are the following (Ljung, 1999):

1. An experiment, providing the data set to be used for estimation.
2. A set of candidate models (a *model structure*).
3. An identification method, for fitting the model structure to data.
4. Routines to validate and accept the identified model(s).

The experiment should be designed so as to obtain a data set that is as informative as possible for constructing a good model. This issue is not covered in the thesis, and the interested reader is referred to (Ljung, 1999). Section 1.2.1 deals with model structures. The selection of a suitable model structure is a crucial point in the system identification procedure, that could benefit of prior knowledge or physical insight about the system, if available, as well as of past experience. In Section 1.2.2, the problem of fitting a given model structure to measured data will be addressed by introducing a common family of identification methods, namely the prediction error methods. Model validation will be the subject of Section 1.4.

1.2.1 Black-box model structures

Consider a discrete-time dynamical system with input $\mathbf{u}_k \in \mathbb{R}^p$ and output $y_k \in \mathbb{R}$, where $k \in \mathbb{Z}$ is the time index. Let

$$\begin{aligned} \mathbf{u}^{k-1} &= [\,\mathbf{u}'_{k-1}\,\mathbf{u}'_{k-2}\,\cdots\,]' \\ \mathbf{y}^{k-1} &= [\,y_{k-1}\,y_{k-2}\,\cdots\,]' \end{aligned}$$

be, respectively, past inputs and outputs up to time $k-1$. The considered system identification problem consists in finding a relationship between past observations $[\mathbf{u}^{k-1}, \mathbf{y}^{k-1}]$ and future outputs y_k:

$$y_k = g(\mathbf{u}^{k-1}, \mathbf{y}^{k-1}) + \varepsilon_k \tag{1.1}$$

The additive term ε_k in (1.1) accounts for the fact that the next output y_k will not be an exact function of past data. When solving the problem, a goal must be that ε_k is small, so that one can think of $g(\mathbf{u}^{k-1}, \mathbf{y}^{k-1})$ as a good prediction of y_k given past observed data. The search for a suitable function g is typically restricted within a specified set of candidate models, the so-called *model set*. The choice of the model set is crucial in the system identification process. It is here that prior knowledge and engineering intuition should be combined with formal properties of the models. When no physical insight about the system is available or used, *black-box* models must be employed. The choice of the model set, in this case, is among families that possess good flexibility, or have been "successful" in the past.

The model set is often parameterized by a finite-dimensional parameter vector $\theta \in \Theta \subseteq \mathbb{R}^d$. In this case, the function g in (1.1) is written as follows:

$$g(\mathbf{u}^{k-1}, \mathbf{y}^{k-1}, \theta) \tag{1.2}$$

in order to make the dependance on θ explicit. The parameterized mapping (1.2) is called a *model structure*. Linear models (*e.g.*, ARX, ARMAX and OE models) are no doubt the most common class of parameterized models used in system identification. They are also known as *ready-made* models because, provided the model order, standard techniques can be easily applied to fit a model to the data. However,

linear models are not always expressive enough to describe accurately the dynamics of a system. Hence, considerable interest has been devoted to nonlinear black-box structures, *i.e.*, model structures that are prepared to describe virtually any nonlinear dynamics. Model structures based, *e.g.*, on neural networks, wavelet networks, radial basis networks, and hinging hyperplanes have been proposed; see the survey papers (Sjöberg *et al.*, 1995; Juditsky *et al.*, 1995) and references therein. Piecewise affine (PWA) models, that are considered in this thesis and will be formally introduced in Section 2.1, are also nonlinear black-box structures, thanks to the universal approximation properties of piecewise affine maps.

Although being general, the form (1.2) is not convenient in practice, since $\mathbf{u}^{k-1}$ and $\mathbf{y}^{k-1}$ contain infinitely many elements. Hence, it is useful to define first a mapping from past observations $[\mathbf{u}^{k-1}, \mathbf{y}^{k-1}]$ to a fixed-length vector φ_k, and then to concatenate it with the mapping to the output as follows:

$$g(\mathbf{u}^{k-1}, \mathbf{y}^{k-1}, \theta) = g(\varphi_k, \theta) \tag{1.3}$$

The vector $\varphi_k = \varphi(\mathbf{u}^{k-1}, \mathbf{y}^{k-1})$ is called the *regression vector*, and its components are referred to as *regressors*. The choice of the nonlinear mapping (1.2) is thus decomposed into two partial problems: the choice of the regression vector φ_k from past inputs and outputs, and the choice of the nonlinear (parameterized) mapping $g(\varphi, \theta)$ from the regressor set to the output set.

From (1.1) and (1.2), the *predictor* is given by:

$$\hat{y}_{k|\theta} = g(\mathbf{u}^{k-1}, \mathbf{y}^{k-1}, \theta) \tag{1.4}$$

If the error ε_k in (1.1) is assumed to be zero-mean and independent of $[\mathbf{u}^{k-1}, \mathbf{y}^{k-1}]$, then (1.4) is the best guess of the output, obtained by replacing ε_k with its expected value. The notation $\hat{y}_{k|\theta}$ is used for the predicted output to emphasize that its calculation depends on the parameter vector θ that parameterizes the model structure.

Example 1.1 The common ARX models are defined by the following structure, where $u_k \in \mathbb{R}$, $y_k \in \mathbb{R}$, and n_a and n_b are fixed orders:

$$y_k = \sum_{i=1}^{n_a} a_i y_{k-i} + \sum_{j=1}^{n_b} b_j u_{k-j} + \varepsilon_k \tag{1.5}$$

By letting:

$$\theta = [\, a_1 \ \dots \ a_{n_a} \ \ b_1 \ \dots \ b_{n_b} \,]' \tag{1.6}$$

$$\varphi_k = [\, y_{k-1} \ \dots \ y_{k-n_a} \ \ u_{k-1} \ \dots \ u_{k-n_b} \,]' \tag{1.7}$$

the predictor corresponding to (1.5) can be written in compact form as follows:

$$\hat{y}_{k|\theta} = \varphi_k' \theta \tag{1.8}$$

Model structures like (1.8), that are linear in θ, are known as *linear regressions*. Note that, in the linear regression (1.8), the regression vector φ_k is not necessarily needed to be of the form (1.7) or to contain the raw measurements. Rather, it could be any nonlinear function of past inputs and outputs, and contain filtered data. For instance, in this thesis, a simple extension of the standard regression vector (1.7) will be considered, with the last entry equal to 1:

$$\varphi_k = [\, y_{k-1} \ \dots \ y_{k-n_a} \ \ u_{k-1} \ \dots \ u_{k-n_b} \ 1 \,]'$$

It allows for an additive constant term c in (1.5), so that θ in this case becomes:

$$\theta = [\, a_1 \ \dots \ a_{n_a} \ \ b_1 \ \dots \ b_{n_b} \ c \,]'$$

Model (1.5) with an additive constant term c is called an *affine* ARX model.

1.2.2 Prediction error methods

As shown in the previous section, a model parameterization leads to the predictor (1.4), that depends on past data $[\mathbf{u}^{k-1}, \mathbf{y}^{k-1}]$ and the unknown parameter vector θ. A method to determine a suitable value of θ, based on the information contained in the measured data set, is called an *identification method*.

Since the essence of a model is its prediction capability, it is natural to judge its performance in this respect. Thus, let the sequence of the *prediction errors* be given by:

$$\varepsilon_{k|\theta} = y_k - \hat{y}_{k|\theta}, \quad k = 1, \dots, N \tag{1.9}$$

where N is the number of data. A model is said to be "good" if it produces "small" prediction errors (1.9) when applied to the observed data. The question is how to qualify what "small" should mean. A common approach is to define the following criterion function:

$$V_N(\theta) = \frac{1}{N}\sum_{k=1}^{N} \ell(\varepsilon_{k|\theta}) \tag{1.10}$$

where ℓ is a scalar valued, nonnegative function that is used to measure the "size" of the prediction error, *e.g.*, $\ell(\varepsilon) = \varepsilon^2$ or $\ell(\varepsilon) = |\varepsilon|$. For simpler notation, the dependance of V_N on the data set has been omitted. Then, a model is fitted to the data by minimizing $V_N(\theta)$, *i.e.*, by taking the estimate $\hat{\theta}$ given by:

$$\hat{\theta} = \arg\min_{\theta\in\Theta} V_N(\theta) \tag{1.11}$$

This procedure can be thought of as "curve fitting" between y_k and $\hat{y}_{k|\theta}$. The underlying principle is independent of the particular model parameterization used, although this will affect the actual minimization. Indeed, except very special cases such as in Example 1.2, no analytic solution for (1.11) is possible, so that minimization must be carried out by resorting to numerical algorithms. A fundamental problem is that $V_N(\theta)$ might have local (nonglobal) minima, where local search algorithms could get caught.

The way of estimating θ given by (1.11) leads to many commonly used *parametric* identification methods, that are known under the general term of *prediction error methods*. Particular methods, with specific names (*e.g.*, the least squares and the maximum likelihood methods), are obtained as special cases of (1.11), depending on the choice of the norm ℓ, the choice of the model structure, and in some cases the choice of the method by which the minimization is accomplished.

Example 1.2 When considering the linear regression (1.8), the prediction error becomes $\varepsilon_{k|\theta} = y_k - \varphi_k'\theta$, and the criterion function (1.10), with $\ell(\varepsilon) = \varepsilon^2$, is:

$$V_N(\theta) = \frac{1}{N}\sum_{k=1}^{N} (y_k - \varphi_k'\theta)^2 \tag{1.12}$$

This is the *least-squares criterion* for the linear regression (1.8). Since V_N is a quadratic function of θ, it can be minimized analytically. The minimization gives,

provided the indicated inverse exists:

$$\hat{\theta}^{LS} = \left[\sum_{k=1}^{N} \varphi_k\varphi_k'\right]^{-1} \sum_{k=1}^{N} \varphi_k y_k \tag{1.13}$$

The estimate (1.13) is the very well-known *least squares estimate*.

1.3 Set-membership identification

Estimation theory deals with the problem of evaluating some unknown variables based on given data. Available data are typically known with some uncertainty and it is necessary to evaluate how this uncertainty affects the estimated variables. A general framework for estimation problems is as follows:

> *Given an unknown element $\mathbf{x} \in \mathscr{X}$, find an estimate of the function $S(\mathbf{x}) \in \mathscr{Z}$, based on a priori information $\mathscr{K} \subseteq \mathscr{X}$ and on measurements of the function $F(\mathbf{x}) \in \mathscr{Y}$ corrupted by additive noise $\mathbf{e}$.*

Here, $\mathscr{X}$ is the problem element space, $\mathscr{Y}$ is the measurement space, and $\mathscr{Z}$ is the solution space. Accordingly, $\mathbf{x}$ is called the problem element and $\mathbf{z} = S(\mathbf{x})$ is called the problem solution. The information available for estimation is twofold:

a priori: $\quad \mathbf{x} \in \mathscr{K} \subseteq \mathscr{X}$

a posteriori (measurements): $\quad \mathbf{y} = F(\mathbf{x}) + \mathbf{e}, \quad F : \mathscr{X} \to \mathscr{Y}$

An *estimation algorithm* (or *estimator*) Φ is a mapping from the measurement to the solution space, *i.e.*, $\Phi : \mathscr{Y} \to \mathscr{Z}$, which provides an estimate $\Phi(\mathbf{y})$ of the problem solution $S(\mathbf{x})$ based on the available information. A schematic representation of the generic estimation problem is shown in Figure 1.1.

In an estimation problem, the goal must be that $\Phi(\mathbf{y})$ is as close to $S(\mathbf{x})$ as possible. Since observed data are affected by errors, it is important to evaluate the effect of the uncertainties on the quality of the estimate. This depends on the type of assumptions made on the uncertainty. In classical estimation theory, data are commonly assumed to be corrupted by additive random noise with (partially) known

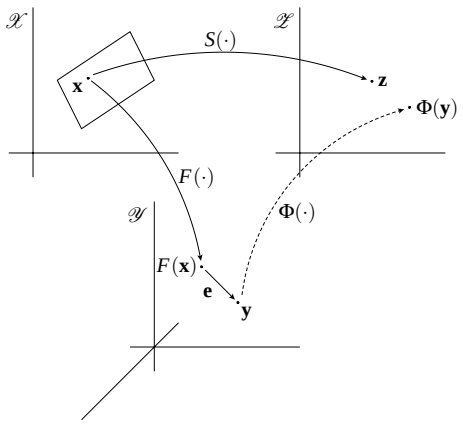


Figure 1.1 A schematic representation of the generic estimation problem

probability density function. However, in situations where the random nature of the uncertainty is questionable, an alternative approach can be used, which does not rely on any statistical assumption on the uncertainty. This approach, which is referred to as *set-membership* or *unknown but bounded* (UBB) error description, characterizes the uncertainty by means of additive noise that is known only to have given bounds, *i.e.*, the noise term $\mathbf{e}$ is assumed to satisfy:

$$\|\mathbf{e}\|_Y \leq \delta \tag{1.14}$$

for some given bound $\delta > 0$. In (1.14), $\|\cdot\|_Y$ denotes the norm used in the $\mathscr{Y}$-space. Under the UBB error description, the estimation problem has different characteristics from the stochastic case. The stochastic approach provides *confidence regions*, to which a given probability of containing the problem solution is associated. Vice versa, in the set-membership approach, the interest is focused on the *feasible solution set*, *i.e.*, the set of all the estimates that are consistent with the available information on the problem.

Widely used classes of (pointwise) estimators in set-membership estimation are the *central* algorithms and the *projection* algorithms. Let the feasible solution

set be defined as follows:

$$FSS_{\mathbf{y}} = \left\{ \mathbf{z} = S(\mathbf{x}) \,\middle|\, \|\mathbf{y} - F(\mathbf{x})\|_Y \leq \delta,\ \mathbf{x} \in \mathscr{K} \right\} \tag{1.15}$$

The first class of algorithms is related to the idea of taking as an estimate the "center" of $FSS_{\mathbf{y}}$, namely the element in $\mathscr{Z}$ that minimizes the maximum distance to the elements in $FSS_{\mathbf{y}}$. By denoting the norm used in the $\mathscr{Z}$-space as $\|\cdot\|_Z$, a *central estimator* is hence given by:

$$\Phi_c(\mathbf{y}) = \arg \inf_{\mathbf{z} \in \mathscr{Z}} \sup_{\tilde{\mathbf{z}} \in FSS_{\mathbf{y}}} \|\tilde{\mathbf{z}} - \mathbf{z}\|_Z \tag{1.16}$$

The second class of algorithms takes as an estimate the solution corresponding to the problem element that minimizes the norm of the measurement error. A *projection estimator* is hence given by:

$$\Phi_p(\mathbf{y}) = S(\mathbf{x}_p), \quad \mathbf{x}_p = \arg \min_{\mathbf{x} \in \mathscr{K}} \|\mathbf{y} - F(\mathbf{x})\|_Y \tag{1.17}$$

Note that the computation of (1.17) does not require the use of the bound δ. The projection estimators enjoy interesting properties over the central estimators. They are *interpolatory* algorithms, *i.e.*, $\Phi_p(\mathbf{y})$ always belongs to $FSS_{\mathbf{y}}$. This property might be not satisfied by (1.16). Moreover, since the following inequalities hold:

$$\sup_{\mathbf{z} \in FSS_{\mathbf{y}}} \|\mathbf{z} - \Phi_c(\mathbf{y})\|_Z \leq \sup_{\mathbf{z} \in FSS_{\mathbf{y}}} \left\|\mathbf{z} - \Phi_p(\mathbf{y})\right\|_Z \leq 2 \sup_{\mathbf{z} \in FSS_{\mathbf{y}}} \|\mathbf{z} - \Phi_c(\mathbf{y})\|_Z$$

they are "robust" with respect to inexact knowledge of the uncertainty bound δ, *i.e.*, with respect to $FSS_{\mathbf{y}}$. Typically, it is rather difficult to know exactly the value of δ, whereas it is likely to estimate an upper bound.

In the following, the set-membership approach will be applied to the parametric identification of the linear regression:

$$y_k = \varphi_k' \theta + \varepsilon_k \tag{1.18}$$

already considered in Examples 1.1 and 1.2. Given N data points (y_k, φ_k), $k = 1, \ldots, N$, the problem of estimating θ based on the available data can be cast in

the general estimation framework defined in this section, by considering the problem element $\mathbf{x} = \theta$, the measurement vector $\mathbf{y} = [\, y_1 \ \ldots \ y_N \,]'$, the error vector $\mathbf{e} = [\, \varepsilon_1 \ \ldots \ \varepsilon_N \,]'$, and the information operator:

$$F(\theta) = \begin{bmatrix} \varphi_1' \\ \vdots \\ \varphi_N' \end{bmatrix} \theta$$

Moreover, the solution operator S is the identity map, *i.e.*, $S(\theta) = \theta$. The error vector $\mathbf{e}$ is assumed to be bounded according to (1.14). The feasible solution set (1.15) is, in this case, the set of all parameter vectors θ that are consistent with the measurements and the error bound. Accordingly, it is called the *Feasible Parameter Set* (*FPS*). Provided that the predictor for a linear regression is given by (1.8), the *FPS* can be used to predict an interval of values $[\underline{\hat{y}_k}, \overline{\hat{y}_k}]$ for the output by computing:

$$\overline{\hat{y}_k} = \sup_{\theta \in FPS} \varphi_k'\theta \quad \text{and} \quad \underline{\hat{y}_k} = \inf_{\theta \in FPS} \varphi_k'\theta \tag{1.19}$$

In this thesis, the norm considered in the $\mathscr{Y}$-space is the ℓ_∞-norm. This corresponds to bound the error vector componentwise by a given $\delta > 0$, *i.e.*:

$$|\varepsilon_k| \leq \delta, \quad k = 1, \ldots, N \tag{1.20}$$

By assuming (1.20), the *FPS* is a convex polytope, described by:

$$FPS = \left\{ \theta \in \mathbb{R}^d \,\middle|\, |y_k - \varphi_k'\theta| \leq \delta,\ k = 1, \ldots, N \right\} \tag{1.21}$$

Hence, the computation of (1.19) amounts to solve two linear programs. Since the complexity of the *FPS* grows with N, and its exact description may become computationally demanding, a key issue in set-membership identification is the approximation of (1.21) by means of simply shaped regions. In the literature, several set approximation techniques have been proposed to get around this problem. They are based on over- or underbounding the polytope by simpler regions such as ellipsoids (Fogel and Huang, 1982), orthotopes (Pearson, 1988), limited complexity polytopes (Broman and Shensa, 1990; Piet-Lahanier and Walter, 1993; Veres, 1994), or parallelotopes (Vicino and Zappa, 1996; Chisci *et al.*, 1998).

The ℓ_p central estimator for (1.18), $p \in [1, \infty]$, is given by:

$$\hat{\theta}^c = \arg\min_{\theta} \max_{\tilde{\theta} \in FPS} \|\tilde{\theta} - \theta\|_p \tag{1.22}$$

If the ℓ_∞-norm is considered both in the $\mathscr{Y}$-space and the $\mathscr{Z}$-space, it has been shown by Milanese and Tempo (1985) that $\hat{\theta}^c$ can be computed as follows:

$$\hat{\vartheta}_i^c = \frac{\overline{\vartheta_i} + \underline{\vartheta_i}}{2}, \quad i = 1, \ldots, d \tag{1.23}$$

where:

$$\overline{\vartheta_i} = \max_{\theta \in FPS} \vartheta_i \quad \text{and} \quad \underline{\vartheta_i} = \min_{\theta \in FPS} \vartheta_i \tag{1.24}$$

Since *FPS* is given by the polytope (1.21), the computation of $\hat{\theta}^c$ by (1.23) and (1.24) amounts to solve $2d$ linear programs. Note that, in this case, $\hat{\theta}^c$ is the center of the smallest axis aligned box which contains (1.21).

If the ℓ_∞-norm is chosen in the $\mathscr{Y}$-space, the ℓ_∞ projection estimator for (1.18), which will be widely used in this thesis, is given by:

$$\hat{\theta}^p = \arg\min_{\theta} \|\mathbf{y} - F(\theta)\|_\infty = \arg\min_{\theta} \max_{k=1,\ldots,N} \left| y_k - \varphi_k'\theta \right| \tag{1.25}$$

The estimate (1.25) can be computed by solving the following linear program:

$$\begin{cases} \min\limits_{\theta, \hat{\delta}} & \hat{\delta} \\ s.t. & \varphi_k'\theta - \hat{\delta} \leq y_k \\ & \varphi_k'\theta + \hat{\delta} \geq y_k \quad k = 1, \ldots, N \end{cases} \tag{1.26}$$

Note that problem (1.26) corresponds to finding the minimum bound for which the resulting *FPS* is nonempty. It is interesting to note that, if the ℓ_2-norm is used in the $\mathscr{Y}$-space, the ℓ_2 projection estimator for (1.18) is given by:

$$\hat{\theta}^p = \arg\min_{\theta} \|\mathbf{y} - F(\theta)\|_2 = \arg\min_{\theta} \sqrt{\sum_{k=1}^{N} (y_k - \varphi_k'\theta)^2}$$

which coincides with the least squares estimate (1.13) obtained by minimizing the criterion function (1.12).

The set-membership approach has been pioneered by the work of Witsenhausen (1968) and Schweppe (1968) on state estimation problems for dynamical

systems. An overview of this approach up to the early 90's is given in (Milanese and Vicino, 1991). A short introduction, oriented to the system identification problem, can be found in (Ninness and Goodwin, 1995). See also the collection of papers (Milanese *et al.*, 1996).

1.4 Model validation

After having identified a model of the system, it remains to test the "goodness" of the model itself. *Model validation* deals with this problem. It involves various tests to assess how the model relates to observed data, to prior knowledge, and/or to its intended use. Deficient model behavior in these respects will make one reject the model, whereas good performance will develop a certain confidence in it. Indeed, the aim is never to accept a model as being true or correct, rather to discard the obviously incorrect ones. It is however important to stress the subjective ingredient in model validation. Model validation tools should only be viewed as advisors to the user. It is the user that finally makes the decision based on several tests.

In system identification, the most natural entity with which the model must be compared, are the data themselves. The quality of the model in this respect is checked by quantifying the agreement between the model and the measured data from the system. The prime method is to investigate how well the model reproduces the behavior of a new set of data (the *validation data*) that was not used to fit the model. The model is simulated with a new input, and the simulated output is then compared with the measured output corresponding to the same input. Numerical measurements of fit or visual inspection can be used to decide whether the fit is good enough. A suitable indicator, that will be used in this thesis, is the percentage of *Variation Accounted For* (*VAF*). Let $\mathbf{y} = (y_1, \ldots, y_N)$ be the vector of system outputs, $\bar{y}$ the average of $\mathbf{y}$, and $\hat{\mathbf{y}} = (\hat{y}_1, \ldots, \hat{y}_N)$ the vector of simulated outputs. The *VAF* indicator is defined as follows (Ljung, 2003):

$$VAF = 100 \cdot \left(1 - \frac{\|\hat{\mathbf{y}} - \mathbf{y}\|_2}{\|\mathbf{y} - \bar{\mathbf{y}}\|_2}\right) \tag{1.27}$$

It can be interpreted as the percentage of the output variation that is explained by the model. For given $\mathbf{y}$, the smaller is the error $\hat{\mathbf{y}} - \mathbf{y}$, the closer is *VAF* to 100%.

The second basic method for model validation is to examine the *residuals*, *i.e.*, what the model could not "explain". These are the prediction errors (1.9), *i.e.*:

$$\varepsilon_k = y_k - \hat{y}_k, \quad k = 1, \ldots, N$$

Most classical model validation tests are based on *residual analysis*. A simple starting point is to compute basic statistics for the residuals, *e.g.*:

- The maximum absolute value: $S_1 = \max_{k=1,\ldots,N} |\varepsilon_k|$
- The average quadratic error: $S_2^2 = \frac{1}{N} \sum_{k=1}^{N} \varepsilon_k^2$

Thresholds can be fixed to decide whether the above quantities are too large or not. They are determined on the basis of prior knowledge, assumptions (*e.g.*, the bound δ on S_1 considered in the set-membership approach; see Section 1.3), or some ad-hoc procedure. However, only such an analysis is not sufficient, in general. It is always good practice to check the residuals for dependencies. Ideally, the residual at time k should be independent of information that was at hand at time $k-1$. For instance, if the residuals ε_k and the inputs $u_{k-\tau}$ are correlated, then there is a part of y_k that originates from $u_{k-\tau}$ and that has been not properly accounted for by $\hat{y}_k$. One can conclude that the model has not extracted all the relevant information about the system from the data, and hence it could be improved. Two tests for checking dependencies of the residuals will be described in the following sections.

1.4.1 A whiteness test

If correlation among the residuals shows up, then part of ε_k could have been predicted from past data. This means that y_k could have been better predicted, which in turn is a sign of deficiency of the model.

Information about the correlation among the residuals is carried by the *autocorrelation* sequence, which is defined as follows:

$$\hat{R}_\varepsilon^N(\tau) = \frac{1}{N} \sum_{k=1}^{N} \varepsilon_k \varepsilon_{k-\tau}, \quad \tau = 0, 1, \ldots, M \tag{1.28}$$

For easier notation, in (1.28) it is assumed that $N+M$ residuals are available, indexed from $1-M$ to N. Investigation of which requirements should be associated with (1.28), can be performed in the context of statistical hypothesis testing. By assuming $\{\varepsilon_k\}$ to be a white noise sequence with zero mean and variance σ_ε^2, it can be shown that (Ljung, 1999):

$$\frac{\sqrt{N}}{\sigma_\varepsilon^2}\hat{R}_\varepsilon^N(\tau) \sim \mathcal{N}(0,1) \quad \text{asymptotically as } N \to \infty$$

Hence, a good whiteness test is to verify if:

$$\frac{\left|\hat{R}_\varepsilon^N(\tau)\right|}{\hat{R}_\varepsilon^N(0)} \leq \frac{x_\alpha}{\sqrt{N}}, \quad \tau = 1,\ldots,M \tag{1.29}$$

where $x_\alpha > 0$ corresponds to the α level of the $\mathcal{N}(0,1)$ distribution, *i.e.*:

$$\int_{-x_\alpha}^{x_\alpha} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx = 1-\alpha$$

For instance, $x_\alpha = 2.58$ for $\alpha = 0.01$. Note that in (1.29) σ_ε^2 has been replaced with its estimate $\hat{R}_\varepsilon^N(0)$. Test (1.29) can be performed graphically by plotting $\hat{R}_\varepsilon^N(\tau)$ as a function of τ, and the confidence limits as horizontal lines. A graphical test based on (1.29) is shown in Example 1.3.

1.4.2 A cross-correlation test

It is of special importance that the residuals do not depend on the particular input used in the data set. The quality of the model might otherwise change when different inputs are considered. In order to check this kind of dependency, it is suitable to study the *cross-correlation* sequence of ε_k and u_k, which is defined as follows:

$$R_{\varepsilon u}^N(\tau) = \frac{1}{N}\sum_{k=1}^{N} \varepsilon_k u_{k-\tau}, \quad -M \leq \tau \leq M \tag{1.30}$$

Another reason for considering (1.30) is that, if there are traces of past inputs in the residuals, then there is a part of y_k that originates from the past inputs and that has been not properly picked up by the model. In (1.30) it is assumed that $N+2M$ inputs are available, indexed from $1-M$ to $N+M$.

Investigation of which requirements should be associated with (1.30), can be again performed in the context of statistical hypothesis testing. By assuming $\{\varepsilon_k\}$ and $\{u_k\}$ to be independent sequences, it can be shown that (Ljung, 1999):

$$\frac{\sqrt{N}}{\sigma_P}\hat{R}_{\varepsilon u}^N(\tau) \sim \mathcal{N}(0,1) \quad \text{asymptotically as } N \to \infty$$

where:

$$\sigma_P^2 = \sum_{k=-\infty}^{+\infty} R_\varepsilon(k)R_u(k), \quad R_\varepsilon(\tau) = E[\varepsilon_k\varepsilon_{k-\tau}], \quad R_u(\tau) = \lim_{N\to\infty}\frac{1}{N}\sum_{k=1}^{N} u_k u_{k-\tau}$$

Hence, a good cross-correlation test is to verify if:

$$\frac{\left|\hat{R}_{\varepsilon u}^N(\tau)\right|}{\hat{\sigma}_P} \leq \frac{x_\alpha}{\sqrt{N}}, \quad -M \leq \tau \leq M \tag{1.31}$$

where $x_\alpha > 0$ corresponds to the α level of the $\mathcal{N}(0,1)$ distribution, as in Section 1.4.1. Note that σ_P^2 has been replaced with its estimate:

$$\hat{\sigma}_P^2 = \sum_{k=-M}^{M} \hat{R}_\varepsilon^N(k)\hat{R}_u^N(k)$$

Test (1.31) can be performed graphically by plotting $\hat{R}_{\varepsilon u}^N(\tau)$ as a function of τ, and the confidence limits as horizontal lines. A graphical test based on (1.31) is shown in Example 1.3. It must be required for a good model that $\hat{R}_{\varepsilon u}^N(\tau)$ does not go significantly outside the confidence region. A peak at lag $\tau > 0$ shows that the effect of $u_{k-\tau}$ on y_k is not properly described, *e.g.*, because the time delay of the system was overestimated. A peak at lag $\tau < 0$ does not imply that the model structure is deficient, rather that output feedback in the input is present.

Example 1.3 A graphical model validation test is illustrated in Figure 1.2. The upper plot shows the whiteness test (1.29), and the lower plot shows the cross-correlation test (1.31). Dashed lines denote the confidence intervals for $\alpha = 0.01$. The model and the true system are irrelevant for this discussion. Since both plots are inside the bounds, there is no evidence of the residuals being non-white, nor there is some significant cross-correlation between the residuals and the input. The model is therefore not falsified by this model validation test.

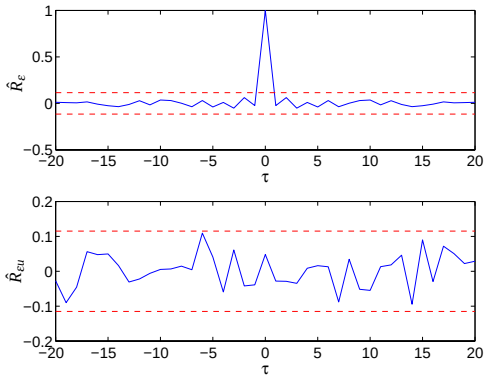


Figure 1.2 Results from a model validation test. The upper plot shows a whiteness test of the residuals and the lower plot shows a cross-correlation test between the residuals and the input

1.5 Thesis outline

The thesis is structured as follows. Chapter 2 gives an introduction to piecewise affine (PWA) systems and to the identification of PWA models. It also contains an overview of different approaches to PWA system identification. Chapter 3 describes the main contribution of the thesis, consisting in a procedure for data point classification and parameter estimation of PWA models based on a MIN PFS formulation. Chapter 4 gives an overview of several techniques for two-class and multi-class linear separation. These techniques are then applied to the problem of region estimation for the identified PWA model. In Chapter 5, the performance of the proposed identification procedure is tested on both numerical examples and a case study. Lastly, conclusions are drawn in Chapter 6, and guidelines for future research are suggested.

1.6 Contributions

The main contributions of this thesis are:

- A procedure for data point classification and parameter estimation of piecewise affine models based on a MIN PFS formulation. This is found in Chapter 3. The proposed approach allows to trade off between the complexity and the accuracy of the identified PWA model by selecting a bound on the fitting error.

- Improvements of the greedy algorithm (Amaldi and Mattavelli, 2002) for the MIN PFS problem with complementary inequalities. The proposed modifications, that are described in Sections 3.2.1 and 3.2.2, allow to obtain a number of feasible subsystems which is closer to be minimal.

- A wide overview of several techniques for two-class and multi-class linear separation. This is found in Chapter 4.

Some of the material in Chapter 3 has been previously published in:

A. Bemporad, A. Garulli, S. Paoletti and A. Vicino (2003). A greedy approach to identification of piecewise affine models. In: *Hybrid Systems: Computation and Control* (O. Maler and A. Pnueli, Eds.). pp. 97–112. Lecture Notes in Computer Science. Springer Verlag.

A. Bemporad, A. Garulli, S. Paoletti and A. Vicino (2003). Set membership identification of piecewise affine models. In: *Proc. 13th IFAC Symposium on System Identification*. Rotterdam, The Netherlands. pp. 1826–1831.

Other parts of Chapter 3 and the case study in Chapter 5 also appear in:

A. Bemporad, A. Garulli, S. Paoletti and A. Vicino (2004). Data classification and parameter estimation for the identification of piecewise affine models. Submitted to *43rd IEEE Conference on Decision and Control*.

2

PWA System Identification

This chapter gives an introduction to piecewise affine (PWA) systems, and to the parametric identification of PWA models. Section 2.3 contains an overview and a classification of different approaches to PWA system identification (Roll, 2003). The identification procedure proposed in this thesis belongs to the category of algorithms which tackle the problem in two steps. First, data are classified and the parameters of the submodels are identified, and then the regions are estimated.

2.1 Piecewise affine systems

In this section, discrete time piecewise affine (PWA) systems both in state space and in regression form are introduced[1]. PWA systems are defined as collections of linear/affine systems with the same continuous state, connected by switches that are determined by a polyhedral partition of the state+input set (Sontag, 1981). They can be used to model a large number of physical processes, and are suitable to approximate nonlinear dynamics, *e.g.*, via multiple linearizations at different operating points. In addition, PWA systems are equivalent to several classes of hybrid systems, and can therefore be used to describe systems exhibiting hybrid structure.

[1] For simplicity of notation, noiseless systems are considered here. The noise will be included later as an additive term.

2.1.1 Systems in state space form

A general discrete time piecewise affine system in *state space* form is described by the following equations:

$$\begin{aligned} \mathbf{x}_{k+1} &= A_{\sigma(k)}\mathbf{x}_k + B_{\sigma(k)}\mathbf{u}_k + \mathbf{b}_{\sigma(k)} \\ \mathbf{y}_k &= C_{\sigma(k)}\mathbf{x}_k + D_{\sigma(k)}\mathbf{u}_k + \mathbf{d}_{\sigma(k)} \end{aligned} \tag{2.1}$$

where $k \in \mathbb{Z}$ is time, $\mathbf{x}_k \in \mathbb{R}^n$ is the (*continuous*) state, $\mathbf{u}_k \in \mathbb{R}^p$ is the input, and $\mathbf{y}_k \in \mathbb{R}^q$ is the output. The *discrete* mode $\sigma(k)$, describing in what affine subsystem the system is at time k, is assumed to take only a finite number of values. Without loss of generality, $\sigma(k) \in \{1,\dots,s\}$, where s is the number of affine subsystems. $\sigma(k)$ could be a function of k, $\mathbf{x}_k$, $\mathbf{u}_k$, or some other external input. A_i, B_i, $\mathbf{b}_i$, C_i, D_i and $\mathbf{d}_i$, $i = 1,\dots,s$, are real matrices and vectors with suitable dimensions that describe each affine subsystem. Hence, system (2.1) can be seen as a collection of affine systems with continuous state $\mathbf{x}_k$, connected by switches that are indexed by the discrete mode $\sigma(k)$. The evolution of the discrete mode can be described in a variety of ways. In *Jump Linear* (JL) systems, $\sigma(k)$ is an unknown, deterministic and finite-valued input. In *Jump-Markov Linear* (JML) systems, the dynamics of $\sigma(k)$ is modelled as an irreducible Markov chain governed by the transition probabilities $\pi(i,j) \triangleq P\big(\sigma(k+1) = j \,\big|\, \sigma(k) = i\big)$. In *PieceWise Affine* (PWA) systems (Sontag, 1981; Bemporad *et al.*, 2000*b*), $\sigma(k)$ is given by:

$$\sigma(k) = i \quad \text{if} \quad (\mathbf{x}_k, \mathbf{u}_k) \in \Omega_i, \quad i = 1,\dots,s \tag{2.2}$$

where $\{\Omega_i\}_{i=1}^s$ is a complete partition[2] of the state+input set $\Omega \subseteq \mathbb{R}^n \times \mathbb{R}^p$ where (2.1) is valid, and each region Ω_i is a convex polyhedron defined as follows:

$$\Omega_i = \big\{(\mathbf{x},\mathbf{u}) \in \mathbb{R}^n \times \mathbb{R}^p \,\big|\, \bar{H}_i\mathbf{x} + \bar{J}_i\mathbf{u} + \mathbf{g}_i \preceq \mathbf{0}\big\} \tag{2.3}$$

with $\bar{H}_i \in \mathbb{R}^{q_i \times n}$, $\bar{J}_i \in \mathbb{R}^{q_i \times p}$, and $\mathbf{g}_i \in \mathbb{R}^{q_i}$, $i = 1,\dots,s$. If the vectors $\mathbf{b}_i$ and $\mathbf{d}_i$ are zero for all $i = 1,\dots,s$, system (2.1) is referred to as *PieceWise Linear* (PWL). From a complexity point of view, PWL and PWA systems are equivalent ($\mathbf{b}_i$ and $\mathbf{d}_i$ can be thought of as generated by integrators with no input).

[2] A collection of sets $\{\mathscr{A}_i\}_{i=1}^s$ is said to be a (complete) partition of $\mathscr{A} \subseteq \mathbb{R}^m$ if $\bigcup_{i=1}^s \mathscr{A}_i = \mathscr{A}$ and $\mathscr{A}_i \bigcap \mathscr{A}_j = \emptyset$, $\forall\, i \neq j$.

Remark 2.1 As introduced in Section 1.1, PWA systems (Sontag, 1981) form a special class of *hybrid* systems. Other descriptions for hybrid systems include *Mixed Logical Dynamical* (MLD) systems (Bemporad and Morari, 1999), *Linear Complementarity* (LC) systems (Van der Schaft and Schumacher, 1998; Heemels *et al.*, 2000), *Extended Linear Complementarity* (ELC) systems (De Schutter, 2000), and *Max-Min-Plus-Scaling* (MMPS) systems (De Schutter and Van den Boom, 2001). Equivalences among these five classes of systems are shown in (Bemporad *et al.*, 2000*b*; Heemels *et al.*, 2001). Such results are very important for transferring theoretical properties and tools (*e.g.*, control and identification techniques) from one class to another, as they imply that one can choose the most convenient hybrid modelling framework for the study of a particular hybrid system. □

2.1.2 Systems in regression form

A switching system in *regression form* is described by the equation:

$$y_k = \varphi_k'\theta_{\sigma(k)} \tag{2.4}$$

where $\varphi_k \in \mathbb{R}^d$ is the regression vector, $y_k \in \mathbb{R}$ is the output, $\sigma(k) \in \{1,\dots,s\}$ is the discrete mode, and s is the number of subsystems. $\theta_i \in \mathbb{R}^d$, $i = 1,\dots,s$, are the parameter vectors defining each subsystem.

The regression vector φ_k could, for instance, be any function of past inputs and outputs. In the following, the focus will be on systems (2.4) where φ_k is formed as follows:

$$\varphi_k = [\, y_{k-1} \,\dots\, y_{k-n_a}\, \mathbf{u}_{k-1}' \,\dots\, \mathbf{u}_{k-n_b}'\, 1\,]' \tag{2.5}$$

and $\mathbf{u}_k \in \mathbb{R}^p$ is the input to the system. Such systems represent a subclass of the piecewise affine systems described by (2.1), and can be easily transformed into that form by defining the state vector as:

$$\mathbf{x}_k = [\, y_{k-1} \,\dots\, y_{k-n_a}\, \mathbf{u}_{k-1}' \,\dots\, \mathbf{u}_{k-n_b}'\,]' \tag{2.6}$$

The last entry of φ_k is set equal to 1 in order to allow for a constant term in equation (2.4). If the constant 1 is omitted in (2.5), so that φ_k coincides with $\mathbf{x}_k$, the system is piecewise linear. In the following, the vector $\mathbf{x}_k$ will be referred to as the (*standard*) regression vector, and φ_k will be called the *extended* regression vector, since it can be written as $\varphi_k = [\,\mathbf{x}_k' \; 1\,]'$. As for the systems in state space form, the evolution of the discrete mode $\sigma(k)$ can be described in a variety of ways. In *PieceWise affine AutoRegressive eXogenous* (PWARX) systems, the switching mechanism is determined by a polyhedral partition of the set $\mathscr{X} \subseteq \mathbb{R}^n$ where (2.4) is valid[3]. This means that for these systems the discrete mode $\sigma(k)$ is given by:

$$\sigma(k) = i \quad \text{if} \quad \mathbf{x}_k \in \mathscr{X}_i, \quad i = 1, \dots, s \tag{2.7}$$

where $\{\mathscr{X}_i\}_{i=1}^s$ is a complete partition of the regressor set $\mathscr{X}$, and each region $\mathscr{X}_i$ is a convex polyhedron represented in the form:

$$\mathscr{X}_i = \left\{ \mathbf{x} \in \mathbb{R}^n \,\middle|\, \bar{H}_i \mathbf{x} + \mathbf{g}_i \preceq \mathbf{0} \right\} \tag{2.8}$$

with $\bar{H}_i \in \mathbb{R}^{q_i \times n}$ and $\mathbf{g}_i \in \mathbb{R}^{q_i}$, $i = 1, \dots, s$. By letting $H_i = [\,\bar{H}_i \; \mathbf{g}_i\,]$, $i = 1, \dots, s$, and by introducing the piecewise affine map $f : \mathscr{X} \to \mathbb{R}$:

$$f(\mathbf{x}) = \begin{cases} \varphi'\theta_1 & \text{if} \quad H_1\varphi \preceq \mathbf{0} \\ \vdots & \vdots \quad , \\ \varphi'\theta_s & \text{if} \quad H_s\varphi \preceq \mathbf{0} \end{cases} \qquad \varphi = [\,\mathbf{x}' \; 1\,]' \tag{2.9}$$

equation (2.4) can be alternatively rewritten as follows:

$$y_k = f(\mathbf{x}_k) \tag{2.10}$$

PWARX systems defined by (2.10), (2.9) and (2.6), can be seen as a collection of ARX systems connected by switches that are determined by a polyhedral partition of the regressor set.

Remark 2.2 The PWA map (2.9) could be discontinuous over the boundaries defined by the polyhedra (2.8). Figure 2.1 shows a discontinuous PWA map of two

[3] In general, the shape of $\mathscr{X}$ will reflect physical constraints on the inputs and the output of the system. For instance, typical constraints on the output may be $|y_k| \leq y_{max}$ or $|y_k - y_{k-1}| \leq \Delta y_{max}$.

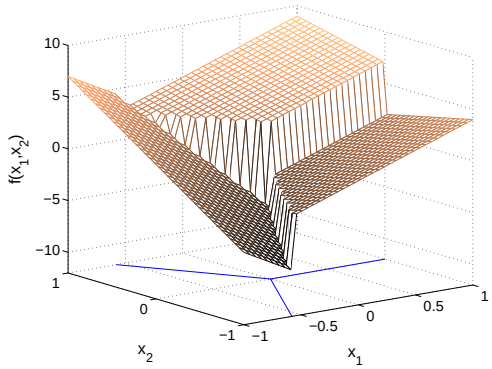


Figure 2.1 Discontinuous PWA map of two variables with $s = 3$ regions

variables. Hence, definition (2.9) is not well posed in general, since the PWA map could be multiply defined over common boundaries of the regions $\mathscr{X}_i$. This issue can be overcome by replacing some of the "$\leq$" inequalities with "$<$" in definition (2.8), but for simplicity of notation it is not addressed here. A similar remark holds also for system (2.1) and definition (2.3). □

The following sections introduce two subclasses of piecewise affine systems that are used in many practical applications, namely Hinging Hyperplane ARX systems, and Hammerstein/Wiener PWARX systems.

2.1.3 Hinging Hyperplane ARX systems

Piecewise affine functions defined by (2.9) may, in general, be discontinuous. A special class of continuous piecewise affine functions used for regression, classification, and function approximation, is represented by *Hinging Hyperplane* (HH) functions (Breiman, 1993). HH functions are defined as the sum of hinge functions:

$$f(\mathbf{x}) = \sum_{i=1}^{M} h_i(\mathbf{x}) \tag{2.11}$$

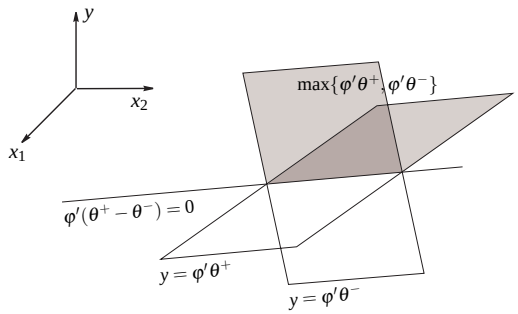


Figure 2.2 Two hinging hyperplanes $y = \varphi'\theta^-$ and $y = \varphi'\theta^+$, and the corresponding hinge function $y = \max\{\varphi'\theta^+, \varphi'\theta^-\}$, where $\varphi = [\, x_1\ x_2\ 1\,]'$

where each hinge function h_i, $i = 1,\dots,M$, geometrically consists of two half-hyperplanes joined continuously at the hinge (see Figure 2.2):

$$h_i(\mathbf{x}) = \pm\max\{\varphi'\theta_i^+, \varphi'\theta_i^-\}, \quad \varphi = [\,\mathbf{x}'\ 1\,]' \tag{2.12}$$

The $\pm$ sign here is needed to represent both convex and nonconvex functions. The class of HH functions is equivalent to the class of PWA functions that can be expressed in the *canonical representation* introduced by Kang and Chua (1978). It has been proved that a large class of (but not all) continuous piecewise affine functions possesses a canonical representation (Chua and Deng, 1988). Although it is not a universal representation of continuous PWA functions, the class of canonical PWA functions is however a universal approximant of all continuous functions on a compact subset of $\mathbb{R}^n$ (Lin and Unbehauen, 1992). This means that one can approximate any continuous function on a compact set arbitrarily well by using sufficiently many hinge functions (*i.e.*, by letting $M \to \infty$).

Using an alternative parameterization for the class of HH functions, *Hinging Hyperplane AutoRegressive eXogenous* (HHARX) systems are usually described in the following form:

$$y_k = \varphi_k'\theta_0 + \sum_{i=1}^{M^+} \max\{\varphi_k'\theta_i, 0\} - \sum_{i=M^++1}^{M} \max\{\varphi_k'\theta_i, 0\} \tag{2.13}$$

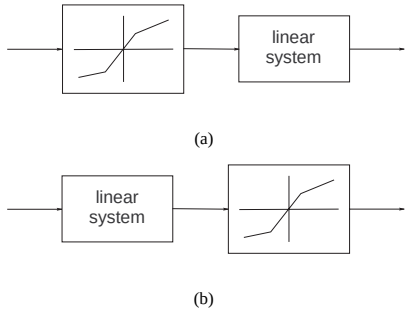


Figure 2.3 (a) Hammerstein and (b) Wiener systems with PWA static nonlinearity

where $y_k \in \mathbb{R}$ is the output of the system, and the extended regression vector φ_k is defined by (2.5). Since $\max\{z,0\} = z + \max\{-z,0\}$, $\forall z \in \mathbb{R}$, the parameters of (2.13) are not uniquely determined, *i.e.*, the same system can be described by several different sets of parameter values.

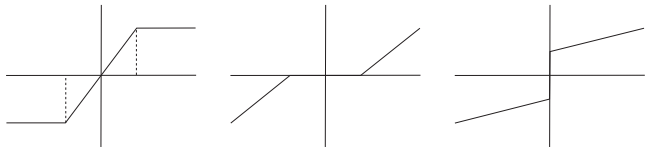
2.1.4 Hammerstein and Wiener PWARX systems

Hammerstein and Wiener systems form special classes of nonlinear systems with many practical applications. They consist of a linear dynamical system preceded (*Hammerstein* systems) or followed (*Wiener* systems) by a static nonlinearity (see Figure 2.3). When the static nonlinearity is piecewise affine, it is easy to verify that the overall system is also piecewise affine.

Hammerstein PWARX (H-PWARX) systems are given by the cascade connection of a piecewise affine static nonlinearity followed by an ARX system; see Figure 2.3(a). They are therefore described by the relations:

$$\begin{cases} x_k = g(u_k) \\ y_k = -\sum_{i=1}^{n_a} a_i y_{k-i} + \sum_{j=1}^{n_b} b_j x_{k-j} \end{cases} \tag{2.14}$$

where $y_k \in \mathbb{R}$ and $u_k \in \mathbb{R}$ are the output and the input of the system, respectively; $x_k \in \mathbb{R}$ is an internal variable that is not measurable; $a_i \in \mathbb{R}$, $i = 1,\dots,n_a$, and



(a) (b) (c)

Figure 2.4 Examples of hard nonlinearities: (a) saturation, (b) dead-zone, (c) preload

$b_j \in \mathbb{R}$, $j = 1,\dots,n_b$, are the parameters of the ARX system, and g is a piecewise affine function. The internal variable x_k is the input to the ARX system.

Conversely, *Wiener* PWARX (W-PWARX) systems are given by the cascade connection of an ARX system followed by a piecewise affine static nonlinearity; see Figure 2.3(b). They are therefore described by the relations:

$$\begin{cases} x_k = -\sum_{i=1}^{n_a} a_i x_{k-i} + \sum_{j=1}^{n_b} b_j u_{k-j} \\ y_k = g(x_k) \end{cases} \tag{2.15}$$

In this case, the internal variable x_k is the output of the ARX system.

PWA nonlinearities include saturations, dead-zones and preloads, that are common in engineering practice (see Figure 2.4).

2.2 Identification of piecewise affine models

PWA system identification concerns obtaining a piecewise affine model of a system from experimental data. PWA models represent an attractive model structure for identification purposes, since they are the "simplest" extension of linear models but can nevertheless describe nonlinear processes with arbitrary accuracy. PWA models are also capable of handling hybrid phenomena. Given the equivalence between PWA systems and several classes of hybrid systems (see Remark 2.1), PWA identification techniques can be used to obtain hybrid models.

PWARX models are suitable when dealing with input-output data, since they provide an input-output description of PWA systems. Assume that a collection $\mathscr{D}$ of N data points from the real system is available, namely:

$$\mathscr{D} = \{(y_k, \mathbf{x}_k),\ k = 1,\dots,N\} \tag{2.16}$$

where $y_k \in \mathbb{R}$ is the *measured* output of the system, and $\mathbf{x}_k \in \mathbb{R}^n$ is the regression vector (2.6) for fixed orders n_a and n_b. A PWARX model is defined as follows:

$$y_k = f(\mathbf{x}_k) + \varepsilon_k \tag{2.17}$$

where $\varepsilon_k \in \mathbb{R}$ is an error term (see Section 1.2.1), and f is the PWA map (2.9). The considered identification problem consists in finding the PWARX model that best matches the given data according to a specified criterion of fit. It involves the estimation of:

- The number of discrete modes s.
- The parameters θ_i, $i = 1,\dots,s$, of the affine submodels.
- The coefficients H_i, $i = 1,\dots,s$, of the hyperplanes defining the partition of the regressor set.

This issue also underlies a classification problem such that each data point is associated to one region, and to the corresponding submodel. The simultaneous optimal estimation of all the quantities mentioned above is a very hard, computationally intractable problem. To the knowledge of the author, no satisfactory formulation in the form of a single optimization problem has been even provided for it. One of the main difficulties is how to choose the number of discrete modes s. For instance, perfect fit is attained by $s = N$, *i.e.*, one submodel per data point, which is clearly an inadequate solution. Constraints on s must be hence introduced, so as to keep the number of submodels low, and to avoid overfit[4]. Heuristic and suboptimal approaches that are applicable, or at least related, to the identification of PWARX models, have been proposed in the literature. Most of these approaches either assume a fixed s, or adjust s iteratively (*e.g.*, by adding one submodel at a time) in

[4] The term *overfit* denotes the situation when a model adjusts itself to the particular noise realization, if given too many degrees of freedom.

order to improve the fit. The approach presented in this thesis, and described in Chapter 3, proposes a formulation of the identification problem which allows for the automatical estimation of a suitable s.

When the number of discrete modes s is fixed, the identification of a PWARX model amounts to a PWA regression problem, namely the problem of reconstructing the PWA map f from the finite data set $\mathscr{D}$. In this case, the identification process could be in principle carried out by minimizing with respect to θ_i and H_i, $i=1,\dots,s$, the following criterion function (see Section 1.2.2):

$$V_N(\theta_i,H_i)=\frac{1}{N}\sum_{k=1}^{N}\ell\big(y_k-f(\mathbf{x}_k)\big) \tag{2.18}$$

where ℓ is a given nonnegative function, *e.g.*, $\ell(\varepsilon)=\varepsilon^2$, or $\ell(\varepsilon)=|\varepsilon|$. The minimization of (2.18) for a fixed s is still a very hard, in general highly nonconvex problem with several local minima. The main difficulty is that the estimation of the regions $\mathscr{X}_i$, $i=1,\dots,s$, which determine the classification of the data points and are defined as follows:

$$\mathscr{X}_i=\left\{\mathbf{x}\in\mathbb{R}^n\,\middle|\,H_i\varphi\preceq\mathbf{0}\right\},\quad \varphi=[\,\mathbf{x}'\;1\,]' \tag{2.19}$$

cannot be decoupled from the identification of each submodel. Moreover, in order the PWA map f to be well defined, the collection $\{\mathscr{X}_i\}_{i=1}^{s}$ is implicitly constrained to form a complete partition of the regressor set $\mathscr{X}\subseteq\mathbb{R}^n$ where model (2.17) is valid. The problem becomes simple if the regions (2.19) are either known or fixed *a priori*. In this case, each regression vector $\mathbf{x}_k$ can be easily classified (*i.e.*, assigned to one region), and by introducing the quantities:

$$\chi_{ki}=\begin{cases}1 & \text{if } \mathbf{x}_k\in\mathscr{X}_i\\ 0 & \text{otherwise}\end{cases},\quad k=1,\dots,N,\quad i=1,\dots,s \tag{2.20}$$

the minimization of (2.18) can be expressed as follows:

$$\min_{\theta_i}\frac{1}{N}\sum_{k=1}^{N}\sum_{i=1}^{s}\ell\big(y_k-\varphi_k'\theta_i\big)\chi_{ki} \tag{2.21}$$

where $\varphi_k=[\,\mathbf{x}_k'\;1\,]'$. If $\ell(\varepsilon)=\varepsilon^2$, problem (2.21) is an ordinary least-squares problem in the unknowns θ_i. An overview of several approaches to PWA system identification will be given in the next section. Most approaches look for good suboptimal solutions of the identification problem, except the one by Roll *et al.* (2004), where the global optimum can be attained for two subclasses of PWA models by reformulations of the minimization of (2.18) into mixed integer linear or quadratic programs.

Remark 2.3 In the field of data analysis, a well-known problem is that of fitting the given data to s hyperplanes (or affine regressions). The number s can be either fixed or not; see, *e.g.*, (Bradley and Mangasarian, 2000; Amaldi and Mattavelli, 2002). Different from the identification of PWARX models, where region estimation must be also addressed, here the aim in only to classify the data points into clusters and to estimate an affine submodel for each cluster. Assuming that N data points $(y_k,\mathbf{x}_k)$ are given, with $y_k\in\mathbb{R}$ and $\mathbf{x}_k\in\mathbb{R}^n$, $k=1,\dots,N$, for a fixed s the considered problem can be formulated as follows:

$$\begin{cases}\min\limits_{\theta_i,\chi_{ki}} & \dfrac{1}{N}\displaystyle\sum_{k=1}^{N}\sum_{i=1}^{s}\ell\big(y_k-\varphi_k'\theta_i\big)\chi_{ki}\\ s.t. & \displaystyle\sum_{i=1}^{s}\chi_{ki}=1 \qquad k=1,\dots,N\\ & \chi_{ki}\in\{0,1\}\quad k=1,\dots,N,\;\; i=1,\dots,s\end{cases} \tag{2.22}$$

where $\varphi_k=[\,\mathbf{x}_k'\;1\,]'$, and ℓ is a given nonnegative function. Each binary variable χ_{ki} decides whether the data point $(y_k,\mathbf{x}_k)$ is assigned to the i-th submodel, under the constraint that each data point must be assigned to only one submodel. Problem (2.22) is a *mixed integer* program that is computationally intractable, except for small instances. Approaches to PWA system identification that first aim at classifying the data and estimating the affine submodels, and then at estimating the regions, can be in many cases easily adapted to deal with this problem. □

2.3 Approaches to PWA system identification

As discussed in the previous section, the identification of PWA models is a challenging problem. It involves the estimation of both the parameters of the affine

submodels, and the coefficients of the hyperplanes defining the partition of the regressor set. The problem is even more complicated when also the number of submodels must be estimated. Additional difficulties arise in the identification of state space models from input-output data. The matrices of system (2.1) are not unique, rather are defined up to a linear state transformation. This transformation must be the same for all submodels. Hence, once the matrices for each affine submodel have been estimated, all these must be suitably transferred to the same state basis.

In this section, an overview of different approaches to PWA system identification will be presented. The description is not intended to be exhaustive, and the interested reader is referred to (Roll, 2003) for more details. Work on regression with PWA maps can be found in many fields, *e.g.*, neural networks, electrical networks, time-series analysis, function approximation. Different categories of approaches can be distinguished depending on how the partitioning into regions is done. It follows from the discussion in Section 2.2 that there are mainly two alternative approaches: either the partition is defined *a priori*, or it is estimated along with the different submodels.

The first approach requires to define *a priori* the gridding of the regressor set (or the state+input set, for models in state space form). For instance, rectangular regions with sides parallel to the coordinate axes are used by Billings and Voon (1987), whereas simplices (*i.e.*, polytopes with $n+1$ corners, where n is the dimension of the space) are considered in (Julián *et al.*, 1999). This approach drastically simplifies the estimation of the linear/affine submodels, since standard linear identification techniques can be used to estimate the submodels, given enough data points in each region. On the other hand, it has the drawback that the number of regions, and hence the computational complexity and the need for experimental data, grow exponentially with n. This approach is therefore impracticable for higher-dimensional systems.

The second approach consists in estimating the submodels and the partition of the regressor set (or the state+input set) either simultaneously or iteratively. This should allow for the use of fewer regions, *i.e.*, for a reduced complexity of the

identified model, since the regions are shaped according to the data. Depending on how the partition is determined, Roll (2003) further distinguishes among four different categories of approaches.

The first category relies on the direct formulation of a suitable criterion function to be minimized, like (2.18). The parameters of the affine submodels and the coefficients of the hyperplanes defining the partition of the regressor set (or the state+input set) are hence estimated simultaneously by minimizing the criterion function through numerical methods (*e.g.*, Gauss-Newton search). The algorithms proposed in (Chan and Tong, 1986; Batruni, 1991; Julian *et al.*, 1998; Pucar and Sjöberg, 1998; Gad *et al.*, 2000) fall into this category. This way of tackling the identification problem is straightforward, but has the drawback that the optimization algorithm might be trapped in a local minimum. Techniques for reducing the risk of getting stuck in a local minimum can be used, at the cost of increased computational complexity.

The second category of approaches is an extension of the first one, allowing more flexibility with respect to the number of submodels. All parameters are identified simultaneously for a model with a very simple partition. If the resulting model is not satisfactory, new submodels/regions are added, in order to improve the value of a criterion function. In other words, instead to be solved at once, the overall identification problem is divided into several steps, each consisting in an easier problem to solve. The algorithms proposed in (Breiman, 1993; Heredia and Arce, 1996; Ernst, 1998; Hush and Horne, 1998) fall into this category. The first one has been further analyzed in (Pucar and Sjöberg, 1998). Julian *et al.* (1998) also describe an iterative method for introducing new partitions on the domain, when the error obtained is not satisfactory. As for the first category of approaches, there is still a risk to get stuck in a local minimum. When adding new submodels, one should also take into consideration the risk of *overfit*.

The third category contains a variety of approaches, sharing the characteristic that the parameters of the submodels and the partition of the regressor set (or the state+input set) are identified iteratively or in different steps, each step consider-

ing either the submodels or the regions. The algorithms proposed in (Bemporad *et al.*, 2003*a*; Ferrari-Trecate *et al.*, 2003; Vidal *et al.*, 2003*b*; Ragot *et al.*, 2003) start by classifying the data points and estimating the linear/affine submodels simultaneously. Then, region estimation is carried out by resorting to standard linear separation techniques. An online algorithm is proposed in (Skeppstedt *et al.*, 1992), where a multiple-model recursive parameter estimation algorithm is used to identify the current parameter values. In (Münz and Krebs, 2002), the position of rectangular regions is optimized one by one iteratively. Then, each rectangular region is divided into simplices, in which affine submodels are finally identified. In (Medeiros *et al.*, 2002), a greedy randomized adaptive search procedure is used to iteratively and heuristically find good partitions of the state space.

The last category of approaches estimates the partition using only information concerning the distribution of the regression vectors, and not the corresponding output values. The algorithms proposed in (Strömberg *et al.*, 1991; Choi and Choi, 1994) fall into this category. The major drawback of this category of approaches is that, without considering the output values, a set of data which really should belong to the same submodel might be split arbitrarily.

It should be noted that most of the aforementioned approaches, *e.g.*, (Batruni, 1991; Breiman, 1993; Choi and Choi, 1994; Heredia and Arce, 1996; Hush and Horne, 1998; Ernst, 1998; Julian *et al.*, 1998; Pucar and Sjöberg, 1998; Gad *et al.*, 2000) assume that the system dynamics is continuous, whereas, *e.g.*, (Bemporad *et al.*, 2003*a*; Ferrari-Trecate *et al.*, 2003; Vidal *et al.*, 2003*b*; Ragot *et al.*, 2003) allow for discontinuities.

Remark 2.4 (PWA identification via mixed-integer programming)
The minimization of the criterion function (2.18) is in general a highly nonconvex problem with several local minima, hence difficult to solve for the global optimum. In (Roll *et al.*, 2004) two subclasses of PWA models, namely HHARX models (2.13) and W-PWARX models (2.15), are considered. For these subclasses the global optimum can be attained by reformulations of the problem into mixed-integer linear or quadratic programs. The drawback of this approach is that the worst-case complexity is high, although for W-PWARX models it is also shown that the worst-case complexity is actually not exponential, rather polynomial. Nevertheless, this approach may be interesting in cases where relatively few data are available (*e.g.*, when it is very costly to obtain data), and where it is important to get a model which is as good as possible. □

Remark 2.5 (Identification of PWA Hammerstein and Wiener models)
Identification of Hammerstein and Wiener models with hard and discontinuous nonlinearities (*e.g.*, saturations, dead-zones, and preloads) is of great practical importance. Such nonlinearities are common in engineering problems, and can severely limit the performance of control systems. The knowledge of nonlinearity parameters may instead enable to cancel or reduce such adverse effects. Although identification of Hammerstein and Wiener models has been discussed quite extensively in the literature, only few approaches have been proposed for the identification of models with hard and discontinuous nonlinearities; see, *e.g.*, (Vörös, 1997; Bai, 2002) for Hammerstein models, and (Vörös, 2001; Pupeikis *et al.*, 2003; Pupeikis, 2003) for Wiener models. These approaches aim at identifying separately both the parameters of the linear system and the parameters of the nonlinearity. However, since Hammerstein and Wiener models with piecewise affine nonlinearities are themselves piecewise affine, another approach to the identification of such models might be to estimate the overall PWA model by using the techniques in this section, and then, if needed, to reconstruct the linear and the nonlinear parts; see, *e.g.*, (Roll *et al.*, 2004). □

The remaining part of this section gives a short description of the three identification procedures for PWARX models that have been recently proposed in (Ferrari-Trecate *et al.*, 2003), (Vidal *et al.*, 2003*b*), and (Ragot *et al.*, 2003). These procedures share with the one proposed in this thesis the idea to tackle the identification problem by first classifying the data and estimating the affine submodels, and then estimating the partition of the regressor set. The description will highlight the different ways in which data classification and parameter estimation are carried out

in each approach. Region estimation then corresponds to a problem of linear separation between clusters that can be addressed by resorting to standard techniques. Chapter 4 is dedicated to this topic. In the following, the notation introduced in Section 2.2 will be used.

The algorithm proposed in (Ferrari-Trecate *et al.*, 2003) exploits the combined use of clustering and linear identification techniques in order to classify the data and estimate the affine submodels of PWARX models. The number s of submodels is fixed a priori. For $k = 1,\dots,N$, a local data set $\mathscr{C}_k$ is formed by collecting $(y_k,\mathbf{x}_k)$ and the data points $(y_j,\mathbf{x}_j) \in \mathscr{D}$ with the $c-1$ nearest neighbors $\mathbf{x}_j$ to $\mathbf{x}_k$. The cardinality c of the local data sets is a parameter of the algorithm satisfying $c > n+1$. Local parameter vectors $\tilde{\theta}_k$ are obtained for each local data set $\mathscr{C}_k$ by using least squares. The centers:

$$m_k = \frac{1}{c} \sum_{(y_j,\mathbf{x}_j)\in\mathscr{C}_k} \mathbf{x}_j$$

of the local data sets are also computed, and the feature vectors $\xi_k = [\ \tilde{\theta}_k' \ m_k'\ ]'$ are formed. These are partitioned into s clusters by using a "K-means"-like algorithm which exploits suitably defined confidence measures for the feature vectors. It is indeed expected that most feature vectors form s dense clouds in the feature space. The resulting clusters $\mathscr{E}_i$ of feature vectors, $i = 1,\dots,s$, are used to form the clusters $\mathscr{D}_i$ of data points by assigning $(y_k,\mathbf{x}_k)$ to $\mathscr{D}_i$ if $\xi_k \in \mathscr{E}_i$. The data points in each cluster $\mathscr{D}_i$ are then used for estimating the parameter vectors θ_i of each submodel through weighted least squares. By performing the clustering in a suitably defined feature space, this algorithm is able to discriminate situations in which the same parameter vector is valid on different regions. A drawback is that the quality of the identified model, including the partition, could be spoiled by the presence of *mixed* local data sets, *i.e.*, local data sets collecting data points generated by different submodels. A modification of this algorithm has been presented in (Ferrari-Trecate and Muselli, 2003), where the use of single-linkage clustering is proposed for dealing with the estimation of the number of submodels.

The algorithm described by Vidal *et al.* (2003*b*) is a nice algebraic geometric solution for the identification of (noiseless) PWA systems. It establishes a connection between PWA system identification, polynomial factorization, and hyperplane clustering. If it is assumed that the data are generated by the noiseless PWARX system (2.4)-(2.5), then by introducing the following vectors:

$$\mathbf{b}_i = [\ \theta_i'\ 1\ ]' \in \mathbb{R}^K, \quad i = 1,\dots,s$$
$$\mathbf{z}_k = [\ \varphi_k'\ -y_k\ ]' \in \mathbb{R}^K$$

where s is the number of submodels and $K = n_a + n_b + 2$, it follows that at each time instant $k = 1,\dots,N$ there exists at least one $i \in \{1,\dots,s\}$ such that $\mathbf{b}_i'\mathbf{z}_k = 0$. Hence, the following constraint must be satisfied by the model parameters and the data:

$$\prod_{i=1}^{s} \mathbf{b}_i'\mathbf{z}_k = 0 \tag{2.23}$$

Equation (2.23) is called the *hybrid decoupling constraint* by the authors, since it allows to estimate the model parameters independently of the filtering of the discrete mode, *i.e.*, of the classification of the data points, and regardless of the mechanism generating the transitions. The polynomial $p_s(\mathbf{z}) = \prod_{i=1}^{s} \mathbf{b}_i'\mathbf{z}_k$, which is of degree s in K variables, can be rewritten as follows:

$$p_s(\mathbf{z}) = \sum h_{\alpha_1,\dots,\alpha_K} z_1^{\alpha_1}\dots z_K^{\alpha_K} \triangleq \mathbf{h}'\nu_s(\mathbf{z}) \tag{2.24}$$

where $h_{\alpha_1,\dots,\alpha_K} \in \mathbb{R}$ is the coefficient of the monomial $z_1^{\alpha_1}\dots z_K^{\alpha_K}$, with $0 \le \alpha_j \le s$, $j = 1,\dots,K$, and $\sum_{j=1}^{K} \alpha_j = s$. Vectors $\mathbf{h}$ and $\nu_s(\mathbf{z})$ have dimension:

$$M_s = \begin{pmatrix} s+K-1 \\ K-1 \end{pmatrix}$$

By considering (2.23) for all the available data, the following linear system of equalities is obtained:

$$L_s\mathbf{h} \triangleq \begin{bmatrix} \nu_s(\mathbf{z}_1)' \\ \vdots \\ \nu_s(\mathbf{z}_N)' \end{bmatrix} \mathbf{h} = 0 \tag{2.25}$$

Vidal *et al.* showed that, under mild assumptions, the number s of submodels can be determined as the minimum ℓ such that $\mathrm{rank}(L_\ell) = M_\ell - 1$. Then, the vector $\mathbf{h}$ can be obtained by solving (2.25), and the parameters θ_i can be retrieved from the

derivatives of (2.24). Lastly, data classification can be carried out by assigning each data point to the submodel i^* such that:

$$i^* = \arg \min_{i=1,\dots,s} (\mathbf{b}_i' \mathbf{z}_k)^2$$

The algorithm is designed for noiseless data, but the authors suggest how to adapt it in order to deal with noisy data. However, robustness of the algorithm with noisy data and outliers is an open issue.

The iterative algorithm proposed by Ragot *et al.* (2003) allows for data classification and sequential estimation of the parameters of a PWARX model through the use of adapted weights. The algorithm is not concerned with the estimation of the number of submodels and the regions, rather it attempts to solve the optimization problem (2.22) with $\ell(\varepsilon) = \varepsilon^2$ by relaxing the constraints on the weights χ_{ki}. An iterative procedure, alternating between parameter estimation given the weights, and weight update given the model parameters, is proposed. The aim is to adapt the weights in such a way that, for each $k = 1, \dots, N$, only one χ_{ki} approaches 1, whereas the other χ_{kj}, $j \neq i$, approach 0. These values are then converted into 0's and 1's, which provide the classification of the data points. Also for this method, performance in the case of noisy measurements needs further investigations.

3

PWA Identification using MIN PFS

In this chapter the main contribution of this thesis will be presented, consisting in a procedure for data classification and parameter estimation of PWARX models. Region estimation will be addressed in Chapter 4.

Inspired by ideas from set membership identification, the key approach here is to characterize the identified model by a bound δ on the fitting error. This allows to carry out data classification and parameter estimation, along with the estimation of the number of submodels (which is not fixed a priori), by partitioning a suitable set of linear inequalities derived from data into a minimum number of feasible subsystems (MIN PFS problem). A refinement procedure is also applied in order to reduce misclassifications, and to improve parameter estimates. It will be shown that the bound δ can be used as a tuning knob to trade off between quality of fit and model complexity. In addition, the identified model associates to each submodel a set of feasible parameters, thus allowing for evaluation of the related parametric uncertainty.

3.1 Problem Statement

In this section, the PWA system identification problem will be formulated by requiring the fitting error (*i.e.*, the difference between the system output and the predicted

output of the model) to be bounded by a given quantity δ. This idea arises from the unknown but bounded (UBB) error description used in set membership identification (see Section 1.3), where the uncertainty affecting the available data is described by means of additive noise, which is only known to have given bounds.

Assume that a collection of input-output samples $(\mathbf{u}_k, y_k)$, with $\mathbf{u}_k \in \mathbb{R}^p$ and $y_k \in \mathbb{R}$, generated by the discrete-time nonlinear dynamical system

$$y_k = F(\mathbf{u}^{k-1}, \mathbf{y}^{k-1}) + e_k \tag{3.1}$$

is given. F is a (possibly discontinuous) nonlinear function, $k \in \mathbb{Z}$ is time, $\mathbf{u}^{k-1}$ and $\mathbf{y}^{k-1}$ are, respectively, past system inputs and outputs up to time $k-1$:

$$\begin{aligned} \mathbf{u}^{k-1} &= [\, \mathbf{u}'_{k-1}\, \mathbf{u}'_{k-2}\, \dots\,]' \\ \mathbf{y}^{k-1} &= [\, y_{k-1}\, y_{k-2}\, \dots\,]' \end{aligned}$$

and e_k is additive noise. The construction of a model from data involves first of all the choice of a *model structure* within which a suitable model will be fitted. As described in Section 2.1.2, PWARX models are suitable when dealing with input-output data, since they provide an input-output description of PWA systems. It is repeated here for convenience that a PWARX model is defined as follows:

$$y_k = f(\mathbf{x}_k) + \varepsilon_k \tag{3.2}$$

where $\varepsilon_k \in \mathbb{R}$ is the error term, $\mathbf{x}_k \in \mathbb{R}^n$ is the regression vector with fixed structure depending on past n_a outputs and n_b inputs:

$$\mathbf{x}_k = [\, y_{k-1}\, \dots\, y_{k-n_a}\, \mathbf{u}'_{k-1}\, \dots\, \mathbf{u}'_{k-n_b}\,]' \tag{3.3}$$

(hence, $n = n_a + p \cdot n_b$), and $f : \mathscr{X} \to \mathbb{R}$ is the piecewise affine map:

$$f(\mathbf{x}) = \begin{cases} \varphi'\theta_1 & \text{if} \quad H_1\varphi \preceq \mathbf{0} \\ \quad\vdots & \qquad\vdots \\ \varphi'\theta_s & \text{if} \quad H_s\varphi \preceq \mathbf{0} \end{cases}, \quad \varphi = [\, \mathbf{x}'\ 1]' \tag{3.4}$$

In (3.4), s is the number of submodels (or *modes*), and $\theta_i \in \mathbb{R}^{n+1}$, $i = 1, \dots, s$, are the parameter vectors of each affine ARX submodel. The convex polyhedra

$$\mathscr{X}_i = \left\{ \mathbf{x} \in \mathbb{R}^n \,\middle|\, H_i\varphi \preceq \mathbf{0} \right\}, \quad \varphi = [\, \mathbf{x}'\ 1]' \tag{3.5}$$

with $H_i \in \mathbb{R}^{q_i \times (n+1)}$, $i = 1, \dots, s$, form a complete partition[1] of the regressor set $\mathscr{X} \subseteq \mathbb{R}^n$, *i.e.*, the region of validity of the PWARX model. Note that, once the orders n_a and n_b in (3.3) are chosen, it is often possible to describe $\mathscr{X}$ by considering the physical constraints on the inputs and the output of the system. In practice, these constraints are commonly specified in terms of box-bounds on each input (or output) sample, or on each input (or output) increment. For instance, typical constraints on the output are:

$$|y_k| \le y_{max} \quad \text{or} \quad |y_k - y_{k-1}| \le \Delta y_{max}$$

For a more compact notation, hereafter the extended regression vector $\varphi_k = [\, \mathbf{x}'_k\ 1\,]'$ will be considered, *i.e.*:

$$\varphi_k = [\, y_{k-1}\, \dots\, y_{k-n_a}\, \mathbf{u}'_{k-1}\, \dots\, \mathbf{u}'_{k-n_b}\, 1\,]'$$

It is also recalled that the PWA map (3.4) is not assumed to be continuous, and hence definition (3.4) is not well posed in general, since the PWA map could be multiply defined over common boundaries of the regions $\mathscr{X}_i$. See Remark 2.2 for how to avoid this problem.

The objective of the considered identification problem is to find a PWARX model (3.2)-(3.4) of system (3.1) that matches as good as possible the given data points $(y_k, \mathbf{x}_k)$, $k = 1, \dots, N$, according to a suitably specified criterion of fit. The approach proposed in this thesis characterizes the identified model by its maximum fitting error, *i.e.*, it is required that:

$$|y_k - f(\mathbf{x}_k)| \le \delta\,, \quad k = 1, \dots, N \tag{3.6}$$

for a fixed $\delta > 0$. The number of submodels s is not assumed to be known, or a priori fixed, rather it is estimated along with the parameters of the model so as to satisfy condition (3.6). In order to obtain a model which is as simple as possible (where "simplicity" is intended in terms of the number of submodels), the minimum s allowing to satisfy (3.6) is sought. Under condition (3.6), the considered identification problem can be stated as follows.

[1] $\bigcup_{i=1}^{s} \mathscr{X}_i = \mathscr{X}$ and $\mathscr{X}_i \bigcap \mathscr{X}_j = \emptyset$, $\forall\, i \neq j$

Problem 3.1 *Given the N data points $(y_k, \mathbf{x}_k)$, $k = 1, \ldots, N$, and $\delta > 0$, estimate a minimum positive integer s, parameter vectors $\{\theta_i\}_{i=1}^{s}$, and a polyhedral partition $\{\mathscr{X}_i\}_{i=1}^{s}$ of the regressor set $\mathscr{X}$, such that the corresponding PWARX model (3.2)-(3.4) satisfies condition (3.6).*

A solution of Problem 3.1 provides a suitable number of submodels s and, for this s, a suboptimal solution for the minimization of the criterion function (2.18). It is stressed that the bound δ is not necessarily given a priori, it is rather a tuning knob of the identification procedure. In Section 3.2.4 it will be shown that δ can be used in order to find the desired trade off between the complexity of the model in terms of the number of submodels, and the quality of fit. Indeed, the smaller δ, the larger the number of submodels needed to fit the data points to a PWA map (3.4). On the other hand, the larger δ, the worse the fit, since large errors are allowed.

Remark 3.1 The problem of finding a PWA approximation of a given nonlinear function can be easily cast into Problem 3.1. In this case, one has a nonlinear function $F : \mathscr{X} \mapsto \mathbb{R}$, and wants to find a PWA function (3.4) that approximates N given samples $(y_k, \mathbf{x}_k)$, $k = 1, \ldots, N$, with desired accuracy, where $y_k \in \mathbb{R}$ represent values of F obtained at certain points $\mathbf{x}_k \in \mathscr{X} \subseteq \mathbb{R}^n$:

$$y_k = F(\mathbf{x}_k) + e_k \tag{3.7}$$

Here $e_k \in \mathbb{R}$ is an error term that can be either zero (*e.g.*, when F can be computed analytically), or nonzero (*e.g.*, when F is evaluated numerically by iterative procedures, as in the case of implicit functions or optimal value functions). By requiring condition (3.6), the approximation problem consists in finding a minimum s and a PWA function (3.4) such that the maximum approximation error is bounded by δ. The bound δ hence represents the desired precision in approximating F. □

The procedure for solving Problem 3.1 proposed in this thesis, consists of the following three steps:

- *Initialization*, in which data classification and parameter estimation are carried out, along with the estimation of the number of submodels, by partitioning a suitable set of linear inequalities derived from data into a minimum number of feasible subsystems (MIN PFS problem).
- *Refinement*, whose aim is to reduce misclassifications and to improve parameter estimates.
- *Region estimation*, performed by separating the clusters of regression vectors via two-class or multi-class linear separation techniques.

The first two steps will be described in Sections 3.2 and 3.3. New ideas for efficiently addressing the MIN PFS problem will be also discussed. Region estimation will be considered in Chapter 4. Note that condition (3.6) naturally leads to a set membership or bounded error approach to the identification problem. In the following, pointwise parameter estimates will be computed by using the ℓ_∞ projection estimator, already introduced in Section 1.3. Given a set $\mathscr{D}$ of data points $(y_k, \mathbf{x}_k)$, the projection estimate is computed as follows:

$$\Phi_p(\mathscr{D}) = \arg\min_{\theta} \max_{(y_k, \mathbf{x}_k) \in \mathscr{D}} \left| y_k - \varphi_k' \theta \right| \tag{3.8}$$

where $\varphi_k = [\, \mathbf{x}_k' \; 1 \,]'$. Computation of (3.8) can be carried out by solving a suitable linear program like (1.26). The projection estimate is preferred because it has favorable properties for the refinement procedure. It could be however replaced by any other pointwise estimate, *e.g.*, the least squares estimate.

The following example will be used throughout this chapter for illustrating the mechanism of the proposed identification procedure.

Example 3.1 Let the data $(y_k, \mathbf{x}_k)$ be generated by the PWARX system:

$$y_k = \begin{cases} -0.4 y_{k-1} + u_{k-1} + 1.5 + e_k & \text{if} \quad 4y_{k-1} - u_{k-1} + 10 < 0 \\ 0.5 y_{k-1} - u_{k-1} - 0.5 + e_k & \text{if} \quad 4y_{k-1} - u_{k-1} + 10 \geq 0 \;\text{ and} \\ & \qquad\; 5y_{k-1} + u_{k-1} - 6 \leq 0 \\ -0.3 y_{k-1} + 0.5 u_{k-1} - 1.7 + e_k & \text{if} \quad 5y_{k-1} + u_{k-1} - 6 > 0 \end{cases}$$

The number of submodels is $\bar{s} = 3$. The input signal u_k and the noise signal e_k are uniformly distributed on $[-4, 4]$ and $[-0.2, 0.2]$, respectively. $N = 200$ estimation data points are used. The (unknown) partition of the regressor set, and the set

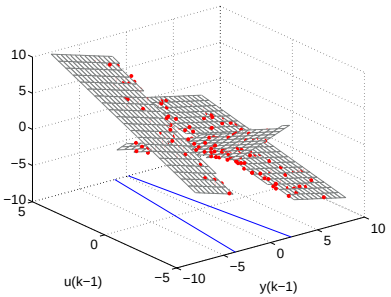


Figure 3.1 The available regression vectors and the partition of the regressor set (which is not assumed to be known during the identification process) for Example 3.1

of regression vectors $\mathbf{x}_k = [\, y_{k-1}\ u_{k-1}\,]'$ available for estimation are shown in Figure 3.1. From left to right, the three regions contain 54, 83 and 63 points.

3.2 Initialization using MIN PFS

In the first part of the proposed procedure for solving Problem 3.1, the estimation of the hyperplanes defining the polyhedral partition of the regressor set is temporarily not addressed. The focus is rather on determining a suitable number of submodels, classifying the data points and estimating the affine submodels. By relaxing the estimation of the regions, Problem 3.1 reduces to the following:

Problem 3.2 *Given $\delta > 0$ and the system of N linear complementary inequalities:*

$$\left| y_k - \varphi_k' \theta \right| \leq \delta\,, \quad k = 1, \ldots, N \tag{3.9}$$

find a partition of (3.9) into a minimum number of feasible subsystems.

The above formulation enables to address simultaneously the fundamental issues of data classification and parameter estimation, along with the estimation of the number of submodels. Given any solution of Problem 3.2, the partition of the linear complementary inequalities (3.9) provides the classification of the data points,

whereas according to the bounded error condition each feasible subsystem defines the set of feasible parameter vectors for the corresponding affine submodel. Note that each inequality (3.9) is termed a *linear complementary* inequality because it corresponds to the pair of linear inequalities:

$$\begin{cases} \varphi_k' \theta \leq y_k + \delta \\ \varphi_k' \theta \geq y_k - \delta \end{cases} \tag{3.10}$$

Problem 3.2 consists in finding a *Partition* of a system of linear inequalities into a *Minimum* number of *Feasible Subsystems* (MIN PFS problem), with the additional constraint that two paired linear inequalities (3.10) must be included in the same subsystem (*i.e.*, they must be simultaneously satisfied by the same parameter vector θ). The MIN PFS problem is NP-hard. Hence, in (Bemporad *et al.*, 2003*a*) it was suggested to tackle Problem 3.2 by resorting to the greedy randomized algorithm proposed by Amaldi and Mattavelli (2002). In the next sections, some modifications to the original algorithm are proposed in order to obtain a number s of subsystems which is closer to be minimal. The algorithm also provides s disjoint sets of indices $\mathscr{I}_i$, $i = 1, \ldots, s$, characterizing the s subsystems extracted from (3.9). These induce the initial classification of the data points $(y_k, \mathbf{x}_k)$, $k = 1, \ldots, N$, into the s clusters $\mathscr{D}_i^{(0)} = \left\{ (y_k, \mathbf{x}_k) \,\middle|\, k \in \mathscr{I}_i \right\}$, $i = 1, \ldots, s$.

3.2.1 A greedy approach to MIN PFS

The greedy approach to the MIN PFS problem with complementary inequalities proposed by Amaldi and Mattavelli (2002) divides the overall partition problem into a sequence of subproblems, where each subproblem consists in finding a parameter vector θ that satisfies the maximum number of linear complementary inequalities. Starting from (3.9), feasible subsystems with maximum cardinality are iteratively extracted (and the corresponding inequalities removed), until the remaining subsystem is feasible. This strategy clearly yields a partition into feasible subsystems. Since finding a *Feasible Subsystem* with *MAXimum* cardinality of a system of linear inequalities (MAX FS problem) is also NP-hard, a randomized and thermal relax-

Set $\bar{\mathscr{I}}_1 = \{1,\dots,N\}$ and $s = 0$

REPEAT

Set $s = s+1$ and $\Sigma_s = \{|y_k - \varphi_k'\theta| \le \delta \,|\, k \in \bar{\mathscr{I}}_s\}$

Find a solution θ_s of the MAX FS problem for system Σ_s (see **Table 3.2**)

Set $i = 1$

WHILE $i < s$

Set $\mathscr{I}_{is} = \{k \in \bar{\mathscr{I}}_i \,|\, |y_k - \varphi_k'\theta_s| \le \delta\}$

IF $\#\mathscr{I}_{is} > \#\mathscr{I}_i$ THEN set $\theta_i = \theta_s$ and $s = i$, BREAK

Set $i = i+1$

END WHILE

Set $\mathscr{I}_s = \{k \in \bar{\mathscr{I}}_s \,|\, |y_k - \varphi_k'\theta_s| \le \delta\}$ and $\bar{\mathscr{I}}_{s+1} = \bar{\mathscr{I}}_s \setminus \mathscr{I}_s$

UNTIL $\bar{\mathscr{I}}_{s+1} = \emptyset$

RETURN s and $\mathscr{I}_i$, $i = 1,\dots,s$

Table 3.1 Modified greedy algorithm for the MIN PFS problem with complementary inequalities. The BREAK command is used to terminate the WHILE loop

ation method, which provides (suboptimal) solutions with a low computational burden, is also proposed in (Amaldi and Mattavelli, 2002). As it will be discussed in Section 3.2.3, due to both the suboptimality of the greedy approach to the MIN PFS problem, and the randomness of the method used to tackle each single MAX FS problem, the resulting greedy randomized algorithm for Problem 3.2 is not guaranteed to yield minimum partitions, *i.e.*, the number of extracted subsystems might be not minimal. In particular, it was observed in extensive trials that both the variance of the results can be quite large (*i.e.*, the number of extracted subsystems may differ considerably from trial to trial), and the average number of extracted subsystems can be quite far from the minimum. Some modifications to the original algorithm by Amaldi and Mattavelli are hence here proposed in order to obtain a number s of subsystems which is closer to be minimal.

The modified greedy algorithm for the MIN PFS problem with complementary inequalities is shown in Table 3.1. It differs from the original version for the addition of the WHILE loop. Let Σ_s be the system consisting of the remaining inequalities after having extracted $s-1$ feasible subsystems from (3.9), and let θ_s be a (suboptimal) solution of the MAX FS problem for system Σ_s provided by the algorithm shown in Table 3.2. The solution θ_s is applied to the systems Σ_i with $i < s$ (WHILE loop). Note that Σ_s is a subsystem of Σ_i for all $i < s$, so that θ_s satisfies at least as many complementary inequalities in Σ_i as in Σ_s. Let i^* be the first index i, if any, such that θ_s satisfies a larger number of complementary inequalities in Σ_i than those satisfied by θ_i. Then, the best solution θ_{i^*} found for system Σ_{i^*} is set equal to θ_s, and s is reset to i^*. Since the number of data points is finite, this algorithm always terminates. Improvements obtained by the proposed modification to the original algorithm are twofold. First, the cardinalities of successively extracted subsystems form a decreasing sequence, as it would be expected if one could solve each MAX FS problem exactly. Second, it allows to form subsystems with larger cardinality, *e.g.*, when two subsystems of complementary inequalities that could be satisfied by one and the same parameter vector, are extracted at two different steps (see Section 3.2.3). The second improvement is obtained also in combination with modifications to the randomized and thermal relaxation algorithm used to tackle each MAX FS problem, that will be described in the next section.

3.2.2 A relaxation method for MAX FS

Given a system of complementary inequalities like (3.9), the problem of determining a parameter vector θ that satisfies as many pairs of complementary inequalities as possible, extends the combinatorial problem of finding a feasible subsystem with maximum cardinality of an infeasible system of linear inequalities, which is known as MAX FS problem. Based on the consideration that the MAX FS problem is NP-hard, Amaldi and Mattavelli (2002) tackle its extension with complementary inequalities by resorting to a randomized and thermal variant of the classical Agmon-Motzkin-Schoenberg relaxation method for solving systems of linear inequalities (Agmon, 1954; Motzkin and Schoenberg, 1954), which provides (suboptimal) so-

lutions with a low computational burden. In this section, some modifications to the original algorithm by Amaldi and Mattavelli are proposed in order to obtain a feasible subsystem with cardinality closer to be maximal.

The modified randomized and thermal relaxation algorithm for the MAX FS problem with complementary inequalities is shown in Table 3.2. It differs from the original version for the addition of the last IF statement. The algorithm requires to define a maximum number of cycles $C > 0$, an initial temperature parameter $T_0 > 0$, an initial estimate $\theta^{(0)} \in \mathbb{R}^{n+1}$ (*e.g.*, randomly generated, or computed by least squares), and a value $\rho \in (0,1)$. It consists in a simple iterative procedure generating a sequence $\theta^{(j)}$ of estimates, where $j = 1,\ldots,CN_s$ is the iteration counter, and N_s is the number of complementary inequalities in the subsystem Σ_s of (3.9) at hand (see Table 3.1). During each of the C outer cycles, all the N_s complementary inequalities in Σ_s are selected in the order defined by a prescribed rule (*e.g.*, cyclicly, or uniformly at random without replacement). Assume that the complementary inequality $\left|y_k - \varphi_k'\theta\right| \leq \delta$ is considered at the j-th iteration, while all the others are relaxed. Then the current estimate is updated as follows:

$$\theta^{(j)} = \theta^{(j-1)} - \text{sign}(v_j^k)\,\lambda_j\,\varphi_k \tag{3.11}$$

where the violation v_j^k of the k-th complementary inequality is computed as:

$$v_j^k = \begin{cases} \varphi_k'\theta^{(j-1)} - y_k - \delta & \text{if } \varphi_k'\theta^{(j-1)} > y_k + \delta \\ \varphi_k'\theta^{(j-1)} - y_k + \delta & \text{if } \varphi_k'\theta^{(j-1)} < y_k - \delta \\ 0 & \text{otherwise} \end{cases} \tag{3.12}$$

and the size of the step λ_j decreases exponentially with $\left|v_j^k\right|$:

$$\lambda_j = \frac{T}{T_0}\exp^{-\frac{\left|v_j^k\right|}{T}} \tag{3.13}$$

Geometrically, the complementary inequality $\left|y_k - \varphi_k'\theta\right| \leq \delta$ defines a hyperstrip in the parameter space (see Figure 3.2). If the current estimate $\theta^{(j-1)}$ belongs to the hyperstrip (*i.e.*, $\theta^{(j-1)}$ satisfies the k-th complementary inequality), then $\theta^{(j)}$ is set equal to $\theta^{(j-1)}$. Otherwise, $\theta^{(j)}$ is obtained by making a step toward the hyperstrip

GIVEN: C, T_0, $\theta^{(0)}$, ρ
Set $j = 0$, $\bar{\theta} = \theta^{(0)}$ and $\bar{\mathscr{I}} = \left\{k \in \mathscr{I}_s \,\middle|\, |y_k - \varphi_k'\bar{\theta}| \leq \delta\right\}$
FOR $c = 0$ TO $C-1$ DO
  Compute (3.14) and set $\mathscr{I} = \bar{\mathscr{I}}_s$
  REPEAT
    Set $j = j+1$
    Pick an index k from $\mathscr{I}$ according to the prescribed rule
    Compute (3.12), (3.13) and (3.11)
    Set $\mathscr{I}^{(j)} = \left\{k \in \bar{\mathscr{I}}_s \,\middle|\, |y_k - \varphi_k'\theta^{(j)}| \leq \delta\right\}$
    IF $\#\mathscr{I}^{(j)} > \#\bar{\mathscr{I}}$ THEN set $\bar{\theta} = \theta^{(j)}$ and $\bar{\mathscr{I}} = \mathscr{I}^{(j)}$
    Set $\mathscr{I} = \mathscr{I} \setminus \{k\}$
  UNTIL $\mathscr{I} = \emptyset$
  IF $c > \rho C$ THEN
    Set $\bar{\mathscr{D}} = \left\{(y_k, \mathbf{x}_k) \,\middle|\, k \in \bar{\mathscr{I}}\right\}$
    Compute $\bar{\theta} = \Phi_p(\bar{\mathscr{D}})$ and set $\theta^{(j)} = \bar{\theta}$
    Set $\bar{\mathscr{I}} = \left\{k \in \bar{\mathscr{I}}_s \,\middle|\, |y_k - \varphi_k'\bar{\theta}| \leq \delta\right\}$
  END IF
END FOR
RETURN $\bar{\theta}$

Table 3.2 Modified randomized and thermal relaxation algorithm for the MAX FS problem with complementary inequalities

along the line orthogonal to the hyperstrip and passing through $\theta^{(j-1)}$. The basic idea of the algorithm is to favor updates of the current estimate which aim at correcting unsatisfied inequalities with a relatively small violation. Decreasing attention to unsatisfied inequalities with large violations (whose correction is likely to corrupt other inequalities that the current estimate satisfies) is obtained by introducing a decreasing temperature parameter T, to which the violations are compared, *e.g.*:

$$T = \left(1 - \frac{c}{C}\right) T_0 \tag{3.14}$$

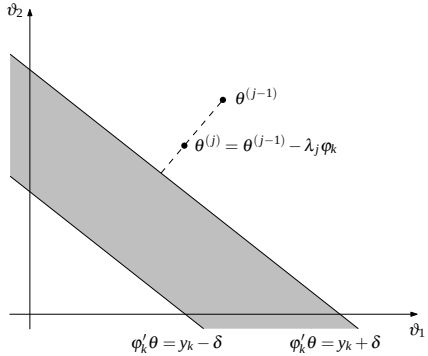


Figure 3.2 Geometric interpretation in the parameter space of a single iteration of the relaxation method for the MAX FS problem with complementary inequalities (case $\theta \in \mathbb{R}^2$)

where $c = 0, 1, \ldots, C-1$ is the outer cycle counter. If c is greater than ρC (last IF statement), the current best solution $\bar{\theta}$ (*i.e.*, the one that has satisfied so far the larger number of complementary inequalities), as well as the current estimate $\theta^{(j)}$, are replaced by the projection estimate (3.8). Namely, by denoting as $\bar{\mathscr{D}}$ the set of data points $(y_k, \mathbf{x}_k)$ such that the corresponding inequalities $|y_k - \varphi_k'\theta| \leq \delta$ are in Σ_s and are satisfied by the current $\bar{\theta}$ ($\bar{\theta}_{\text{old}}$ in the following), $\bar{\theta}$ is updated as follows:

$$\bar{\theta} = \arg\min_{\theta} \max_{(y_k, \mathbf{x}_k) \in \bar{\mathscr{D}}} \left| y_k - \varphi_k'\theta \right| \tag{3.15}$$

The new $\bar{\theta}$ satisfies at least as many complementary inequalities in Σ_s as $\bar{\theta}_{\text{old}}$, since

$$\max_{(y_k, \mathbf{x}_k) \in \bar{\mathscr{D}}} \left| y_k - \varphi_k'\bar{\theta} \right| \leq \max_{(y_k, \mathbf{x}_k) \in \bar{\mathscr{D}}} \left| y_k - \varphi_k'\bar{\theta}_{\text{old}} \right| \leq \delta \,,$$

and possibly might satisfy more complementary inequalities than $\bar{\theta}_{\text{old}}$, thus providing a better solution of the MAX FS problem for system Σ_s. It was found experimentally that suitable values for ρ lie between 0.7 and 0.8. Indeed, in the original version of the algorithm, the current estimate $\theta^{(j)}$ (and hence the number of satisfied complementary inequalities) does not change significantly anymore as c approaches C, because the temperature parameter (3.14) to which the violations are compared, becomes smaller and smaller. By resetting $\theta^{(j)}$ to the current best estimate (3.15) at

the exit of a cycle when c approaches C, focuses the future search in a neighborhood of $\bar{\theta}$, where it is more likely to satisfy an even larger number of complementary inequalities. The solution $\bar{\theta}$ returned by the algorithm is the one that, during the overall process, has satisfied the largest number of complementary inequalities. It is however not guaranteed to be optimal, due to the randomness of the search.

For the choice of T_0, as well as for practical questions concerning the implementation of the algorithm, the reader is referred to (Amaldi and Mattavelli, 2002). In general, the larger the value of C, the better the solution, at the price of a longer computation time. The proposed modifications allow however to obtain better solutions than the original algorithm also for considerably smaller values of C, as shown in Example 3.1 at page 60.

3.2.3 Comments on the initialization

When the greedy randomized algorithm described in Sections 3.2.1 and 3.2.2 is applied for initializing the identification procedure, the estimate of the number of affine submodels and the classification of the data points thus obtained, may suffer two drawbacks. The major drawback is that the algorithm is not guaranteed to yield minimum partitions. Due to the suboptimality of the greedy approach, the number s of feasible subsystems extracted from (3.9) might be not minimal, even if feasible subsystems with maximum cardinality were available at each step, as shown by the following simple example.

Example 3.2 Consider the infeasible system of equalities:

$$\vartheta_1 - \vartheta_2 = 0 \tag{3.16}$$

$$\vartheta_1 + \vartheta_2 = 0 \tag{3.17}$$

$$\vartheta_1 + 2\vartheta_2 = 3 \tag{3.18}$$

$$0.5\vartheta_1 + \vartheta_2 = 0.5 \tag{3.19}$$

which can be seen as a particular case of (3.9) for $\delta = 0$. A partition of this system into a minimum number of feasible subsystems consists of only two subsystems,

e.g., the one composed by (3.16) and (3.18), and the one composed by (3.17) and (3.19). However note that, if the greedy algorithm starts by extracting, among the feasible subsystems with maximum cardinality, the one composed by (3.16) and (3.17), then the two remaining equalities are infeasible, and the resulting partition consists of three subsystems. Note also that the optimal solution of the MIN PFS problem for the system of equalities (3.16)-(3.19) is not unique. Another minimum partition consists of the two subsystems composed by (3.16) and (3.19), and by (3.17) and (3.18), respectively.

In fact, feasible subsystems with maximum cardinality might be not even available at each step, due to the randomness of the algorithm used to tackle each single MAX FS problem. For instance, two subsystems of complementary inequalities that could be satisfied by one and the same parameter vector, might be extracted at different steps, because the search process was not able to find at once a parameter vector θ satisfying all the inequalities in both subsystems. Moreover, after extracting a certain number of subsystems with larger cardinalities, the algorithm typically starts to extract relatively small "mixed" subsystems. These contain leftover complementary inequalities corresponding to outliers, or that should have been assigned to previously extracted subsystems, if the search process was able to find suitable parameter vectors. Both circumstances determine an overestimation of the minimum number of feasible subsystems of (3.9). Based on these considerations, some modifications to the original greedy randomized algorithm by Amaldi and Mattavelli (2002) for Problem 3.2 were proposed in Sections 3.2.1 and 3.2.2 in order to obtain a number of feasible subsystems which is closer to be minimal. Note also that one could decide to stop the algorithm when the cardinalities of the extracted feasible subsystems become too small. This might be useful in order to penalize (most likely, "mixed") subsystems that account for just a few complementary inequalities.

The second drawback is related to a kind of ambiguity that is inherent to the data. Some data points may be consistent with more than one affine submodel, *i.e.*, they may satisfy $|y_k - \varphi_k'\theta_i| \leq \delta$ for more than one $i = 1,\ldots,s$. These data points will be termed *undecidable* in Section 3.3. Due to the undecidable data points, the

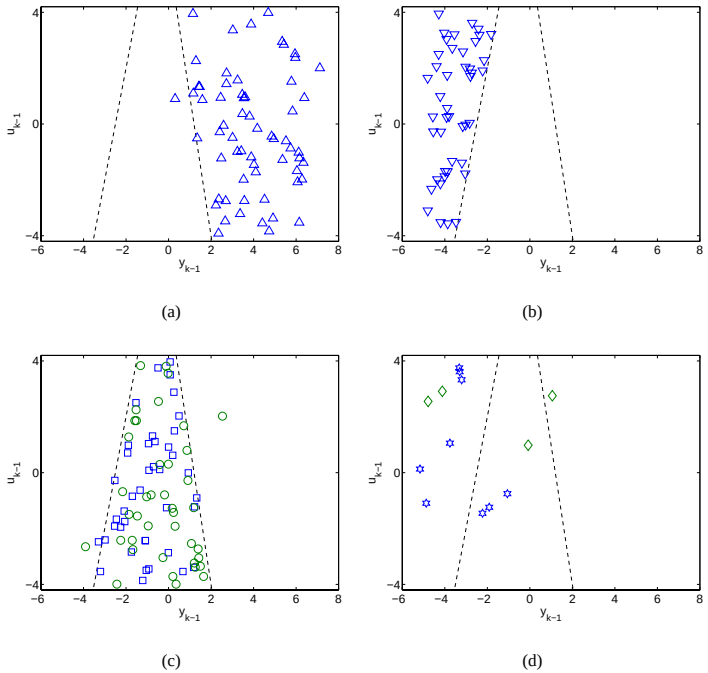


Figure 3.3 Initial classification of the regression vectors for Example 3.1 by using the original greedy randomized algorithm for Problem 3.2. Each mark corresponds to a different cluster, for a total of six clusters. The clusters in Figures 3.3(a) and 3.3(b) consist of 62 and 44 points, respectively. The two clusters in Figure 3.3(c) consist of 41 and 40 points. Last, the two clusters in Figure 3.3(d) consist of 9 and 4 points. The dashed lines represent the true partition of the regressor set, which is assumed to be unknown

cardinality and the composition of the feasible subsystems of (3.9) could depend on the order in which the feasible subsystems are extracted by the greedy algorithm.

In order to cope with these drawbacks, a refinement procedure will be described in Section 3.3. Its aim is to iteratively improve both data classification and quality of fit by properly reassigning the data points, and updating the parameter estimates. It also allows to reduce the number of submodels by exploiting parameter similarities and cluster cardinalities.

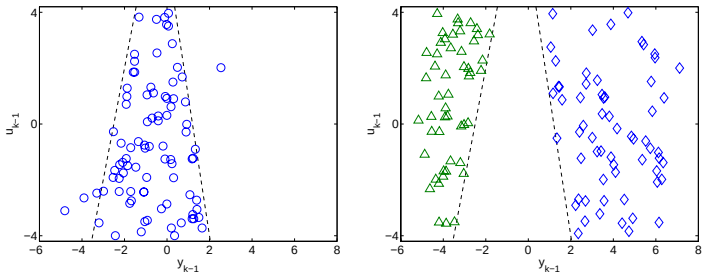


Figure 3.4 Initial classification of the regression vectors for Example 3.1 by using the modified greedy randomized algorithm for Problem 3.2. Each mark corresponds to a different cluster, for a total of three clusters. The cluster in Figure 3.4(a) consists of 87 points. The left and right clusters in Figure 3.4(b) consist of 51 and 62 points, respectively. The dashed lines represent the true partition of the regressor set, which is assumed to be unknown

Example 3.1 (cont'd**)** The original and the modified version of the greedy randomized algorithm for Problem 3.2 described in Sections 3.2.1 and 3.2.2 were applied to the data set of Example 3.1. Since the noise was uniformly distributed on $[-0.2, 0.2]$, the bound δ was chosen equal to 0.2 accordingly.

By running the original algorithm by Amaldi and Mattavelli (2002) with parameters $C = 200$ and $T_0 = 100$, and cyclic selection of the complementary inequalities, a partition of (3.9) into $s = 6$ feasible subsystems was found, consisting of 62, 44, 41, 40, 9 and 4 complementary inequalities, respectively. It is stressed that this was the best solution obtained after several trials (*i.e.*, all the other trials provided a larger number of subsystems). The true number of submodels is overestimated. The corresponding six clusters of regression vectors $\mathbf{x}_k$ are shown in Figure 3.3. Figure 3.3(c) illustrates a situation in which two subsystems of complementary inequalities corresponding to data points generated by the same submodel were extracted at different steps, although a single parameter vector could satisfy both subsystems. Figure 3.3(d) shows two "mixed" subsystems with small cardinality that were last extracted when few complementary inequalities formed the remaining system after having removed from (3.9) four subsystems with larger cardinalities.

By running the modified algorithm proposed in Sections 3.2.1 and 3.2.2 with parameters $C = 10$ and $T_0 = 100$, and cyclic selection of the complementary inequalities, a partition of (3.9) into $s = 3$ feasible subsystems was found, consisting of 87, 62 and 51 complementary inequalities, respectively. The estimated number of submodels equals the true one. This correct result was obtained with just one trial and a smaller number of cycles ($C = 10$) than that ($C = 200$) used in the runs of the original algorithm. The corresponding three clusters of regression vectors are shown in Figure 3.4. In Figure 3.4(a) some data points clearly look as misclassified. They are *undecidable* data points (*i.e.*, consistent with more than one submodel) that were associated by the greedy strategy to the compatible submodel corresponding to the largest feasible subsystem extracted from (3.9).

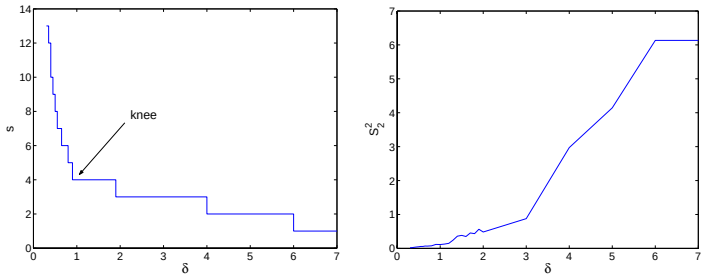
3.2.4 On the choice of δ

The bound δ is a tuning knob of the algorithm allowing to trade off between model complexity and quality of fit. For too large values of δ, very large subsystems of (3.9) are feasible, and beyond a certain value of δ the whole system (3.9) becomes feasible. The identified PWARX model is simple because it contains very few affine submodels, but the submodels do not fit well the corresponding data points, as large errors are allowed. Conversely, too small values of δ lead to a very large number of subsystems. There is a risk of overfit, *i.e.*, the model starts to adjust to the particular noise realization.

When *a priori* information on the system structure and the noise characteristics is not available, an appropriate value of δ can be selected solving Problem 3.2 for a range of values of δ. Given the low computational burden of the greedy randomized algorithm for Problem 3.2, the curves expressing the number of feasible subsystems of (3.9) and the average quadratic error

$$S_2^2 = \frac{1}{N} \sum_{i=1}^{s} \sum_{k \in \mathscr{S}_i} \left| y_k - \varphi_k' \theta_i \right|^2 \tag{3.20}$$

as a function of δ, can be easily plotted. Typically, when δ increases starting from a very small value, the number of feasible subsystems first sharply decreases and then



(a) (b)

Figure 3.5 (a) Number of submodels and (b) average quadratic error versus the maximum fitting error δ for Example 3.3

remains almost constant on a range of values. Conversely, the average quadratic error increases with δ. An appropriate value of δ should be chosen close to the knee of the first curve, trying to keep the average quadratic error small.

Example 3.3 $N = 500$ data points generated by a PWARX system composed by $\overline{s} = 4$ subsystems with $n_a = 1$ and $n_b = 1$, were considered. The additive noise was normally distributed with zero mean and variance $\sigma^2 = 0.1$, and the signal-to-noise ratio was about 10. The number of feasible subsystems of (3.9), and the corresponding average quadratic error are plotted as a function of δ in Figure 3.5. For values of δ below 0.9 ($\simeq 2.8\sigma$), the average quadratic error is small, but the large number of submodels clearly indicates overfit of the data. For values of δ between 0.9 and 1.9 ($\simeq 6\sigma$), the number of submodels remains constant and equal to the true number $\overline{s}$, whereas the average quadratic error grows moderately with δ. For values beyond $\delta = 6$, system (3.9) becomes feasible, and only one submodel is sufficient. It is evident in Figure 3.5 that the best trade-off between model accuracy and model complexity is achieved in this example for δ ranging from 0.9 to 1.

3.3 A Refinement Procedure

The initialization procedure described in Section 3.2 provides a number s of submodels, and s clusters $\mathscr{D}_i^{(0)}$ formed by the data points $(y_k, \mathbf{x}_k)$ corresponding to the i-th feasible subsystem extracted from (3.9), $i = 1, \ldots, s$. In order to cope with the drawbacks of the initialization that were discussed in Section 3.2.3, a procedure for the refinement of the estimates is presented in Table 3.3. It consists in a basic procedure (steps 2, 4, 5 and 6) whose aim is to iteratively improve both data classification and quality of fit by properly reassigning the data points, and updating the parameter estimates. The additional steps 1 and 3 allow to reduce the number of submodels by exploiting parameter similarities and cluster cardinalities. The refinement procedure returns the final number s of submodels, the parameter vectors θ_i, and the classification of the (feasible) data points into the clusters $\mathscr{D}_i$, $i = 1, \ldots, s$.

3.3.1 Dealing with undecidable data

As it was discussed in Section 3.2.3, there may exist data points $(y_k, \mathbf{x}_k)$ that are consistent with more than one submodel, *i.e.*, satisfying $|y_k - \varphi_k' \theta_i| \leq \delta$ for more than one $i = 1, \ldots, s$. These data points are termed *undecidable*. Undecidable data points could be classified correctly only by exploiting the partition of the regressor set, which is however not available at this stage of the identification process. When solving Problem 3.2 via the greedy approach described in Section 3.2.1, undecidable data points are classified depending on the order in which the feasible subsystems are extracted from (3.9). As an alternative, each undecidable data point $(y_k, \mathbf{x}_k)$ could be associated a posteriori to the submodel i^* such that the error is minimized, *i.e.*:

$$i^* = \arg \min_{i=1,\ldots,s} |y_k - \varphi_k' \theta_i| \tag{3.21}$$

Both criteria may lead to misclassifications when the partition of the regressor set is estimated (see Figure 3.6). Thus, in (Bemporad *et al.*, 2003*a*), undecidable data points were discarded during the classification procedure. This approach works well

GIVEN: α, β, γ, c

Set $t=1$ and $\theta_i^{(1)} = \Phi_p(\mathscr{D}_i^{(0)})$, $i=1,\dots,s$

1. **Merge submodels**

   Compute $(i^*, j^*) = \arg\min_{1\le i<j\le s} \mu(\theta_i^{(t)}, \theta_j^{(t)})$

   IF $\alpha_{i^*,j^*} \le \alpha$ THEN merge submodels i^* and j^*, and set $s = s-1$

2. **Data point reassignment**

   For each data point $(y_k, \mathbf{x}_k)$, $k=1,\dots,N$:

   - IF $\left|y_k - \varphi_k'\theta_i^{(t)}\right| \le \delta$ for only one $i=1,\dots,s$ THEN assign $(y_k,\mathbf{x}_k)$ to $\mathscr{D}_i^{(t)}$ and mark it as *feasible*
   - IF $\left|y_k - \varphi_k'\theta_i^{(t)}\right| \le \delta$ for more than one $i=1,\dots,s$ THEN mark $(y_k,\mathbf{x}_k)$ as *undecidable*
   - OTHERWISE mark $(y_k,\mathbf{x}_k)$ as *infeasible*

3. **Discard submodels**

   Compute $i^* = \arg\min_{i=1,\dots,s} \#\mathscr{D}_i^{(t)}/N$

   IF $\beta_{i^*} \le \beta$ THEN discard submodel i^*, set $s=s-1$ and go to step 2

4. **Assignment of undecidable data points**

   For each undecidable data point $(y_k,\mathbf{x}_k)$:

   Compute $\mathscr{C}_i(\mathbf{x}_k)$, $i=1,\dots,s$, and $i^* = \arg\max_{i=1,\dots,s} \#\mathscr{C}_i(\mathbf{x}_k)$

   IF $\left|y_k - \varphi_k'\theta_{i^*}^{(t)}\right| \le \delta$ THEN assign $(y_k,\mathbf{x}_k)$ to $\mathscr{D}_{i^*}^{(t)}$ and mark it as *feasible*

5. **Parameter estimation**

   Compute $\theta_i^{(t+1)} = \Phi_p(\mathscr{D}_i^{(t)})$, $i=1,\dots,s$

6. **Termination**

   IF $\left\|\theta_i^{(t+1)} - \theta_i^{(t)}\right\| \le \gamma \left\|\theta_i^{(t)}\right\|$ for all $i=1,\dots,s$

   THEN RETURN s, $\theta_i = \theta_i^{(t+1)}$ and $\mathscr{D}_i = \mathscr{D}_i^{(t)}$, $i=1,\dots,s$

   ELSE set $t=t+1$ and go to step 1

Table 3.3 Algorithm for the refinement of the estimates

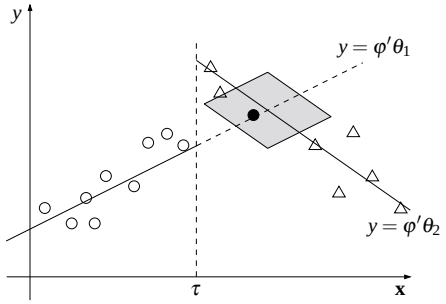


Figure 3.6 PWA model with two discrete modes, $\mathbf{x} \in \mathbb{R}$. The gray set represents the region of all possible undecidable data points for a fixed δ. By applying both the greedy approach to Problem 3.2, and (3.21), the only undecidable data point in the data set (the black circle) is associated to the first submodel. This yields two non-linearly separable clusters of points. If the switching point τ, defining the partition of the **x**-axis, was known, the undecidable data point could be correctly associated to the second submodel.

in many cases. However, when a large number of undecidable data points shows up, a lot of information useful for identification and contained in the discarded data, is actually not used. Hence, a modification to the classification procedure is here proposed in order to associate undecidable data points to submodels by exploiting spatial localization. This improves both the data classification (in regard to the estimation of the regions) and the parameter estimates.

The basic procedure for the refinement of the estimates consists of four steps (steps 2, 4, 5 and 6 in Table 3.3) to be iterated. Initial parameter estimates for each submodel are computed as $\theta_i^{(1)} = \Phi_p(\mathscr{D}_i^{(0)})$, $i=1,\dots,s$, where $\Phi_p(\cdot)$ denotes the projection estimator (3.8). At each iteration indexed by $t=1,2,\dots$, in step 2 all data points are processed, and classified as *feasible*, *infeasible* or *undecidable* according to the current estimated parameter vectors $\theta_i^{(t)}$, $i=1,\dots,s$. A feasible data point $(y_k,\mathbf{x}_k)$ satisfies the complementary inequality

$$\left|y_k - \varphi_k'\theta_i^{(t)}\right| \le \delta \tag{3.22}$$

for only one $i=1,\dots,s$, say i^*. Hence, it can be uniquely associated to the i^*-th submodel, and assigned to the corresponding cluster $\mathscr{D}_{i^*}^{(t)}$. The classification of the

feasible data points induces also a classification of the (feasible) regression vectors $\mathbf{x}_k$ into the clusters:

$$\mathscr{F}_i^{(t)} = \left\{ \mathbf{x}_k \,\middle|\, (y_k, \mathbf{x}_k) \in \mathscr{D}_i^{(t)} \right\}, \quad i = 1, \ldots, s$$

Infeasible data points do not satisfy (3.22) for any $i = 1, \ldots, s$. Undecidable data points satisfy (3.22) for more than one $i = 1, \ldots, s$, *i.e.*, they are consistent with more than one submodel. Step 4 tries to solve this ambiguity by exploiting spatial localization in the regressor set. For each undecidable data point $(y_k, \mathbf{x}_k)$, the set $\mathscr{C}(\mathbf{x}_k)$ of the c feasible regression vectors that are closest to $\mathbf{x}_k$, is computed. Here, c is a fixed positive integer, and the Euclidean distance is used. The feasible points around $\mathbf{x}_k$ are indeed expected to provide useful information for correctly classifying the undecidable data point $(y_k, \mathbf{x}_k)$. A set $\mathscr{C}(\mathbf{x}_k)$ may in principle collect regression vectors from different clusters $\mathscr{F}_i^{(t)}$. Hence, the clusters $\mathscr{C}_i(\mathbf{x}_k) = \mathscr{C}(\mathbf{x}_k) \bigcap \mathscr{F}_i^{(t)}$, $i = 1, \ldots, s$, are computed, and the index i^* such to maximize the cardinality of $\mathscr{C}_i(\mathbf{x}_k)$, $i = 1, \ldots, s$, is considered, *i.e.*:

$$i^* = \arg \max_{i=1,\ldots,s} \#\mathscr{C}_i(\mathbf{x}_k)$$

If the undecidable data point $(y_k, \mathbf{x}_k)$ satisfies $\left| y_k - \varphi_k' \theta_{i^*}^{(t)} \right| \leq \delta$, then it is associated to the i^*-th submodel and assigned to $\mathscr{D}_{i^*}^{(t)}$, otherwise it is left undecidable.

New parameter estimates for each submodel are computed in step 5 by using the projection estimator (3.8), *i.e.*, $\theta_i^{(t+1)} = \Phi_p(\mathscr{D}_i^{(t)})$, $i = 1, \ldots, s$. The use of the projection estimate is favorable because it guarantees that no feasible data point at refinement t becomes infeasible at refinement $t+1$, since:

$$\max_{(y_k, \mathbf{x}_k) \in \mathscr{D}_i^{(t)}} \left| y_k - \varphi_k' \theta_i^{(t+1)} \right| \leq \max_{(y_k, \mathbf{x}_k) \in \mathscr{D}_i^{(t)}} \left| y_k - \varphi_k' \theta_i^{(t)} \right| \leq \delta \,, \quad i = 1, \ldots, s$$

In step 6 the termination condition is checked. Given a tolerance $\gamma > 0$, if the new and the current parameter vectors satisfy:

$$\frac{\left\| \theta_i^{(t+1)} - \theta_i^{(t)} \right\|}{\left\| \theta_i^{(t)} \right\|} \leq \gamma, \quad \forall i = 1, \ldots, s$$

then the iterations are stopped. However, in order to avoid that the procedure does not terminate, a maximum number $t_{\max}$ of refinements can be predefined.

It is evident that the proposed refinement procedure relies on the distinction among infeasible, undecidable, and feasible data points, and the alternation between data point reassignment and parameter update. Infeasible data points are not consistent with any submodel. If the corresponding violations are large, they are expected to be outliers. Neglecting them in the parameter estimation helps to improve the quality of fit. On the other hand, infeasible data points with a small violation may be recovered to be feasible iteration by iteration as the parameter estimates are updated, thus improving the quality of the classification. Also undecidable data points may be recovered to be feasible in step 4. Here, good choices for the parameter c depend on the density of the data set. In general, c should not be chosen too small, in order to avoid that the sets $\mathscr{C}(\mathbf{x}_k)$ do not contain enough points for correct classification. On the other hand, for large values of c, a set $\mathscr{C}(\mathbf{x}_k)$ might contain points distant from $\mathbf{x}_k$. In this case, the data point $(y_k, \mathbf{x}_k)$ could be badly assigned to a "far" cluster, or left undecidable. Indeed, if many data points are still classified as undecidable at the exit of the refinement procedure, it is likely that c was chosen too large. Neglecting the undecidable data points will help to reduce the number of misclassifications when estimating the partition of the regressor set (see Section 4.5).

3.3.2 Reducing the number of submodels

The basic procedure for the refinement of the estimates described in Section 3.3.1, does not change the number of submodels. Hence, additional steps are required to cope with the case when the initialization provides an overestimation of the minimum number of submodels needed to fit the data, as discussed in Section 3.2.3.

In order to reduce the number of submodels, one can exploit parameter similarities and cluster cardinalities. Two submodels characterized by similar parameter vectors can be merged in step 1. Here, the quantity

$$\mu(\theta_1, \theta_2) \triangleq \frac{\|\theta_1 - \theta_2\|}{\min\{\|\theta_1\|, \|\theta_2\|\}}$$

is used as a measure of the similarity of two vectors θ_1, $\theta_2 \in \mathbb{R}^{n+1}$, and two close

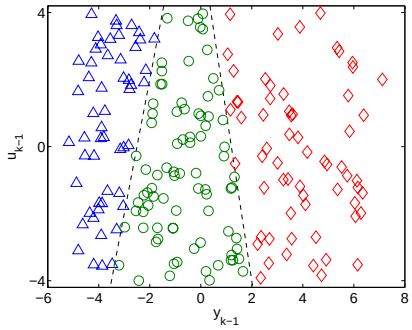


Figure 3.7 Classification of the regression vectors (*triangles*, *circles*, *diamonds*) for Example 3.1 after the refinement. The dashed lines represent the true partition of the regressor set, which is assumed to be unknown. All data points are correctly classified

submodels i^* and j^* are combined at iteration t by computing the joined parameter vector as $\Phi_p(\mathscr{D}_{i^*}^{(t-1)} \bigcup \mathscr{D}_{j^*}^{(t-1)})$. Note that, when two parameter vectors are very similar, a large number of undecidable data points might show up in step 2. On the other hand, if the cardinality of a cluster of feasible data points is too small, the corresponding submodel (which accounts only for few data) can be discarded in step 3. The positive thresholds α and β in steps 1 and 3 should be suitably chosen in order to reduce the number of submodels still preserving a good fit of the data. For too large values of α and β, a large number of infeasible data points typically will show up as the number of submodels decreases and some significant submodel is neglected. One could use this information in order to adjust α and β, and then repeat the refinement.

Example 3.1 (cont'd**)** The classification results shown in Figure 3.7 were obtained by applying the (basic) refinement procedure after the initialization using the modified version of the greedy randomized algorithm for Problem 3.2. All data points were correctly classified after the refinement: compare Figures 3.4 and 3.7. In particular, all undecidable data points were correctly associated to submodels by exploiting spatial localization in the regressor set.

In order to better illustrate the main features of the refinement procedure, it is

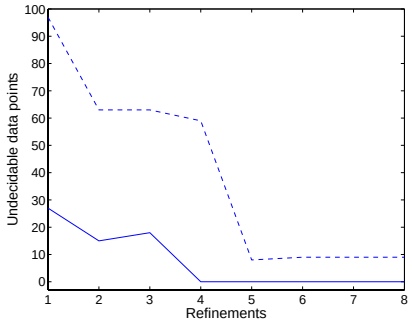


Figure 3.8 The number of undecidable data points versus the number of refinements for Example 3.1. The dashed line represents the number of undecidable data points in step 2 of the refinement procedure, and the solid line represents the number of data points that are left undecidable after step 4

here described more in detail its application after the initialization using the original version of the greedy randomized algorithm for Problem 3.2. By choosing $\alpha = 10\%$, $\beta = 1.5\%$, $\gamma = 0.001\%$ and $c = 5$, the termination condition was reached after 8 refinements, and the number s of submodels decreased from 6 to 3, which equals the true one. Two submodels accounting for too few data points (situation shown in Figure 3.3(d)) were immediately discarded at refinement 1, whereas two submodels with similar parameter vectors (corresponding to a situation like in Figure 3.3(c)) were merged at refinement 4. The final three clusters of regression vectors $\mathbf{x}_k$ are again shown in Figure 3.7. From left to right, they consist of 54, 83

Table 3.4 True ($\bar{\theta}_i$) and estimated (θ_i) parameter vectors for each submodel in Example 3.1

$\bar{\theta}_1$	θ_1	$\bar{\theta}_2$	θ_2	$\bar{\theta}_3$	θ_3
-0.4	-0.3961	0.5	0.5018	-0.3	-0.2989
1	0.9903	-1	-0.9980	0.5	0.5045
1.5	1.5472	-0.5	-0.4994	-1.7	-1.7072

and 63 points, respectively. All data points were correctly classified, and no data point was left undecidable or infeasible. The parameter vectors estimated for each submodel are shown in Table 3.4, and provide very good estimates of the true ones. Figure 3.8 shows a plot of the number of undecidable data points versus the number of refinements (solid line). A significant decrease of the number of undecidable data points occurred at refinement 4, when two submodels with similar parameter vectors were merged. The dashed line in the same figure represents the number of undecidable data points in step 2 of the refinement procedure, *i.e.*, before they are associated to submodels in step 4, if possible. Note that step 4 is really effective in reducing the number of undecidable data points, which is initially about 50% of all the available data points.

3.4 Multi-output models

In this section, the PWA system identification problem for single-output models considered in Section 3.1 will be extended to multi-output models. It is here assumed that a collection of N samples $(\mathbf{y}_k, \mathbf{x}_k)$, $k = 1, \ldots, N$, is given, where $\mathbf{y}_k \in \mathbb{R}^q$ represent values of a nonlinear map $F : \mathscr{X} \to \mathbb{R}^q$ obtained at certain points $\mathbf{x}_k \in \mathscr{X} \subseteq \mathbb{R}^n$:

$$\mathbf{y}_k = F(\mathbf{x}_k) + \mathbf{e}_k, \quad k = 1, \ldots, N$$

where $\mathbf{e}_k \in \mathbb{R}^q$ is either noise in system identification or an error term in function approximation (see Remark 3.1). The objective is to find a PWA approximation f of F that matches as good as possible the given data points $(\mathbf{y}_k, \mathbf{x}_k)$, $k = 1, \ldots, N$, according to a specified criterion of fit. The PWA map f is defined as:

$$f(\mathbf{x}) = \begin{cases} \Theta_1' \varphi & \text{if} \quad \mathbf{x} \in \mathscr{X}_1 \\ \vdots & \vdots \\ \Theta_s' \varphi & \text{if} \quad \mathbf{x} \in \mathscr{X}_s \end{cases}, \quad \varphi = [\, \mathbf{x}' \; 1 \,]' \tag{3.23}$$

where $\Theta_i \in \mathbb{R}^{(n+1)\times q}$, $i = 1, \ldots, s$, are matrices of parameters, and $\{\mathscr{X}_i\}_{i=1}^s$ is a collection of polyhedral regions like (3.5), which form a complete partition of $\mathscr{X}$.

For instance, the problem of identifying a PWA model of the nonlinear system in state-space form:

$$\zeta_{k+1} = F(\zeta_k, \mathbf{u}_k) + \mathbf{e}_k$$

where $k \in \mathbb{Z}$ is time, ζ_k is the state, $\mathbf{u}_k$ is the input, and $\mathbf{e}_k$ is additive noise, can be easily cast into this framework, if the state vector is measurable, by defining $\mathbf{x}_k = [\, \zeta_k' \; \mathbf{u}_k' \,]'$ and $\mathbf{y}_k = \zeta_{k+1}$.

When one looks for a PWA approximation (3.23) of F, there are two alternative approaches that can be taken. Letting $F = (F_1, \ldots, F_q)$, the first approach consists in estimating a PWA approximation f_ℓ for each scalar function F_ℓ, $\ell = 1, \ldots, q$. The PWA map (3.23) is then defined as $f = (f_1, \ldots, f_q)$. However, this simple approach leads in general to a larger number s of regions than necessary, since the regions $\mathscr{X}_i$, $i = 1, \ldots, s$, are constructed by intersecting the partitions of the functions f_ℓ, $\ell = 1, \ldots, q$. The second approach requires to directly estimate a PWA approximation (3.23) of F. To this aim, the idea proposed in Section 3.1 (*i.e.*, to characterize the model by its maximum fitting error) can be here extended by requiring:

$$\|\mathbf{y}_k - f(\mathbf{x}_k)\|_\infty \leq \delta, \quad k = 1, \ldots, N$$

for a fixed $\delta > 0$. Then, Problem 3.1 can be straightforward restated, and an estimation procedure following the same steps as proposed in Sections 3.2 and 3.3 for single-output models, can be adopted. By relaxing the estimation of the regions as in Section 3.2, Problem 3.2 here becomes:

Problem 3.3 *Given $\delta > 0$ and the system of N multiple linear complementary inequalities:*

$$\left\|\mathbf{y}_k - \Theta' \varphi_k\right\|_\infty \leq \delta, \quad k = 1, \ldots, N \tag{3.24}$$

find a partition of (3.24) into a minimum number of feasible subsystems.

The unknown in (3.24) is the matrix $\Theta \in \mathbb{R}^{(n+1)\times q}$. The solution of Problem 3.3 can be tackled by suitably amending the greedy randomized algorithm for Problem 3.2 described in Sections 3.2.1 and 3.2.2. In particular, the greedy algorithm shown in Table 3.1 should be modified by replacing θ with Θ, and the inequalities of the type

$|y_k - \varphi_k'\theta| \le \delta$ with $\|\mathbf{y}_k - \Theta'\varphi_k\|_\infty \le \delta$. The same should be done for the algorithm shown in Table 3.2. A more detailed explanation is only needed for what concerns the update of the current solution $\Theta^{(j)}$ at iteration j. By letting

$$\Theta = \left[\, \theta_1 \,\dots\, \theta_q \,\right] \quad \text{and} \quad \mathbf{y}_k = \left[\, y_{k,1} \,\dots\, y_{k,q} \,\right]'$$

with $\theta_\ell \in \mathbb{R}^{n+1}$ and $y_{k,\ell} \in \mathbb{R}$, $\ell = 1,\dots,q$, and noting that each inequality (3.24) is equivalent to the set of complementary inequalities:

$$\begin{cases} |y_{k,1} - \varphi_k'\theta_1| \le \delta \\ \vdots \\ |y_{k,q} - \varphi_k'\theta_q| \le \delta \end{cases}$$

each column $\theta_\ell^{(j)}$, $\ell = 1,\dots,q$, forming the current estimate $\Theta^{(j)}$ can be independently updated as in Table 3.2 by considering only the violation of the corresponding complementary inequality. In the final IF statement, the notation $\bar{\Theta} = \Phi_p(\bar{\mathscr{D}})$ should intend that each column $\bar{\theta}_\ell$, $\ell = 1,\dots,q$, forming $\bar{\Theta}$ is computed by using the projection estimator (3.8), *i.e.*:

$$\bar{\theta}_\ell = \arg\min_{\theta} \max_{(\mathbf{y}_k,\mathbf{x}_k)\in\bar{\mathscr{D}}} \left| y_{k,\ell} - \varphi_k'\theta \right|, \quad \ell = 1,\dots,q$$

Also the refinement procedure shown in Table 3.3 can be straightforward modified as the algorithms before. The details, of ready implementation, are omitted. The estimation of the regions can be finally carried out as it will be described in the next chapter.

4

Estimation of the regions

In this chapter, the final step of the proposed PWA system identification procedure will be addressed. This step consists in estimating the partition of the regressor set for the identified PWARX model. After introducing the problem in Section 4.1, several approaches to two-class and multi-class linear separation, both in the separable and the inseparable case, will be reviewed. Lastly, in Section 4.5 these approaches will be applied to region estimation in PWA system identification.

4.1 The linear separation problem

Given the classification of the (feasible) data points into the clusters $\mathscr{D}_i$, $i = 1,\dots,s$, that is returned by the refinement procedure described in Section 3.3, let

$$\mathscr{F}_i = \left\{ \mathbf{x}_k | (y_k, \mathbf{x}_k) \in \mathscr{D}_i \right\}, \quad i = 1,\dots,s \tag{4.1}$$

be the corresponding clusters of regression vectors. Region estimation for the identified PWARX model (3.2)–(3.4) consists in finding a complete partition $\left\{ \mathscr{X}_i \right\}_{i=1}^{s}$ of the regressor set $\mathscr{X}$ such that, if $\mathbf{x}_k \in \mathscr{F}_i$, then $\mathbf{x}_k \in \mathscr{X}_i$. The polyhedral regions are defined by sets of linear inequalities as follows:

$$\mathscr{X}_i = \left\{ \mathbf{x} \in \mathbb{R}^n \,\middle|\, H_i \varphi \preceq \mathbf{0} \right\}, \quad \varphi = \left[\, \mathbf{x}' \; 1 \,\right]' \tag{4.2}$$

where $H_i \in \mathbb{R}^{q_i \times (n+1)}$, $i = 1, \ldots, s$. Hence, the above problem is equivalent to that of separating s sets of points by means of linear classifiers (hyperplanes), which has been extensively investigated in different fields (*e.g.*, machine learning, operations research).

The problem of linearly separating s sets $\mathscr{A}_1, \ldots, \mathscr{A}_s$ of points in $\mathbb{R}^n$ can be tackled in two different ways:

a) Consider pairwise the sets $\mathscr{A}_i$, and construct a linear classifier for each pair $(\mathscr{A}_i, \mathscr{A}_j)$, with $i \neq j$.

b) Consider all the sets $\mathscr{A}_i$ at the same time, and construct a piecewise linear classifier which is able to discriminate among s classes.

The first approach consists in finding a hyperplane that separates the convex hull of $\mathscr{A}_i$ from the convex hull of $\mathscr{A}_j$, for any $i \neq j$. This amounts to solve $s(s-1)/2$ *two-class* linear separation problems. Two-class linear separation will be addressed in Sections 4.2 and 4.3. According to basic results in Statistical Learning Theory (Vapnik, 1998), a convenient way to accomplish this task is to employ a *Support Vector Machine* (SVM) (Cortes and Vapnik, 1995) with a linear kernel. SVMs solve the problem of finding the optimal two-class linear discriminant, *i.e.*, the one which maximizes the separation margin. This problem can be posed as a quadratic program with linear constraints. The resulting linear discriminant is known as support vector machine because it is a function of a subset of the data points known as *support vectors*. In order to reduce the computational complexity, alternative approaches, which are based on linear programming, can also be employed. The resulting methods are sometimes referred to as *Robust Linear Programming* (RLP) (Bennett and Mangasarian, 1992). Note that the assumption that there exists a hyperplane separating without errors the points in $\mathscr{A}_i$ from those in $\mathscr{A}_j$, might be not satisfied in practice. For instance, two clusters of regression vectors $\mathscr{F}_i$ and $\mathscr{F}_j$ defined by (4.1) could have intersecting convex hulls due to errors in classifying the data points. However, in such a case, both SVM and RLP look for a separating hyperplane that additionally minimizes a weighted sum of the misclassification errors.

Alternatively, one could look for a separating hyperplane that minimizes the number of misclassified points. This problem amounts to find a feasible subsystem with maximum cardinality of an infeasible system of linear inequalities. The *Maximum Feasible Subsystem* (MAX FS) problem is well studied (Amaldi and Kann, 1995), and has many interesting applications besides machine learning. In spite of its inherent computational complexity, several heuristics have been developed which perform well in practice. See, *e.g.*, (Pfetsch, 2002) and references therein.

The former approach, based on the estimation of each separating hyperplane, is computationally appealing. It does not involve all the data at the same time, and amounts to solve either simple linear/quadratic programs, or MAX FS problems for whose solution efficient heuristics exist. A major drawback is that the estimated regions are not guaranteed to form a complete partition of the domain, *i.e.*, the union of the regions might not cover the whole domain (see Section 4.5). If the presence of "holes" in the partition is not acceptable, the second approach can be employed. It consists in solving a *multi-class* linear separation problem, in which a piecewise linear classifier is constructed as the maximum of s linear classification functions. Multi-class linear separation will be addressed in Section 4.4.

A first way to tackle the multi-class problem (Vapnik, 1995; Bredensteiner and Bennett, 1999) is to compute the s linear classifiers by separating each set $\mathscr{A}_i$ from the union of all the others. This requires the solution of s two-class linear separation problems. Unless each set $\mathscr{A}_i$ is linearly separable from the union of the remaining sets, this approach has the drawback that multiply classified points or unclassified points may occur, when all s classifiers are applied to the original data set. This ambiguity is avoided by assigning a point to the class corresponding to the classification function that is maximal at that point. A second way to tackle the multi-class problem is to directly construct s classification functions such that, at each data point, the corresponding class function is maximal. Classical two-class separation methods such as SVM and RLP have been extended to this multi-class case (Bredensteiner and Bennett, 1999; Bennett and Mangasarian, 1994). The resulting methods are called *Multicategory* SVM (M-SVM) or *Multicategory* RLP

(M-RLP), to stress their ability of dealing with problems involving more than two classes. Quite notably, a single linear or quadratic program (involving all available data) is still used to construct the piecewise linear classifier.

4.2 Two-class linear separation: the separable case

Let $\mathscr{A}_1$ and $\mathscr{A}_2$ be two sets of points in $\mathbb{R}^n$ with cardinality m_1 and m_2, respectively. The objective of *linear separation* is to find a pair $(\mathbf{w},\gamma)$, with $\mathbf{w}\in\mathbb{R}^n$ and $\gamma\in\mathbb{R}$, such that the following condition is satisfied:

$$\begin{aligned}\mathbf{w}'\mathbf{x}+\gamma>0 \quad &\text{if } \mathbf{x}\in\mathscr{A}_1\\ \mathbf{w}'\mathbf{x}+\gamma<0 \quad &\text{if } \mathbf{x}\in\mathscr{A}_2\end{aligned} \tag{4.3}$$

Geometrically, the pair $(\mathbf{w},\gamma)$ defines the hyperplane:

$$\mathscr{H}=\left\{\mathbf{x}\in\mathbb{R}^n \,\middle|\, \mathbf{w}'\mathbf{x}+\gamma=0\right\} \tag{4.4}$$

i.e., an affine subspace with dimension $n-1$ which divides the n-dimensional space into two half-spaces. Condition (4.3) requires that all points belonging to the same set lie on the same side of the hyperplane, *i.e.*, in the same half-space (see Figure 4.1). Note that the *normal* $\mathbf{w}$ and the *bias* γ defining the hyperplane (4.4) are not unique. The pairs $(\mathbf{w},\gamma)$ and $(\lambda\mathbf{w},\lambda\gamma)$ define the same hyperplane for any $\lambda\neq 0$. For simplicity of notation, let $\mathscr{A}_1\cup\mathscr{A}_2=\{\mathbf{x}_1,\ldots,\mathbf{x}_m\}$, where $m=m_1+m_2$, and define the target values z_k, $k=1,\ldots,m$, as follows:

$$z_k=\begin{cases}1 & \text{if } \mathbf{x}_k\in\mathscr{A}_1\\ -1 & \text{if } \mathbf{x}_k\in\mathscr{A}_2\end{cases} \tag{4.5}$$

Hence, condition (4.3) can be rewritten in the compact form:

$$z_k[\mathbf{w}'\mathbf{x}_k+\gamma]>0\,,\quad k=1,\ldots,m \tag{4.6}$$

The two sets of points $\mathscr{A}_1$ and $\mathscr{A}_2$ are said to be *linearly separable* if there exist $\mathbf{w}\in\mathbb{R}^n$ and $\gamma\in\mathbb{R}$ such that condition (4.6) holds. In such a case, the hyperplane defined by (4.4) is called a *separating hyperplane*.

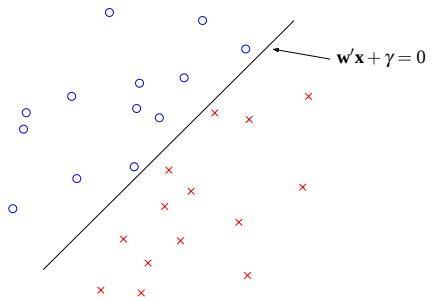


Figure 4.1 Two linearly separable sets and a separating hyperplane

Remark 4.1 It is worth to note that the problem of linear separation can be seen as a particular case of binary classification, in which a real-valued function $f:\mathbb{R}^n\to\mathbb{R}$ is used to discriminate between two classes by assigning the input $\mathbf{x}$ to the positive class if $f(\mathbf{x})\geq 0$, and otherwise to the negative class. That is, the decision rule is given by the sign of $f(\mathbf{x})$. In two-class linear discrimination, $f(\mathbf{x})$ is a linear function, *i.e.*, $f(\mathbf{x})=\mathbf{w}'\mathbf{x}+\gamma$, where $(\mathbf{w},\gamma)\in\mathbb{R}^n\times\mathbb{R}$ are the parameters that control the function. In *Supervised Learning* these parameters must be learned from available input-output data. □

If the two sets $\mathscr{A}_1$ and $\mathscr{A}_2$ are linearly separable, there exist infinitely many hyperplanes that separate them correctly. In principle any of these could be chosen, but in practice a good choice is the separating hyperplane such that the distance of the closest point in the data set to the hyperplane is maximized (see Remark 4.2). For a given norm $\|\cdot\|$ on $\mathbb{R}^n$, the distance $d(\mathbf{x}_0;\mathscr{H})$ of a point $\mathbf{x}_0\in\mathbb{R}^n$ to the hyperplane (4.4) is defined as:

$$d(\mathbf{x}_0;\mathscr{H})\triangleq\min_{\mathbf{x}\in\mathscr{H}}\|\mathbf{x}-\mathbf{x}_0\| \tag{4.7}$$

which can be interpreted as the distance between $\mathbf{x}_0$ and its projection onto the hyperplane. In (Mangasarian, 1999) it is shown that definition (4.7) can be rewritten in terms of the dual norm $\|\cdot\|^{\#}$ of $\|\cdot\|$:

$$d(\mathbf{x}_0;\mathscr{H})=\frac{|\mathbf{w}'\mathbf{x}_0+\gamma|}{\|\mathbf{w}\|^{\#}} \tag{4.8}$$

where, for $\mathbf{x} \in \mathbb{R}^n$, $\|\mathbf{x}\|^{\#} \triangleq \max_{\|\mathbf{y}\|=1} \mathbf{x}'\mathbf{y}$. Examples of distance of a point to a hyperplane that will be considered in the following, are[1]:

- ℓ_1-norm distance

$$d_1(\mathbf{x}_0;\mathscr{H}) = \min_{\mathbf{x}\in\mathscr{H}} \|\mathbf{x}-\mathbf{x}_0\|_1 = \frac{|\mathbf{w}'\mathbf{x}_0+\gamma|}{\|\mathbf{w}\|_\infty} \tag{4.9}$$

- ℓ_2-norm distance

$$d_2(\mathbf{x}_0;\mathscr{H}) = \min_{\mathbf{x}\in\mathscr{H}} \|\mathbf{x}-\mathbf{x}_0\|_2 = \frac{|\mathbf{w}'\mathbf{x}_0+\gamma|}{\|\mathbf{w}\|_2} \tag{4.10}$$

- ℓ_∞-norm distance

$$d_\infty(\mathbf{x}_0;\mathscr{H}) = \min_{\mathbf{x}\in\mathscr{H}} \|\mathbf{x}-\mathbf{x}_0\|_\infty = \frac{|\mathbf{w}'\mathbf{x}_0+\gamma|}{\|\mathbf{w}\|_1} \tag{4.11}$$

Two sets $\mathscr{A}_1$ and $\mathscr{A}_2$ are said to be *optimally separated* by the hyperplane (4.4) if they are separated without errors and the distance between the closest point in the data set to the hyperplane is maximal. For a given distance (4.7), finding the *optimal separating hyperplane* amounts to solve the following optimization problem:

$$\begin{cases} \max\limits_{\mathbf{w},\gamma} & \min\limits_{k=1,\ldots,m} d(\mathbf{x}_k;\mathscr{H}) \\ s.t. & z_k[\mathbf{w}'\mathbf{x}_k+\gamma] > 0 \quad k=1,\ldots,m \end{cases} \tag{4.12}$$

Problem (4.12) has not a unique solution. In particular, if the pair $(\mathbf{w}^*,\gamma^*)$ is optimal for problem (4.12), then the pair $(\lambda\mathbf{w}^*,\lambda\gamma^*)$ is also optimal for any $\lambda > 0$. Without loss of generality, it is therefore appropriate to restrict the attention to hyperplanes in Vapnik's *canonical* form (Vapnik, 1995), for which the parameters $\mathbf{w}$ and γ are constrained by:

$$\min_{k=1,\ldots,m} \left|\mathbf{w}'\mathbf{x}_k+\gamma\right| = 1 \tag{4.13}$$

Constraint (4.13) is very incisive in simplifying the formulation of the *Optimal Separation* problem (4.12). For any arbitrary norm $\|\cdot\|$ on $\mathbb{R}^n$, it implies that:

$$\min_{k=1,\ldots,m} d(\mathbf{x}_k;\mathscr{H}) = \frac{\min\limits_{k=1,\ldots,m} \left|\mathbf{w}'\mathbf{x}_k+\gamma\right|}{\|\mathbf{w}\|^{\#}} = \frac{1}{\|\mathbf{w}\|^{\#}} \tag{4.14}$$

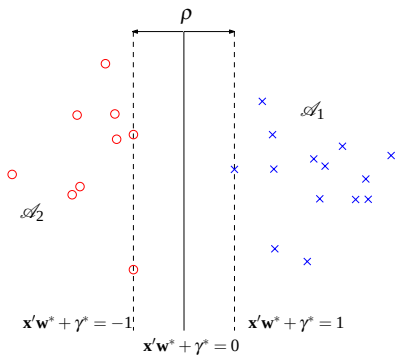


Figure 4.2 Optimal separation of two sets of points $\mathscr{A}1$ and $\mathscr{A}2$: optimal separating hyperplane (solid line), supporting hyperplanes (dashed lines), and margin of separation ρ

Hence, a hyperplane in canonical form is such that the distance of the closest point in the data set to the hyperplane is equal to the inverse of the dual norm of the normal vector $\mathbf{w}$. In view of (4.14), problem (4.12) can be easily rewritten as:

$$\begin{cases} \min\limits_{\mathbf{w},\gamma} & \|\mathbf{w}\|^{\#} \\ s.t. & z_k[\mathbf{w}'\mathbf{x}_k+\gamma] \geq 1 \quad k=1,\ldots,m \end{cases} \tag{4.15}$$

Note that it was possible to relax the constraint (4.13) in the formulation of problem (4.15) since it can be shown that at optimality there always exist $\mathbf{x}_i \in \mathscr{A}_1$ and $\mathbf{x}_l \in \mathscr{A}_2$ such that the corresponding constraints are active, *i.e*, $\mathbf{w}'\mathbf{x}_i+\gamma = 1$ and $\mathbf{w}'\mathbf{x}_l+\gamma = -1$. This implies that constraint (4.13) is automatically satisfied at optimality for problem (4.15), and the optimal separating hyperplane is equally distant from the closest point of each set. By defining the *margin of separation* between the sets $\mathscr{A}_1$ and $\mathscr{A}_2$ as:

$$\rho = \min_{k:\mathbf{x}_k\in\mathscr{A}_1} d(\mathbf{x}_k;\mathscr{H}) + \min_{k:\mathbf{x}_k\in\mathscr{A}_2} d(\mathbf{x}_k;\mathscr{H}) \tag{4.16}$$

it turns out that the optimal separating hyperplane also maximizes the margin of separation. If the pair $(\mathbf{w}^*,\gamma^*)$ is optimal for problem (4.15), *i.e.*, it defines the

[1] For $p,q \in [1,\infty]$, $\frac{1}{p}+\frac{1}{q}=1$, the ℓ_p-norm and the ℓ_q-norm are dual norms.

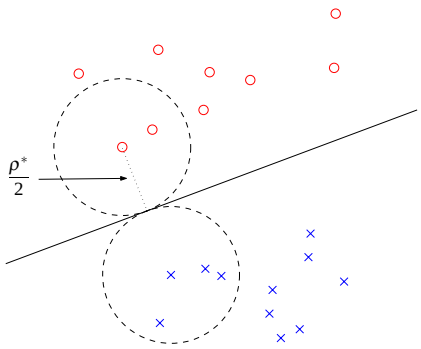


Figure 4.3 Optimal separating hyperplane using the ℓ_2-norm

optimal separating hyperplane:

$$\mathscr{H}_o = \left\{\mathbf{x} \in \mathbb{R}^n \,\middle|\, \mathbf{x}'\mathbf{w}^* + \gamma^* = 0\right\}$$

then the optimal margin of separation is $\rho^* = 2/\|\mathbf{w}^*\|^{\#}$. The parallel hyperplanes:

$$\mathscr{H}_+ = \left\{\mathbf{x} \in \mathbb{R}^n \,\middle|\, \mathbf{x}'\mathbf{w}^* + \gamma^* - 1 = 0\right\}$$
$$\mathscr{H}_- = \left\{\mathbf{x} \in \mathbb{R}^n \,\middle|\, \mathbf{x}'\mathbf{w}^* + \gamma^* + 1 = 0\right\}$$

are called *supporting hyperplanes* (see Figure 4.2), and all the points $\mathbf{x}_k$ lying on $\mathscr{H}_+$ or $\mathscr{H}_-$ are called *support vectors*.

In the following sections, the *Optimal Separation* problem (4.12) will be further specified for each distance (4.9), (4.10) and (4.11). It will be shown that the ℓ_1-norm and the ℓ_∞-norm formulation lead to linear programs, whereas the ℓ_2-norm formulation leads to a quadratic program.

Remark 4.2 Using either the ℓ_1-norm, the ℓ_2-norm or the ℓ_∞-norm formulation determines, in general, different optimal hyperplanes, margins and support vectors. However, research on learning machines found no empirical evidence that one norm is preferable to the others in terms of generalization. In particular, the effects on Statistical Learning Theory caused by changing norms are an open question.

Statistical Learning Theory addresses mathematically the problem of how to best construct functions that will generalize well on future points. According to structural risk minimization, maximizing the margin is essential for good generalization. Larger margins should lead to better generalizations and prevent overfitting. □

4.2.1 Optimal separation using the ℓ_2-norm

From (4.10) and (4.15), the *Optimal Separation* problem using the ℓ_2-norm becomes:

$$\begin{cases} \min\limits_{\mathbf{w},\gamma} & \|\mathbf{w}\|_2 \\ s.t. & z_k[\mathbf{w}'\mathbf{x}_k + \gamma] \geq 1 \quad k = 1,\ldots,m \end{cases} \tag{4.17}$$

By taking into account that $\|\mathbf{w}\|_2^2 = \sum_{i=1}^{n} w_i^2$, problem (4.17) can be equivalently rewritten as the quadratic program:

$$\begin{cases} \min\limits_{\mathbf{w},\gamma} & \frac{1}{2}\sum_{i=1}^{n} w_i^2 \\ s.t. & z_k[\mathbf{w}'\mathbf{x}_k + \gamma] \geq 1 \quad k = 1,\ldots,m \end{cases} \tag{4.18}$$

Figure 4.3 shows an example of optimal separation using the ℓ_2-norm. The optimal margin ρ^* is twice the ℓ_2-norm distance of the support vectors to the optimal separating hyperplane.

In order to better understand the role played by the support vectors in the solution of problem (4.18), define the *Lagrangian* associated with problem (4.18) as:

$$L(\mathbf{w},\gamma,\lambda) = \frac{1}{2}\sum_{i=1}^{n} w_i^2 - \sum_{k=1}^{m} \lambda_k \left(z_k[\mathbf{w}'\mathbf{x}_k + \gamma] - 1\right)$$

where $\lambda = (\lambda_1,\ldots,\lambda_m)$ are the Lagrange multipliers (or dual variables). Then, define the *dual function* $g(\lambda)$ as the minimum value of the Lagrangian with respect to $(\mathbf{w},\gamma)$, *i.e.*, $g(\lambda) = \inf_{\mathbf{w},\gamma} L(\mathbf{w},\gamma,\lambda)$. It is easy to verify that:

$$g(\lambda) = \begin{cases} -\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} z_i z_j \mathbf{x}_i'\mathbf{x}_j \lambda_i \lambda_j + \sum_{k=1}^{m} \lambda_k & \text{if } \sum_{k=1}^{m} z_k \lambda_k = 0 \\ -\infty & \text{otherwise} \end{cases}$$

where the minimum of $L(\mathbf{w},\gamma,\lambda)$ with respect to $(\mathbf{w},\gamma)$ when $\sum_{k=1}^{m} z_k\lambda_k = 0$, is attained by:

$$\mathbf{w} = \sum_{k=1}^{m} z_k\lambda_k\mathbf{x}_k \tag{4.19}$$

The *dual function* $g(\lambda)$ has to be maximized (equivalently, $-g(\lambda)$ has to be minimized) over $\lambda \succeq 0$, thus leading to the following *Lagrange dual problem* associated with problem (4.18):

$$\begin{cases} \min\limits_{\lambda} & \frac{1}{2}\sum\limits_{i=1}^{m}\sum\limits_{j=1}^{m} z_i z_j \mathbf{x}_i'\mathbf{x}_j\lambda_i\lambda_j - \sum\limits_{k=1}^{m}\lambda_k \\ s.t. & \sum\limits_{k=1}^{m} z_k\lambda_k = 0 \\ & \lambda_k \geq 0 \quad k = 1,\ldots,m \end{cases} \tag{4.20}$$

Note that the dual of problem (4.18) is still a quadratic program. Let the pair $(\mathbf{w}^*,\gamma^*)$ be optimal for the primal problem (4.18), and let λ^* be optimal for the dual problem (4.20). Since strong duality holds in this case, the value of $\mathbf{w}^*$ can be retrieved from (4.19):

$$\mathbf{w}^* = \sum_{k=1}^{m} z_k\lambda_k^*\mathbf{x}_k \tag{4.21}$$

At optimality complementary slackness implies:

$$\lambda_k^*\left(z_k[\mathbf{x}_k'\mathbf{w}^* + \gamma^*] - 1\right) = 0, \quad k = 1,\ldots,m$$

and hence only points $\mathbf{x}_k$ that are support vectors, *i.e.*, satisfying $z_k[\mathbf{x}_k'\mathbf{w}^* + \gamma^*] = 1$, will have non-zero Lagrange multipliers. This in turn implies that only support vectors will contribute to equation (4.21). By defining the set of indices:

$$SV = \left\{k = 1,\ldots,m \;\middle|\; \mathbf{x}_k \text{ is a support vector}\right\}$$

the optimal parameters $\mathbf{w}^*$ and γ^* can be finally computed as:

$$\mathbf{w}^* = \sum_{k\in SV} z_k\lambda_k^*\mathbf{x}_k \tag{4.22}$$

$$\gamma^* = \frac{\sum\limits_{k\in SV} z_k - \left(\sum\limits_{k\in SV}\mathbf{x}_k'\right)\mathbf{w}^*}{\#SV} \tag{4.23}$$

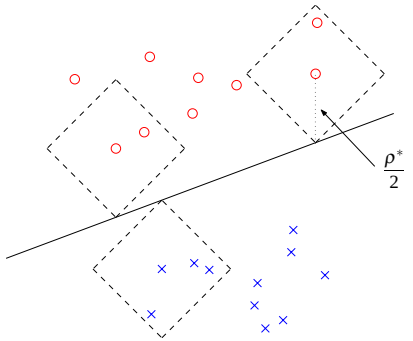


Figure 4.4 Optimal separating hyperplane using the ℓ_1-norm

Equations (4.22) and (4.23) show that the optimal separating hyperplane is determined only by the support vectors, that in general form a small subset of the data set (see again Figure 4.3). The other points could be removed from the data set, and recalculating the optimal separating hyperplane would produce the same result. In other words, the support vectors summarize all the information needed to optimally separate two linearly separable sets.

4.2.2 Optimal separation using the ℓ_1-norm

From (4.9) and (4.15), the *Optimal Separation* problem using the ℓ_1-norm becomes:

$$\begin{cases} \min\limits_{\mathbf{w},\gamma} & \|\mathbf{w}\|_\infty \\ s.t. & z_k[\mathbf{w}'\mathbf{x}_k + \gamma] \geq 1 \quad k = 1,\ldots,m \end{cases} \tag{4.24}$$

By recalling that $\|\mathbf{w}\|_\infty = \max\limits_{i=1,\ldots,n}|w_i|$, problem (4.24) can be easily rewritten as a linear program by introducing the nonnegative auxiliary variable s:

$$\begin{cases} \min\limits_{\mathbf{w},\gamma,s} & s \\ s.t. & z_k[\mathbf{w}'\mathbf{x}_k + \gamma] \geq 1 \quad k = 1,\ldots,m \\ & -s \leq w_i \leq s \qquad i = 1,\ldots,n \end{cases} \tag{4.25}$$

At optimality for problem (4.25), it holds $s = \|\mathbf{w}\|_\infty$. Figure 4.4 shows an example of optimal separation using the ℓ_1-norm for the same data set as in Figure 4.3. The

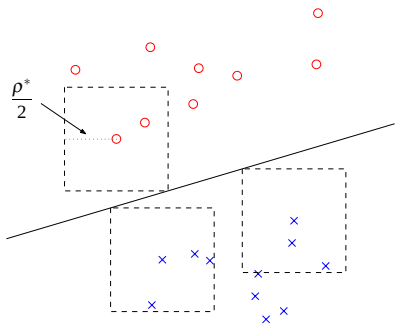


Figure 4.5 Optimal separating hyperplane using the ℓ_∞-norm

optimal margin ρ^* is twice the ℓ_1-norm distance of the support vectors to the optimal separating hyperplane. It is worth to note that, in this example, the same optimal separating hyperplane was found as a solution of both problems (4.18) and (4.25), although the margins using the ℓ_1-norm and the ℓ_2-norm are numerically different.

4.2.3 Optimal separation using the ℓ_∞-norm

From (4.11) and (4.15), the *Optimal Separation* problem using the ℓ_∞-norm becomes:

$$\begin{cases} \min\limits_{\mathbf{w},\gamma} & \|\mathbf{w}\|_1 \\ s.t. & z_k[\mathbf{w}'\mathbf{x}_k + \gamma] \geq 1 \quad k = 1,\dots,m \end{cases} \tag{4.26}$$

By recalling that $\|\mathbf{w}\|_1 = \sum\limits_{i=1}^{n} |w_i|$, problem (4.26) can be easily rewritten as a linear program by introducing the nonnegative auxiliary variables s_i, $i = 1,\dots,n$:

$$\begin{cases} \min\limits_{\mathbf{w},\gamma,s_i} & \sum\limits_{i=1}^{n} s_i \\ s.t. & z_k[\mathbf{w}'\mathbf{x}_k + \gamma] \geq 1 \quad k = 1,\dots,m \\ & -s_i \leq w_i \leq s_i \qquad i = 1,\dots,n \end{cases} \tag{4.27}$$

At optimality for problem (4.27), it holds $s_i = |\mathbf{w}_i|$, $i = 1,\dots,n$, so that $\sum\limits_{i=1}^{n} s_i = \|\mathbf{w}\|_1$. Figure 4.5 shows an example of optimal separation using the ℓ_∞-norm for the same

data set as in Figures 4.3 and 4.4. The optimal margin ρ^* is twice the ℓ_∞-norm distance of the support vectors to the optimal separating hyperplane.

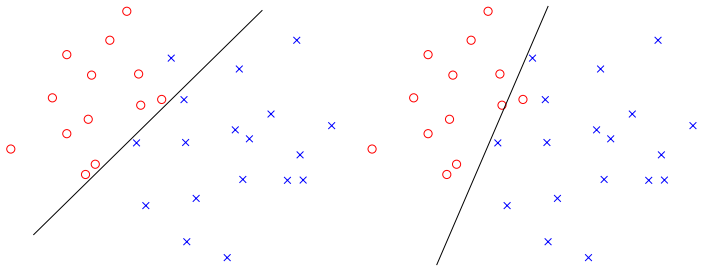
4.3 Two-class linear separation: the inseparable case

So far the discussion has been restricted to the case where the data set is linearly separable. It is however not always possible for a single linear function to separate without errors two given sets of points $\mathscr{A}_1$ and $\mathscr{A}_2$. This occurs when the sets $\mathscr{A}_1$ and $\mathscr{A}_2$ have intersecting convex hulls. Thus, it is important to find the linear function that discriminates best between the two sets according to some error minimization criterion.

When the two sets $\mathscr{A}_1$ and $\mathscr{A}_2$ are not linearly separable, a first reasonable approach is to look for a hyperplane that maximizes the number of well-separated points (equivalently, that minimizes the number of misclassified points). Such a hyperplane is called *generalized separating hyperplane* of the two sets. A Linear Programming formulation with Equilibrium Constraints (LPEC) for the Generalized Separating Hyperplane problem was proposed by Mangasarian (1994). An alternative Maximum Feasible Subsystem (MAX FS) formulation derives from the fact that computing a generalized separating hyperplane of two sets $\mathscr{A}_1$ and $\mathscr{A}_2$ amounts to find a pair $(\mathbf{w},\gamma)$, with $\mathbf{w} \in \mathbb{R}^n$ and $\gamma \in \mathbb{R}$, such that the number of satisfied inequalities in the following system:

$$\begin{cases} \mathbf{w}'\mathbf{x}_k + \gamma > 0 & \forall \mathbf{x}_k \in \mathscr{A}_1 \\ \mathbf{w}'\mathbf{x}_k + \gamma < 0 & \forall \mathbf{x}_k \in \mathscr{A}_2 \end{cases} \tag{4.28}$$

is maximized. Although the MAX FS problem is known to be NP-hard (Amaldi and Kann, 1995), and even difficult to approximate in polynomial time, several heuristics have been developed which work well in practice, *e.g*, the randomized and thermal relaxation algorithm proposed by Amaldi and Hauser (2001), and the branch-and-cut algorithm proposed by Pfetsch (2002). In Section 4.3.1, the former algorithm will be specified to the particular case (4.28), and a suitable initialization of the algorithm will be also proposed. Given a solution $(\mathbf{w},\gamma)$ of the MAX FS



(a) (b)

Figure 4.6 Two different solutions of the Generalized Separating Hyperplane problem for the same data set. One cross is misclassified in (a), whereas one circle is misclassifed in (b)

problem for system (4.28), the misclassified points, if any, are removed from $\mathscr{A}_1$ and/or $\mathscr{A}_2$. The remaining points hence form a linearly separable data set, which can be optimally separated, *e.g.*, by solving either problem (4.18), (4.25) or (4.27).

One drawback of the above approach is that the Generalized Separating Hyperplane problem might not have a unique solution (see Figure 4.6). An alternative approach for linear separation in the inseparable case considers optimization problems in which an additional cost function associated with misclassifications is minimized. In Section 4.3.2, problems (4.18), (4.25) and (4.27) will be extended so as to minimize the sum of misclassification errors along with maximizing the separation margin. The resulting linear and quadratic programming techniques are referred to as *Robust Linear Programming* (RLP) and *Support Vector Machines* (SVM).

4.3.1 Minimizing the number of misclassifications

Given a system of linear inequalities, the problem of finding a feasible subsystem with maximum cardinality is known as Maximum Feasible Subsystem (MAX FS) problem. As anticipated in Section 4.3, finding a generalized separating hyperplane of two sets $\mathscr{A}_1$ and $\mathscr{A}_2$ is equivalent to solve the MAX FS problem for system (4.28). In this section, the randomized and thermal relaxation algorithm for the MAX FS

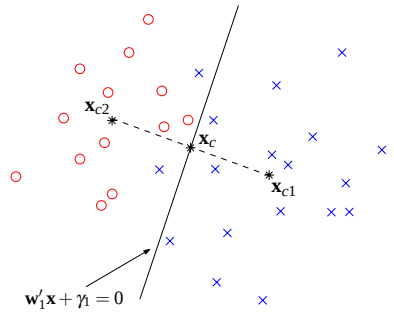


Figure 4.7 Initialization of the relaxation algorithm for finding a generalized separating hyperplane of two linearly inseparable sets

problem proposed in (Amaldi and Hauser, 2001) will be specified to the particular case of system (4.28). A simple initialization of the algorithm will be also proposed, which may help to obtain better final solutions.

In order to simplify the notation, by introducing the target values (4.5), system (4.28) can be rewritten in the compact matrix form:

$$A\mathbf{h} \succeq \varepsilon \mathbf{1} \tag{4.29}$$

where:

$$A = \begin{bmatrix} z_1\mathbf{x}_1' & z_1 \\ \vdots & \vdots \\ z_m\mathbf{x}_m' & z_m \end{bmatrix} \qquad \mathbf{h} = \begin{bmatrix} \mathbf{w} \\ \gamma \end{bmatrix}$$

and $\varepsilon > 0$ is introduced for guaranteeing a margin of separation nonzero. The algorithm is shown in Table 4.1. It is a randomized and thermal variant of the classical Agmon-Motzkin-Schoenberg relaxation method for solving systems of linear inequalities (Agmon, 1954; Motzkin and Schoenberg, 1954). It provides (suboptimal) solutions with a low computational burden, and is insensitive to numerical instabilities. First, it requires to define a maximum number of cycles $C > 0$, an initial temperature parameter $T_0 > 0$, and an initial solution $\mathbf{h}_0 \in \mathbb{R}^{n+1}$. For the choice of C and T_0, as well as for practical questions concerning the implementation of the algorithm, the reader is referred to (Amaldi and Hauser, 2001). As regards the

GIVEN: C, T_0, $\mathbf{h}_0$
Set $j = 0$ and $\bar{\mathbf{h}} = \mathbf{h}_0$
FOR $c = 0$ TO $C-1$ DO
  Compute (4.33) and set $\mathscr{I} = \{1,\dots,m\}$
  REPEAT
    Set $j = j+1$
    Pick an index k from $\mathscr{I}$ according to the prescribed rule
    Compute (4.30)
    IF $v_j^k > 0$ THEN
      Compute (4.32) and (4.31)
      IF $\mathbf{h}_j$ satisfies more inequalities in (4.29) than $\bar{\mathbf{h}}$ THEN set $\bar{\mathbf{h}} = \mathbf{h}_j$
    ELSE set $\mathbf{h}_j = \mathbf{h}_{j-1}$
    Set $\mathscr{I} = \mathscr{I} \setminus \{k\}$
  UNTIL $I = \emptyset$
END FOR
RETURN $\bar{\mathbf{h}}$

Table 4.1 The randomized and thermal relaxation algorithm used to solve the MAX FS problem for system (4.29)

initial solution $\mathbf{h}_0$, a simple but effective choice in this particular case is to compute the centers of the two sets, and then to consider the plane orthogonal to the segment drawn between the two centers, and passing through the middle point of it (see Figure 4.7). Defining the centers of the sets $\mathscr{A}_1$ and $\mathscr{A}_2$ as

$$\mathbf{x}_{c1} = \frac{\sum_{k:\mathbf{x}_k \in \mathscr{A}_1} \mathbf{x}_k}{\#\mathscr{A}_1} \quad \text{and} \quad \mathbf{x}_{c2} = \frac{\sum_{k:\mathbf{x}_k \in \mathscr{A}_2} \mathbf{x}_k}{\#\mathscr{A}_2},$$

respectively, and the mean point of $\mathbf{x}_{c1}$ and $\mathbf{x}_{c2}$ as $\mathbf{x}_c = (\mathbf{x}_{c1} + \mathbf{x}_{c2})/2$, the considered hyperplane is determined by the coefficient vector:

$$\mathbf{h}_0 \triangleq \begin{bmatrix} \mathbf{w}_0 \\ \gamma_0 \end{bmatrix} = \begin{bmatrix} \mathbf{x}_{c1} - \mathbf{x}_{c2} \\ (\mathbf{x}_{c2} - \mathbf{x}_{c1})' \mathbf{x}_c \end{bmatrix}$$

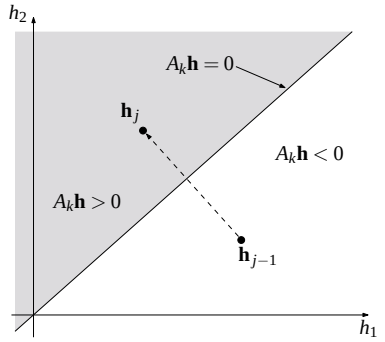


Figure 4.8 Geometric interpretation in the coefficient space of a single iteration of the relaxation method for the MAX FS problem (case $\mathbf{h} \in \mathbb{R}^2$)

The orientation of the normal vector $\mathbf{w}_0$ is chosen such that $\mathbf{x}_{c1}$ lies on the positive side of the hyperplane, and $\mathbf{x}_{c2}$ lies on the negative side. This choice provides in general better initial solutions (*i.e.*, satisfying a larger number of inequalities) than a random selection. Indeed, it is expected that many points of the two sets lie nearby the corresponding centers, and are therefore correctly classified by the proposed hyperplane. The algorithm then consists in a simple iterative procedure that generates a sequence of solutions. During each of the C outer cycles, all the rows of A are selected in the order defined by a prescribed rule (*e.g.*, cyclicly, or uniformly at random without replacement). Let the k-th row A_k of A be selected at iteration j, with $j = 1,\dots,Cm$, and $\mathbf{h}_{j-1}$ be the current solution. The corresponding violation is computed as:

$$v_j^k = \max\{0, \varepsilon - A_k \mathbf{h}_{j-1}\} \tag{4.30}$$

If v_j^k is zero, the current solution satisfies the k-th inequality in (4.29), and $\mathbf{h}_j$ is set equal to $\mathbf{h}_{j-1}$. If v_j^k is greater than zero, the current solution violates the k-th inequality in (4.29). Then, it is updated as follows:

$$\mathbf{h}_j = \mathbf{h}_{j-1} + \lambda_j A_k' \tag{4.31}$$

where $\lambda_j > 0$. Geometrically, the inequality $A_k \mathbf{h} \geq \varepsilon$ defines a half-space in the

coefficient space (the gray region in Figure 4.8). If the current solution belongs to the half-space, then it is left unchanged. Otherwise, $\mathbf{h}_j$ is obtained by making a step toward the half-space along the line orthogonal to the half-space and passing through $\mathbf{h}_{j-1}$ (see again Figure 4.8). The step size λ_j decreases exponentially with the violation:

$$\lambda_j = \frac{T}{T_0} \exp^{-\frac{v_j^k}{T}} \tag{4.32}$$

The basic idea of the algorithm is to favor updates of the current solution which aim at correcting unsatisfied inequalities with a relatively small violation. The correction of unsatisfied inequalities with large violations is likely to corrupt other inequalities that the current solution satisfies. Decreasing attention to unsatisfied inequalities with large violations is obtained by introducing a decreasing temperature parameter T, to which the violations are compared, *e.g.*:

$$T = \left(1 - \frac{c}{C}\right) T_0 \tag{4.33}$$

where c is the outer cycle counter. The solution returned by the algorithm is the coefficient vector $\bar{\mathbf{h}}$ that, during the process, has satisfied the largest number of inequalities in (4.29). The solution $\bar{\mathbf{h}}$ is not guaranteed to be optimal due to the randomness of the search, but extensive trials that are not presented in this thesis, have shown that the algorithm provides very good solutions in practice.

4.3.2 Minimizing the misclassification errors

Another approach for the inseparable case, that is alternative to minimizing the number of misclassifications, is the minimization of a suitable cost function associated with errors. In this section, the linear and quadratic programs formulated for the separable case, are extended with an additional error criterion to be minimized. In the simplest case, this idea leads to the following linear program:

$$\begin{cases} \min\limits_{\mathbf{w},\gamma,v_k} & \sum\limits_{k=1}^{m} c_k v_k & \\ s.t. & z_k[\mathbf{w}'\mathbf{x}_k + \gamma] \geq 1 - v_k & k = 1,\ldots,m \\ & v_k \geq 0 & k = 1,\ldots,m \end{cases} \tag{4.34}$$

where $c_k > 0$ are misclassification weights. If the data set is linearly separable, and therefore there exist $\mathbf{w} \in \mathbb{R}^n$ and $\gamma \in \mathbb{R}$ such that the constraints:

$$z_k[\mathbf{w}'\mathbf{x}_k + \gamma] \geq 1\,, \quad k = 1,\ldots,m \tag{4.35}$$

are satisfied, all the auxiliary variables v_k can be taken to be zero, and the optimal value for problem (4.34) is also zero. If the data set is not linearly separable, the auxiliary variables v_k allow the constraints (4.35) to be violated. Since, at optimality for problem (4.34), $v_k = \max\left\{0, 1 - z_k[\mathbf{w}'\mathbf{x}_k + \gamma]\right\}$, $k = 1,\ldots,m$, each variable v_k can be interpreted as a *misclassification error*. In the inseparable case, problem (4.34) hence looks for a hyperplane that minimizes a weighted sum of the misclassification errors. The original *Robust Linear Programming* (RLP) method proposed by Bennett and Mangasarian (1992) was a particular case of problem (4.34), where:

$$c_k = \begin{cases} \dfrac{1}{m_1} & \text{if } \mathbf{x}_k \in \mathscr{A}_1 \\ \dfrac{1}{m_2} & \text{if } \mathbf{x}_k \in \mathscr{A}_2 \end{cases} \tag{4.36}$$

Remark 4.3 Note that, at optimality for problem (4.34), only points $\mathbf{x}_k$ with $v_k \geq 1$ are truly misclassified according to (4.6). These points can be removed from the original data set. The remaining points then form a linearly separable data set, for which the optimal separating hyperplane can be computed. □

Although the low computational burden makes the RLP method appealing (its solution involves only a single linear program), problem (4.34) does not include any notion of margin maximization. Maximizing the margin is instead essential for good generalization (see Remark 4.2). In principle, as suggested in Remark 4.3, the misclassified points could be removed from the data set after solving problem (4.34), and the optimal separating hyperplane could be computed for the remaining linearly separable data set by solving either problem (4.18), (4.25) or (4.27). This approach requires however to solve two optimization problems. A single multi-objective problem, which aims to minimize the absolute sum of the misclassification errors as well as to maximize the separation margin, can be constructed by combining problem (4.34) with either problem (4.18), (4.25) or (4.27).

The *Soft Margin Optimal Separation* problem originally proposed by Cortes and Vapnik (1995), derives directly from the optimal separation problem using the ℓ_2-norm described in Section 4.2.1. It consists in solving the following quadratic program:

$$\begin{cases} \min\limits_{\mathbf{w},\gamma,v_k} & \frac{1}{2}\sum\limits_{i=1}^{n} w_i^2 + C\sum\limits_{k=1}^{m} v_k \\ s.t. & z_k[\mathbf{w}'\mathbf{x}_k+\gamma] \geq 1-v_k \quad k=1,\ldots,m \\ & v_k \geq 0 \qquad\qquad\qquad\quad k=1,\ldots,m \end{cases} \tag{4.37}$$

where $C>0$ is a given value. For sufficiently large C, the functional in (4.37) describes the problem of constructing a hyperplane which minimizes the sum of misclassification errors, and maximizes the margin for the correctly classified points. If the data set can be separated without errors, the constructed hyperplane will coincide with the optimal separating hyperplane using the ℓ_2-norm. It should be noted that the weight C introduces additional control on the classifier. Indeed, as C tends to infinity, the solution of problem (4.37) converges to one where the misclassification minimization term dominates, whereas as C tends to zero, the solution converges to one where the margin maximization term dominates. As it will be shown in Remark 4.4, the weight C can be also interpreted as an upper bound on the Lagrange multipliers for problem (4.37).

Remark 4.4 The dual problem associated with (4.37) is (Cortes and Vapnik, 1995):

$$\begin{cases} \min\limits_{\lambda} & \frac{1}{2}\sum\limits_{i=1}^{m}\sum\limits_{j=1}^{m} z_i z_j \mathbf{x}_i'\mathbf{x}_j\lambda_i\lambda_j - \sum\limits_{k=1}^{m}\lambda_k \\ s.t. & \sum\limits_{k=1}^{m} z_k\lambda_k = 0 \\ & 0 \leq \lambda_k \leq C \quad k=1,\ldots,m \end{cases} \tag{4.38}$$

where $\lambda = (\lambda_1,\ldots,\lambda_m)$ are the Lagrange multipliers associated with the constraints $z_k[\mathbf{w}'\mathbf{x}_k+\gamma] \geq 1-v_k$. Note that the Lagrange multipliers $\mu = (\mu_1,\ldots,\mu_m)$ associated with the constraints $v_k \geq 0$ do not appear in problem (4.38). However, they can be easily retrieved as:

$$\mu_k = C - \lambda_k, \quad k=1,\ldots,m \tag{4.39}$$

The dual problem (4.38) in the inseparable case is identical to the dual problem (4.20) in the separable case, except for introducing an upper bound on the Lagrange multipliers λ_k. Since, at optimality, complementary slackness implies that $\mu_k v_k = 0$, $k=1,\ldots,m$, it turns out that, if v_k is greater than zero, then μ_k is equal to zero. From (4.39), one can hence conclude that the misclassified points are those whose corresponding Lagrange multiplier λ_k saturates the upper bound. Note that, if C is chosen too small, then the Lagrange multipliers λ_k will all take on the value of C.

It is worth to note that the dual problem (4.38) is the basis for *Support Vector Machine* (SVM) classification. See, *e.g.*, (Vapnik, 1995; Vapnik, 1998; Cristianini and Shawe-Taylor, 2000) for an introduction to this topic. □

Linear programming versions of the Soft Margin Optimal Separation problem (4.37) can be defined by considering the minimization of either the ℓ_1-norm or the ℓ_∞-norm of the normal vector $\mathbf{w}$. The following linear program is derived from problem (4.25), where the ℓ_∞-norm of $\mathbf{w}$ is minimized:

$$\begin{cases} \min\limits_{\mathbf{w},\gamma,s,v_k} & s + C\sum\limits_{k=1}^{m} v_k \\ s.t. & z_k[\mathbf{w}'\mathbf{x}_k+\gamma] \geq 1-v_k \quad k=1,\ldots,m \\ & -s \leq w_i \leq s \qquad\quad i=1,\ldots,n \\ & v_k \geq 0 \qquad\qquad\quad k=1,\ldots,m \end{cases} \tag{4.40}$$

whereas the following linear program is derived from problem (4.27), where the ℓ_1-norm of $\mathbf{w}$ is minimized:

$$\begin{cases} \min\limits_{\mathbf{w},\gamma,s_i,v_k} & \sum\limits_{i=1}^{n} s_i + C\sum\limits_{k=1}^{m} v_k \\ s.t. & z_k[\mathbf{w}'\mathbf{x}_k+\gamma] \geq 1-v_k \quad k=1,\ldots,m \\ & -s_i \leq w_i \leq s_i \qquad\quad i=1,\ldots,n \\ & v_k \geq 0 \qquad\qquad\quad k=1,\ldots,m \end{cases} \tag{4.41}$$

Problems (4.40) and (4.41) are also referred to as *Robust Linear Programming* (RLP) methods. The advantage of RLP formulations over the SVM formulation is that the former can be solved by using linear programming instead of quadratic programming. It is known that a quadratic program can be much more computationally demanding than a linear program for the same problem size. As regards the

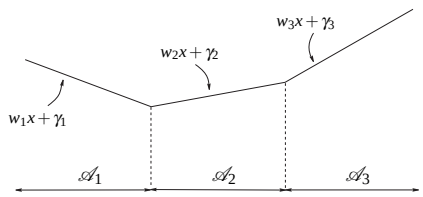


Figure 4.9 Piecewise linear separation of three sets $\mathscr{A}_1$, $\mathscr{A}_2$ and $\mathscr{A}_3$ in $\mathbb{R}$ by using the piecewise linear function $f(x) = \max_{i=1,2,3} w_i x + \gamma_i$

property of generalization, there are instead no results proving that one formulation is preferable to the others (see Remark 4.2).

Note that the minimization of the sum of the misclassification errors is a substitute to the problem of minimizing the sum of the distances of the misclassified points to the separating hyperplane. This problem is formulated in (Mangasarian, 1999) for arbitrary-norm distances of a point to a hyperplane. However, for the ℓ_1-norm only the problem is easily solvable by taking the best solution of $2n$ linear programs.

4.4 Multi-class linear separation

In *multi-category* classification, a piecewise linear function[2] is used to discriminate between $s > 2$ sets $\mathscr{A}_1, \ldots, \mathscr{A}_s$ of points in $\mathbb{R}^n$. A piecewise linear classifier f is constructed as the maximum of s linear classification functions $f_i(\mathbf{x}) = \mathbf{w}_i'\mathbf{x} + \gamma_i$, where $\mathbf{w}_i \in \mathbb{R}^n$ and $\gamma_i \in \mathbb{R}$ are, respectively, a weight vector and a bias associated with the i-th class, $i = 1, \ldots, s$. Hence:

$$f(\mathbf{x}) = \max_{i=1,\ldots,s} \mathbf{w}_i'\mathbf{x} + \gamma_i \tag{4.42}$$

[2] In order to be consistent with the terminology used in the literature of linear classification, in this section the term *linear* (or *piecewise linear*) function will be generally used to indicate also affine (respectively, piecewise-affine) functions.

(4.42) is a continuous and convex function that is defined for all $\mathbf{x} \in \mathbb{R}^n$ (see Figure 4.9). The corresponding decision rule is given by:

$$\mathscr{C}(\mathbf{x}) = \arg \max_{i=1,\ldots,s} \mathbf{w}_i'\mathbf{x} + \gamma_i \tag{4.43}$$

Remark 4.5 Definition (4.43) is not well-posed for those points $\mathbf{x} \in \mathbb{R}^n$ such that the optimizer of (4.43) is not unique. Such points lie on subspaces with zero measure, namely affine subspaces with dimension $n-1$ defined by equations of the type $\mathbf{w}_i'\mathbf{x} + \gamma_i = \mathbf{w}_j'\mathbf{x} + \gamma_j$ for $i \neq j$, and are therefore unlikely to occur in practice. This issue is however not of practical interest in the problem at hand. □

According to (4.43), if a point $\mathbf{x} \in \mathbb{R}^n$ is assigned to the i-th class, *i.e.*, $\mathscr{C}(\mathbf{x}) = i$ for some $i = 1, \ldots, s$, then it satisfies:

$$\mathbf{w}_i'\mathbf{x} + \gamma_i \geq \mathbf{w}_j'\mathbf{x} + \gamma_j, \quad j \neq i \tag{4.44}$$

By introducing the matrices $H_i \in \mathbb{R}^{(s-1)\times(n+1)}$, $i = 1, \ldots, s$, defined as follows:

$$H_i = \begin{bmatrix} \mathbf{w}_1 - \mathbf{w}_i & \ldots & \mathbf{w}_{i-1} - \mathbf{w}_i & \mathbf{w}_{i+1} - \mathbf{w}_i & \ldots & \mathbf{w}_s - \mathbf{w}_i \\ \gamma_1 - \gamma_i & \ldots & \gamma_{i-1} - \gamma_i & \gamma_{i+1} - \gamma_i & \ldots & \gamma_s - \gamma_i \end{bmatrix}' \tag{4.45}$$

(4.44) can be rewritten in the compact form $H_i\varphi \preceq \mathbf{0}$, where $\varphi = [\ \mathbf{x}'\ 1\ ]'$. The decision rule (4.43) hence corresponds to a partition of the n-dimensional space into s polyhedral regions (4.2), that is the partition of the piecewise linear map (4.42). A different class is associated to each region. As discussed in Remark 4.5, multiple classification may occur only for those points lying on the boundaries of the regions.

Let each set $\mathscr{A}_i$ with m_i points be represented by the matrix $A_i \in \mathbb{R}^{m_i \times n}$ whose rows are the points in $\mathscr{A}_i$, $i = 1, \ldots, s$. In order to separate the sets $\mathscr{A}_1, \ldots, \mathscr{A}_s$ correctly by using (4.43), according to (4.44) the following inequalities must be satisfied:

$$A_i\mathbf{w}_i + \gamma_i \mathbf{1} \succ A_i\mathbf{w}_j + \gamma_j \mathbf{1}, \quad i, j = 1, \ldots, s,\ j \neq i \tag{4.46}$$

Hence, the sets $\mathscr{A}_1, \ldots, \mathscr{A}_s$ are said to be *piecewise linearly separable* if there exist $\mathbf{w}_i \in \mathbb{R}^n$ and $\gamma_i \in \mathbb{R}$, $i = 1, \ldots, s$, such that (4.46) holds (see Figure 4.10). It turns

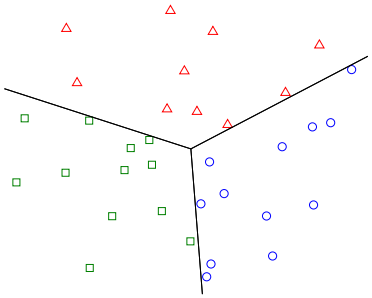


Figure 4.10 Three sets $\mathscr{A}_1$, $\mathscr{A}_2$ and $\mathscr{A}_3$ (*triangles*, *squares* and *circles*) that are piecewise linearly separable, and the partition of the space corresponding to a piecewise linear function (4.42) that obtains correct classification

out from this definition that, if the sets $\mathscr{A}_1,\ldots,\mathscr{A}_s$ are piecewise linearly separable, then each pair $(\mathscr{A}_i,\mathscr{A}_j)$, with $i\neq j$, is linearly separable. The converse is however not true, as shown by the "whirlwind" counterexample (Bennett and Mangasarian, 1994) in Figure 4.11.

Two different approaches for computing the pairs $(\mathbf{w}_i,\gamma_i)$, $i=1,\ldots,s$, defining the piecewise linear classifier (4.42), will be described in the following. The first approach (Vapnik, 1995; Bredensteiner and Bennett, 1999) consists in computing the pair $(\mathbf{w}_i,\gamma_i)$ associated with the i-th class by separating the set $\mathscr{A}_i$ from the union of the remaining $s-1$ sets, $i=1,\ldots,s$. For each set $\mathscr{A}_i$, this amounts to solve a two-class linear separation problem, for whose solution the MAX FS, SVM and RLP approaches described in Section 4.3, can be applied. When either SVM or RLP is used to solve each single two-class linear separation problem, this approach is sometimes referred to as s-SVM and s-RLP, respectively (where s is the number of classes). If each set $\mathscr{A}_i$ is linearly separable from the union of the remaining $s-1$ sets, all the pairs $(\mathbf{w}_i,\gamma_i)$, $i=1,\ldots,s$, can be taken such that:

$$\begin{cases} A_i\mathbf{w}_i+\gamma_i\mathbf{1}\succ\mathbf{0} \\ A_j\mathbf{w}_i+\gamma_i\mathbf{1}\prec\mathbf{0}, \quad j\neq i \end{cases}$$

(*i.e.*, each set $\mathscr{A}_i$ lies on the positive side of the hyperplane $\mathbf{w}_i'\mathbf{x}+\gamma_i=0$). Each point $\mathbf{x}\in\mathscr{A}_i$ is then classified correctly by using (4.43), since it satisfies $\mathbf{w}_i'\mathbf{x}+\gamma_i>0$ and

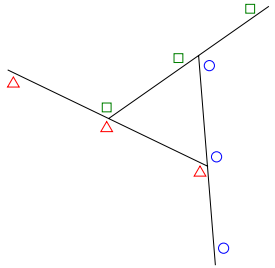


Figure 4.11 The whirlwind counterexample. Three sets (*triangles*, *squares* and *circles*) that are pairwise linearly separable, but not piecewise linearly separable

$\mathbf{w}_j'\mathbf{x}+\gamma_j<0$, $j\neq i$, and hence $\mathscr{C}(\mathbf{x})=i$. However, when at least one set $\mathscr{A}_i$ is not linearly separable from the union of the remaining $s-1$ sets, this approach may fail to classify correctly all the points, even if the data set is actually piecewise linearly separable (see Figure 4.12).

The former approach requires to solve s two-class linear separation problems. The second approach (Bennett and Mangasarian, 1994; Bredensteiner and Bennett, 1999) looks for the pairs $(\mathbf{w}_i,\gamma_i)$, $i=1,\ldots,s$, if any, satisfying (4.46), or the following equivalent normalized constraints:

$$A_i\mathbf{w}_i+\gamma_i\mathbf{1}\succeq A_i\mathbf{w}_j+\gamma_j\mathbf{1}+\mathbf{1}, \quad i,j=1,\ldots,s,\ j\neq i \tag{4.47}$$

by solving a single (computationally more demanding) optimization problem. The *Multicategory* RLP (M-RLP) method proposed by Bennett and Mangasarian (1994) consists in a linear program that generates a piecewise linear separation between the sets $\mathscr{A}_1,\ldots,\mathscr{A}_s$, if one exists; otherwise, a weighted sum of the misclassification errors is minimized. By introducing the auxiliary variables $\mathbf{v}_{ij}\in\mathbb{R}^{m_i}$, $i,j=1,\ldots,s$, $j\neq i$, in (4.47), the M-RLP linear program looks as follows:

$$\begin{cases} \min\limits_{\mathbf{w}_i,\gamma_i,\mathbf{v}_{ij}} & \displaystyle\sum_{i=1}^{s}\sum_{\substack{j=1\\ j\neq i}}^{s}\frac{\mathbf{1}'\mathbf{v}_{ij}}{m_i} & \\ s.t. & A_i(\mathbf{w}_i-\mathbf{w}_j)+(\gamma_i-\gamma_j)\mathbf{1}\succeq\mathbf{1}-\mathbf{v}_{ij} & i,j=1,\ldots,s,\ j\neq i \\ & \mathbf{v}_{ij}\succeq\mathbf{0} & i,j=1,\ldots,s,\ j\neq i \end{cases} \tag{4.48}$$

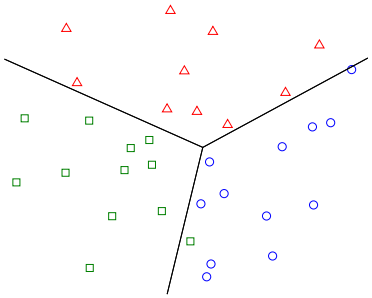


Figure 4.12 The *s*-SVM approach fails to separate correctly the same three piecewise linearly separable sets shown in Figure 4.10. It is pointed out that the partition of the space is not defined by the lines $\mathbf{w}_i'\mathbf{x}+\gamma_i=0$ computed by trying to linearly separate each set $\mathscr{A}_i$ from the union of the other two, $i=1,2,3$, rather by the lines $(\mathbf{w}_i-\mathbf{w}_j)'\mathbf{x}+(\gamma_i-\gamma_j)=0$, $i,j=1,2,3,\ j\neq i$.

It is easy to show that the sets $\mathscr{A}_1,\ldots,\mathscr{A}_s$ are piecewise linearly separable if and only if the optimum for problem (4.48) is zero, *i.e.*, there exists a feasible solution of (4.48) with all the auxiliary variables $\mathbf{v}_{ij}$ equal to zero. Since, at optimality for problem (4.48), it holds:

$$\mathbf{v}_{ij}=\max\left\{\mathbf{0},\mathbf{1}-A_i(\mathbf{w}_i-\mathbf{w}_j)-(\gamma_i-\gamma_j)\mathbf{1}\right\},\quad i,j=1,\ldots,s,\ j\neq i$$

the variables $\mathbf{v}_{ij}$ can be interpreted as *misclassification errors*. The averaged sum of the misclassification errors is hence minimized by solving problem (4.48) in the piecewise linearly inseparable case. If $s=2$, problem (4.48) is equivalent to problem (4.34), with the weights c_k chosen as in (4.36). M-RLP is therefore an extension of two-class RLP to the multi-class case. Figure 4.10 shows an example of three sets that were piecewise linearly separated by solving problem (4.48). Like the original two-class RLP problem (4.34), the M-RLP problem (4.48) does not include any term for maximizing the margins of separation between pairs of the sets $\mathscr{A}_1,\ldots,\mathscr{A}_s$. To this aim, in (Bredensteiner and Bennett, 1999) M-RLP and SVM were combined so as to include margin maximization. The resulting method is called *Multicategory* SVM (M-SVM). In the piecewise linearly separable case, a quadratic program is constructed by starting from observing that, if the piecewise

linear function (4.42) classifies correctly the sets $\mathscr{A}_1,\ldots,\mathscr{A}_s$, then the hyperplane defined by the equation $(\mathbf{w}_i-\mathbf{w}_j)'\mathbf{x}+(\gamma_i-\gamma_j)=0$ is a separating hyperplane for the pair $(\mathscr{A}_i,\mathscr{A}_j)$, with $i\neq j$. As discussed in Section 4.2.1, maximizing the margin between the sets $\mathscr{A}_i$ and $\mathscr{A}_j$ using the ℓ_2-norm is equivalent to minimize $\left\|\mathbf{w}_i-\mathbf{w}_j\right\|_2$. This leads to the following M-SVM quadratic program[3] in the piecewise linearly separable case:

$$\begin{cases}\min\limits_{\mathbf{w}_i,\gamma_i} & \frac{1}{2}\sum\limits_{i=1}^{s}\sum\limits_{j=1}^{i-1}\left\|\mathbf{w}_i-\mathbf{w}_j\right\|_2^2+\frac{1}{2}\sum\limits_{i=1}^{s}\|\mathbf{w}_i\|_2^2\\ s.t. & A_i(\mathbf{w}_i-\mathbf{w}_j)+(\gamma_i-\gamma_j)\mathbf{1}\geq\mathbf{1}\quad i,j=1,\ldots,s,\ j\neq i\end{cases}\tag{4.49}$$

where the additional term $\frac{1}{2}\sum\limits_{i=1}^{s}\|\mathbf{w}_i\|_2^2$ is also introduced in the objective. In the piecewise linearly inseparable case, the general M-SVM method is easily obtained by combining problems (4.48) and (4.49) as follows (Bredensteiner and Bennett, 1999):

$$\begin{cases}\min\limits_{\mathbf{w}_i,\gamma_i,\mathbf{v}_{ij}} & \frac{1}{2}\sum\limits_{i=1}^{s}\sum\limits_{j=1}^{i-1}\left\|\mathbf{w}_i-\mathbf{w}_j\right\|_2^2+\frac{1}{2}\sum\limits_{i=1}^{s}\|\mathbf{w}_i\|_2^2+C\sum\limits_{i=1}^{s}\sum\limits_{\substack{j=1\\ j\neq i}}^{s}\mathbf{1}'\mathbf{v}_{ij}\\ s.t. & A_i(\mathbf{w}_i-\mathbf{w}_j)+(\gamma_i-\gamma_j)\mathbf{1}\geq\mathbf{1}-\mathbf{v}_{ij}\quad i,j=1,\ldots,s,\ j\neq i\\ & \mathbf{v}_{ij}\succeq\mathbf{0}\qquad\qquad\qquad\qquad\qquad i,j=1,\ldots,s,\ j\neq i\end{cases}\tag{4.50}$$

where $C>0$ is a given value. As for problem (4.37), the weight C introduces additional control on the classifier. If C is large, the misclassification minimization term dominates, whereas for small C the margin maximization term dominates.

4.5 Estimation of the regions in PWA identification

In this section, it is shown how both the two-class and the multi-class linear separation techniques described in this chapter, can be used to carry out the estimation of the regions of a PWA model, once the data points have been classified and associated to submodels. This is the last step of the PWA system identification procedure proposed in this thesis and of other procedures (*e.g.*, Ferrari-Trecate *et al.*, 2003; Vidal

[3] The same reasoning can be repeated when considering margin maximization in terms of either the ℓ_1-norm or the ℓ_∞-norm. The resulting linear programs are not presented here.

et al., 2003*b*; Ragot *et al.*, 2003) tackling the identification problem by first classifying the data points and estimating the parameters of the affine submodels, and then reconstructing the regions of the identified PWA model.

The region estimation problem was introduced in Section 4.1. It consists in finding a polyhedral partition $\{\mathscr{X}_i\}_{i=1}^{s}$ of the regressor set $\mathscr{X}$ such that $\mathscr{F}_i \subseteq \mathscr{X}_i$ for all $i = 1,\dots,s$, where $\mathscr{F}_i$ are the clusters of regression vectors (4.1) and $\mathscr{X}_i$ are the regions (4.2). Otherwise, the number of misclassified points should be minimized. This is equivalent to linearly separating the s sets $\mathscr{F}_1,\dots,\mathscr{F}_s$, as it was discussed in Sections 4.1– 4.4.

When two-class linear separation techniques are used, the clusters $\mathscr{F}_i$ are considered pairwise, and a separating hyperplane for each pair $(\mathscr{F}_i,\mathscr{F}_j)$, with $i \neq j$, is computed. The misclassified points, if any, are then removed from $\mathscr{F}_i$ and/or $\mathscr{F}_j$. This procedure amounts to solve $s(s-1)/2$ two-class linear separation problems of the types described in Sections 4.2 and 4.3. The output are the pairs $(\mathbf{w}_{ij},\gamma_{ij})$, with $\mathbf{w}_{ij} \in \mathbb{R}^n$ and $\gamma_{ij} \in \mathbb{R}$, $i = 1,\dots,s-1$, $j = i+1,\dots,s$, such that the discrimination between the i-th and the j-th class, with $i < j$, is performed according to the following rule:

$$\begin{aligned} \mathbf{w}_{ij}'\mathbf{x}+\gamma_{ij} > 0 \quad &\Rightarrow \quad \mathbf{x} \text{ is assigned to the } i\text{-th class} \\ \mathbf{w}_{ij}'\mathbf{x}+\gamma_{ij} < 0 \quad &\Rightarrow \quad \mathbf{x} \text{ is assigned to the } j\text{-th class} \end{aligned}$$

Each polyhedral region $\mathscr{X}_i$, $i = 1,\dots,s$, is hence defined by the inequalities:

$$\begin{cases} \mathbf{w}_{ji}'\mathbf{x}+\gamma_{ji} \leq 0, & j = 1,\dots,i-1 \\ \mathbf{w}_{ij}'\mathbf{x}+\gamma_{ij} \geq 0, & j = i+1,\dots,s \end{cases}$$

and characterized in the compact form (4.2) by introducing the matrix:

$$H_i = \begin{bmatrix} \mathbf{w}_{1i} & \cdots & \mathbf{w}_{i-1,i} & -\mathbf{w}_{i,i+1} & \cdots & -\mathbf{w}_{is} \\ \gamma_{1i} & \cdots & \gamma_{i-1,i} & -\gamma_{i,i+1} & \cdots & -\gamma_{is} \end{bmatrix}' \in \mathbb{R}^{(s-1)\times(n+1)}$$

The approach based on pairwise linear separation of the clusters $\mathscr{F}_1,\dots,\mathscr{F}_s$ is computationally advantageous, because it does not involve all the data at the same time, and requires to solve either simple linear/quadratic programs (RLP/SVM), or MAX FS problems for whose solution efficient heuristics exist. On the other hand, a

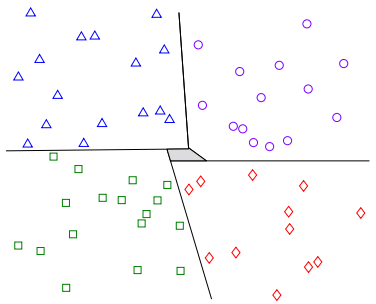


Figure 4.13 The partition of the regressor set constructed by pairwise linearly separating four clusters of regression vectors, is not complete (the gray area is not covered)

major drawback is that, when $n > 1$, the regions $\mathscr{X}_i$ are not guaranteed to form a complete partition of the regressor set, *i.e.*, there might exist $\mathbf{x} \in \mathscr{X}$ such that $\mathbf{x} \notin \mathscr{X}_i$ for any $i = 1,\dots,s$. Figure 4.13 shows an example in which a "hole" in the partition of the regressor set occurs. This drawback is quite important, since it causes the model to be not completely defined over the whole regressor set. When the PWARX model (3.2)–(3.4) is used in simulation, unclassified regression vectors $\mathbf{x}$ can be reasonably assigned to the nearest region. This corresponds to apply the following decision rule:

$$\mathscr{C}(\mathbf{x}) = \arg \min_{i=1,\dots,s} \min_{\mathbf{z}\in\mathscr{X}_i} \|\mathbf{z}-\mathbf{x}\| \tag{4.51}$$

When the PWARX model (3.2)–(3.4) is used in optimal control, trajectories passing through a "hole" are simply automatically discarded as infeasible, with the only consequent drawback of inducing suboptimal solutions.

In order to avoid the "holes" in the partition of the regressor set, the multi-class linear separation approach described in Section 4.4 can be used, since it classifies new points by using the piecewise linear function (4.42), that is defined over the whole n-dimensional space. Different methods for computing the parameters of the piecewise linear classifier have been also described in Section 4.4. They return the pairs $(\mathbf{w}_i,\gamma_i)$, with $\mathbf{w}_i \in \mathbb{R}^n$ and $\gamma_i \in \mathbb{R}$, $i = 1,\dots,s$, defining the linear classification functions associated with each class. In this case, the matrices $H_i \in \mathbb{R}^{(s-1)\times(n+1)}$

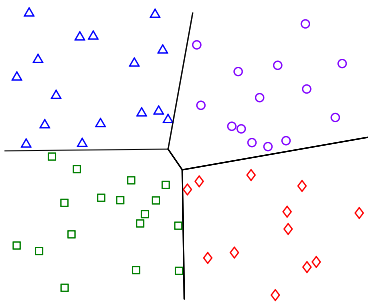


Figure 4.14 Partition of the regressor set constructed by piecewise linearly separating the same four clusters of regression vectors as in Figure 4.13. The separating hyperplane between the triangles and the diamonds is not drawn because it does not contribute to delimiting any region

characterizing the regions $\mathscr{X}_i$, $i = 1,\dots,s$, can be constructed as in (4.45). Figure 4.14 shows an example in which the same four clusters as in Figure 4.13 are correctly piecewise linearly separated. Note the absence of the "hole" in the partition, as compared to Figure 4.13. Multi-class linear separation problems involve all the available data at the same time, and are therefore computationally more demanding than pairwise linear separation problems.

A drawback that is common to both the pairwise two-class and the multi-class linear separation approach is that, if two clusters $\mathscr{F}_i$ and $\mathscr{F}_j$ are not contiguous, the corresponding separating hyperplane could be redundant, *i.e.*, it could not contribute to delimiting the region $\mathscr{X}_i$ and/or the region $\mathscr{X}_j$ (see again Figure 4.14). This drawback can be however easily overcome by eliminating redundant hyperplanes through standard linear programming techniques.

Even if the true system is piecewise-affine, and the number of submodels is correctly estimated, there might occur misclassifications while linearly separating the clusters $\mathscr{F}_i$, $i = 1,\dots,s$. Misclassifications are due to errors in clustering the data points during the refinement procedure. These errors are actually expected to be reduced by the distinction into *infeasible*, *undecidable*, and *feasible* data points (see Section 3.3). The infeasible data points should mainly account for outliers, whereas

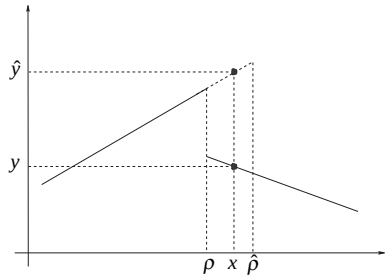


Figure 4.15 Discontinuous PWA system defined by two subsystems (solid lines) and one switching point ρ. It is assumed that the true subsystems are known, whereas $\hat{\rho}$ denotes the estimated switching point. y and $\hat{y}$ are the measured and the predicted output corresponding to a misclassified point x. The residual $y-\hat{y}$ is large, due to the discontinuity of the system

the undecidable data points are those that most likely could determine misclassifications, since they are consistent with more than one submodel. Once the regions $\mathscr{X}_i$, $i = 1,\dots,s$, have been estimated, all the data points can be finally classified by exploiting both the partition and the bounded-error condition. For $k = 1,\dots,N$, if $\mathbf{x}_k \in \mathscr{X}_i$ for some $i = 1,\dots,s$, and $|y_k - \varphi_k'\theta_i| \leq \delta$, then $(y_k,\mathbf{x}_k)$ is assigned to the cluster $\mathscr{D}_i$, otherwise it is marked as *infeasible*. A feasible parameter set:

$$FPS_i = \left\{\theta \in \mathbb{R}^{n+1} \,\middle|\, |y_k - \varphi_k'\theta| \leq \delta\,, \forall (y_k,\mathbf{x}_k) \in \mathscr{D}_i\right\}$$

can be also associated with the i-th submodel, $i = 1,\dots,s$, thus allowing for evaluation of the related parametric uncertainty (see Section 1.3). A second possibility is to assign all the data points $(y_k,\mathbf{x}_k)$ such that $\mathbf{x}_k \in \mathscr{X}_i$ to the cluster $\mathscr{D}_i$, $i = 1,\dots,s$, and then to exploit standard linear identification techniques (*e.g.*, Ljung, 1999) for identifying a final model in each region.

The quality of different PWARX models obtained by partitioning the regressor set using different linear separation techniques, should be compared by validating the models as described in Section 1.4. Note that, if the true system is characterized by a continuous dynamics, small differences in hyperplane orientations are not expected to alter considerably the quality of the model. On the other hand, even small errors in shaping the surfaces along which the true system is discontinuous,

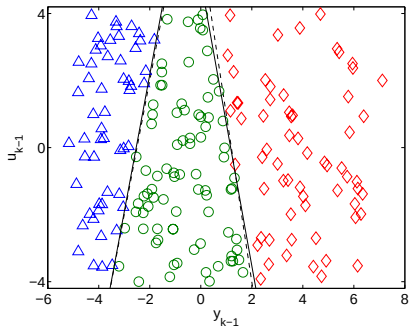


Figure 4.16 Final classification of the regression vectors (*triangles*, *circles*, *diamonds*), and true (*dashed lines*) and estimated (*solid lines*) partition of the regressor set

might determine large residuals, if a regression vector lying close to a discontinuity, is associated to the wrong submodel. This situation is illustrated in Figure 4.15. Hence, it would be desirable to have many regression vectors concentrated along the discontinuities, in order to shape them as good as possible.

If the number of misclassified points when linearly separating two clusters $\mathscr{F}_i$ and $\mathscr{F}_j$, is large, it likely means that at least one of the two clusters corresponds to either a nonconvex region (which then needs to be split into convex polyhedra), or nonconnected regions where the submodel is the same. Recall that the classification procedure groups together all the data points that are fitted by the same affine submodel. Efficient techniques for detecting and splitting the clusters corresponding to such situations, are currently under investigation.

Example 4.1 The final classification of the regression vectors and the estimated partition of the regressor set for Example 3.1, are plotted in Figure 4.16. The partition was estimated by using support vector machines. The line separating the triangles and the diamonds is not drawn, since it is redundant, whereas the two solid lines are defined by the coefficients ($\mathbf{w}_i \in \mathbb{R}^2$, $\gamma_i \in \mathbb{R}$, $i = 1,2$):

$$(\mathbf{w}_1, \gamma_1) = \left(3.9591, -0.9665, 10.0196\right), \quad (\mathbf{w}_2, \gamma_2) = \left(5.0513, 1.1876, -5.9223\right)$$

that are very similar to the true ones.

5

Applications

In this chapter, the performance of the proposed PWA system identification procedure will be tested on some numerical examples. A case study will be also presented, where the procedure was applied to the identification of a real process, namely the electronic component placement process in a pick-and-place machine. Effective use of the parameter δ of the procedure as a tuning knob to trade off between model complexity and quality of fit, will be demonstrated.

5.1 Numerical examples

Numerical examples, in which the PWA system identification procedure was applied to piecewise affine regression problems of different type, are illustrated in this section. Hints for the selection of the parameters of the procedure, and for improving the identified model in the validation phase, are also provided.

Example 5.1 The proposed estimation procedure was tested on the reconstruction of the following PWA map (Ferrari-Trecate and Muselli, 2003):

$$f(x_1,x_2) = \begin{cases} 4x_1+2x_2+3 & \text{if} \quad 0.5x_1+0.29x_2 \geq 0 \quad \text{and} \quad x_2 \geq 0 \\ -6x_1+6x_2-5 & \text{if} \quad 0.5x_1+0.29x_2 < 0 \quad \text{and} \quad 0.5x_1-0.29x_2 < 0 \\ 4x_1-2x_2-2 & \text{if} \quad 0.5x_1-0.29x_2 \geq 0 \quad \text{and} \quad x_2 < 0 \end{cases}$$

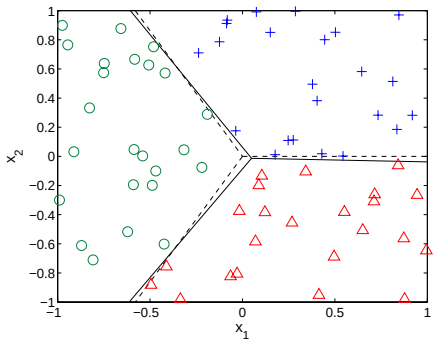


Figure 5.1 Example 5.1: The points (x_1, x_2) available for regression, and the true (dashed lines) and the estimated (solid lines) partition of the domain $\mathscr{X}$

over the domain $\mathscr{X} = [-1,1] \times [-1,1]$. The plot of the function f is shown in Figure 2.1. A data set containing $N = 70$ samples (y, x_1, x_2) was generated according to the model $y = f(x_1, x_2) + e$, where the noise e was a normal random variable with zero mean and variance $\sigma^2 = 0.01$. The points (x_1, x_2) available for regression are shown in Figure 5.1, together with the true partition of the domain $\mathscr{X}$ (dashed lines). This benchmark problem was not challenging for data classification and parameter estimation, thanks to the very high Signal-to-Noise Ratio (SNR). However, it will be useful to highlight some of the problems related to the reconstruction of the regions from a finite number of data. The bound δ of the procedure was chosen equal to 0.4. Hence, $\delta = 4\sigma$. The initialization of the procedure provided a correct estimate of the true number of submodels, *i.e*, $s = 3$. All the data points were correctly classified after the refinement, and the estimated parameter vectors are shown in Table 5.1. They are very similar to the true ones. In order to avoid "holes" in the partition, the regions were reconstructed by using Multicategory RLP. The estimated coefficients of the guardlines are shown in Table 5.2, and the corresponding partition of the domain $\mathscr{X}$ is drawn in Figure 5.1 (solid lines). Differences between the true and the estimated partition are most evident around the intersection of the three guardlines. Errors in the estimation of the guardlines from a finite data set are in general inevitable. However, they might be detected a posteriori during validation of the model. Figure 5.2 shows the plot of the approximation error for a different data set from that used for estimation. Since the PWA map is discontinuous, two misclassified data points (*i.e.*, data points that are associated to the wrong

Table 5.1 Example 5.1: True ($\bar{\theta}_i$) and estimated (θ_i) parameter vectors for each submodel

$\bar{\theta}_1$	θ_1	$\bar{\theta}_2$	θ_2	$\bar{\theta}_3$	θ_3
4	4.0197	-6	-5.9545	4	4.0571
2	1.9836	6	5.9508	-2	-2.1706
3	3.0119	-5	-4.9937	-2	-2.1393

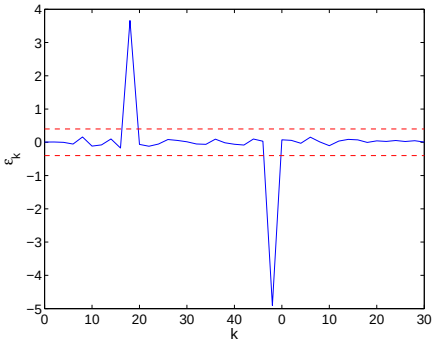


Figure 5.2 Example 5.1: Plot of the approximation error using validation data points. The horizontal lines bound the interval $[-\delta, \delta]$

Table 5.2 Example 5.1: True ($\bar{\mathbf{h}}_i$) and estimated ($\mathbf{h}_i$) coefficients of the guardlines

$\bar{\mathbf{h}}_1$	$\mathbf{h}_1$	$\bar{\mathbf{h}}_2$	$\mathbf{h}_2$	$\bar{\mathbf{h}}_3$	$\mathbf{h}_3$
5	4.8539	5	4.8005	0	0.0259
2.9	3.1384	-2.9	-3.2196	1	0.9997
0	-0.1962	0	-0.2769	0	0.0118

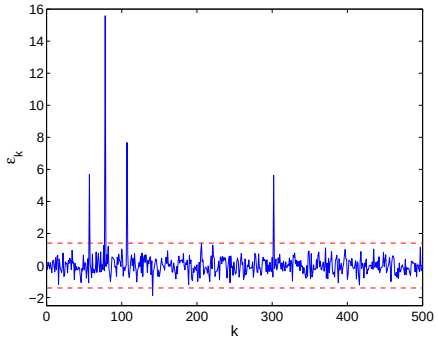


Figure 5.3 Example 5.2: Plot of the residuals using validation data. The horizontal lines bound the interval $[-\delta, \delta]$. Spikes are due to regression vectors incorrectly classified, and to discontinuity of the PWA map

submodel due to the errors in estimating the regions) determine distinct spikes in the plot of the approximation error. These data points can be re-attributed, *e.g.*, to the nearest region with a compatible submodel. Then, the augmented data set is used to re-estimate the regions.

Example 5.2 The PWA system identification procedure was successfully applied to fit the data generated by a discontinuous PWARX system with orders $n_a = 2$ and $n_b = 2$, and $\bar{s} = 4$ regions. The input signal was generated according to a uniform distribution on $[-5, 5]$, and the noise signal was drawn from a normal distribution with zero mean and variance $\sigma^2 = 0.2$. The estimation data set contained $N = 1000$ data points, of which 292, 234, 361 and 113 were generated by each of the four submodels, respectively. The SNR was about 17. The bound δ was chosen equal to 1.14, approximately 3.13σ. The initialization of the procedure provided the correct number $s = 4$ of submodels, and clusters containing 363, 287, 229 and 121 data points, respectively. The refinement procedure was run with $c = 10$, and terminated after 5 iterations. The estimated parameter vectors after the refinement are shown in Table 5.3. They are very good estimates of the true ones. At this stage, the classification of the data points consisted of clusters with 293, 207, 365 and 101

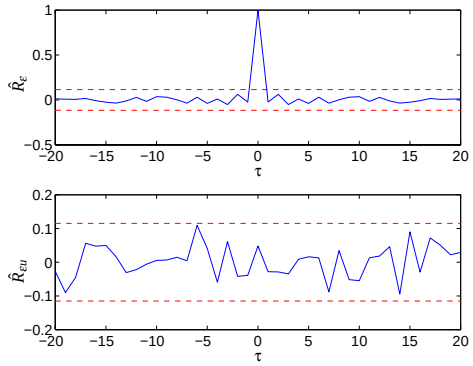


Figure 5.4 Example 5.2: Results from the residual analysis of the identified PWARX model using validation data

data points, respectively. One data point was infeasible, and only 33 data points out of ≈ 350 had been left undecidable. Then, the regions were computed by using Multicategory RLP. The final reassignment of the data points provided clusters with 291, 235, 360 and 113 data points. Only one data point was left infeasible. The 99.7% of the data points were correctly classified.

The model was validated by computing the residuals using $N_V = 500$ validation data. The plot of the residuals is shown in Figure 5.3. They are mostly contained in the interval $[-\delta, \delta]$. Recall that the noise follows a normal distribu-

Table 5.3 Example 5.2: True ($\bar{\theta}_i$) and estimated (θ_i) parameter vectors for each submodel

$\bar{\theta}_1$	θ_1	$\bar{\theta}_2$	θ_2	$\bar{\theta}_3$	θ_3	$\bar{\theta}_4$	θ_4
-0.05	-0.0593	1.21	1.2208	1.49	1.4939	-1.20	-1.1838
0.76	0.7818	-0.49	-0.4957	-0.50	-0.4995	-0.72	-0.7275
1.00	1.0081	-0.30	-0.3007	0.20	0.2115	0.60	0.5716
0.50	0.5054	0.90	0.9035	-0.45	-0.4481	-0.70	-0.7013
-0.50	-0.4782	0	0.0242	1.70	1.7451	2.00	1.8076

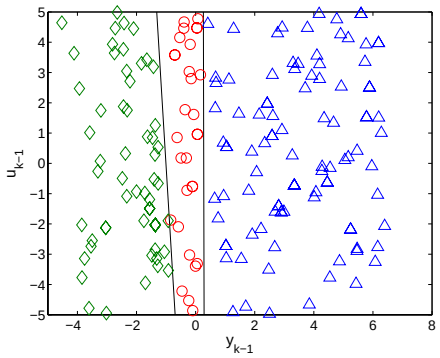


Figure 5.5 Example 5.3: Partition of the regressor set (solid lines) and classification (*diamonds*, *circles* and *triangles*) of some of the regression vectors used for estimation

tion, and that δ was taken $\approx 3\sigma$. Spikes are due to discontinuity of the PWA map, and to regression vectors incorrectly classified because of errors in estimating the regions. For them, the wrong parameter vector was used to compute the prediction. Nevertheless, the identified PWARX model was not falsified by the whiteness test of the residuals or by the cross-correlation test between the residuals and the input, as it can be seen in Figure 5.4. Note that this example was quite challenging, due to the high number of parameters to be estimated with respect to the available data, and the high number ($\approx 35\%$) of undecidable data points.

Example 5.3 The PWA system identification procedure was applied to fit a PWARX model with $n_a = 1$ and $n_b = 1$ to the data generated by the following nonlinear system (Bemporad *et al.*, 2003*b*):

$$y_k = \sqrt{|y_{k-1}|} - u_{k-1} + e_k \tag{5.1}$$

The input signal u_k was generated according to a uniform distribution on $[-5,5]$, and the noise signal e_k was drawn from a normal distribution with zero-mean and variance $\sigma^2 = 0.05$. An estimation data set composed by $N = 1000$ data points was considered. The SNR was about 12.7. The bound δ was chosen equal to 0.33, approximately 1.47σ. The initialization of the procedure provided 6 submodels.

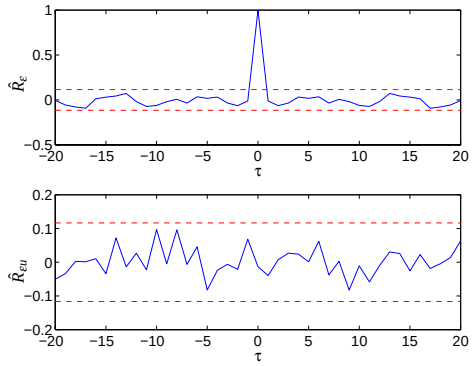


Figure 5.6 Example 5.3: Results from the residual analysis of the identified PWARX model using validation data

Then, the refinement procedure was run with $\alpha = 10\%$, $\beta = 5\%$, and $c = 10$, and terminated after 25 iterations. The number of submodels was reduced to 3. The regions were finally computed by using SVM. The final classification of the regression vectors, and the partition of the regressor set is shown in Figure 5.5. The parameter vectors and the coefficients of the guardlines that were returned by the identification procedure, are reported in Table 5.4. The reconstructed PWA map reproduces almost correctly the symmetry of the nonlinear function in (5.1).

The model was validated by computing the residuals using $N_V = 500$ validation data. The average quadratic error S_2^2 was equal to 0.057, which is very close to the variance of the noise. Recall that the prediction error is influenced by both

Table 5.4 Example 5.3: Parameter vectors θ_i and coefficients $\mathbf{h}_i$ of the guardlines

θ_1	θ_2	θ_3	$\mathbf{h}_1$	$\mathbf{h}_2$
-0.3057	-0.6512	0.2833	0.9981	1.000
-0.9981	-1.0151	-1.0016	0.0621	0.0024
0.7889	0.3378	0.8117	1.0026	-0.2741

the noise and the model error. Figure 5.6 shows the whiteness test of the residuals, and the cross-correlation test between the residuals and the input. Since both plots are inside the bounds, the identified PWARX model is not falsified by this model validation test. The fit between the measured and the simulated response was also considered. The value of the percentage of variance-accounted-for (*VAF*) was 91.04%.

5.2 A case study

The PWA system identification procedure described in this thesis, was successfully applied to the identification of an electronic component placement process in a pick-and-place machine[1]. Pick-and-place machines are used to automatically place electronic components on printed circuit boards. The process consists of a mounting head carrying the electronic component. The component is pushed down until it comes in contact with the circuit board, and then is released. Models of this process are of great importance for control design, since the whole operation should be as fast as possible (to achieve maximal throughput), while satisfying technological and safety constraints (*e.g.*, the exerted forces must not damage the component). Input-output data were gathered from a real experimental setup consisting of a mounting head and an impacting surface simulating the printed circuit board. A physical model of the experimental setup is shown in Figure 5.7. The mounting head is represented by the mass M, whose movement is only enabled along the vertical axis. The springs c_1 and c_2 simulate elasticity. The dampers d_1 and d_2 provide linear friction, whereas the blocks f_1 and f_2 provide dry friction. The input and the output of the system are the voltage applied to the motor driving the mounting head (represented by the exerted force F in Figure 5.7), and the position of the mounting head, respectively. Four operating conditions of the system can be distinguished. In the

[1]The author would like to thank Aleksandar Juloski (Department of Electrical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands) for providing the data used for identification in this section. Data are by courtesy of Assembleon, Eindhoven (`www.assembleon.com`).

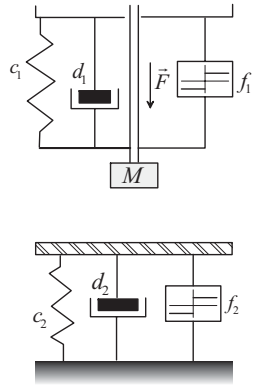


Figure 5.7 Physical model of the mounting head

free mode, the mounting head moves unconstrained, *i.e.*, without being in contact with the impacting surface. In the *impact* mode, the mounting head moves in contact with the impacting surface. The *upper* and *lower saturation* modes correspond to situations in which the mounting head cannot move upwards or downwards, respectively, due to physical constraints. The reader is referred to (Juloski *et al.*, 2003) for a more detailed description of the experimental setup.

The considered data set was such that only two operating conditions were excited, namely the free mode and the impact mode. Input-output data used for identification and validation are plotted in Figure 5.8. Nonlinear phenomena due to

Table 5.5 Identification of the mounting head: Fit between the measured and the simulated response using the identified PWARX models with $s = 2$, $s = 3$ and $s = 4$ discrete modes

	VAF
$s = 2$	81.33%
$s = 3$	90.18%
$s = 4$	93.48%

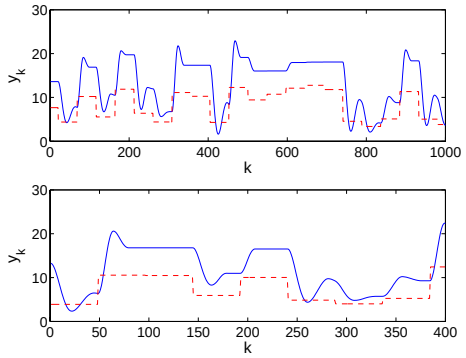
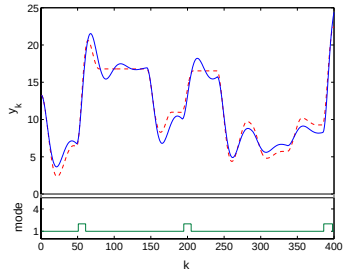
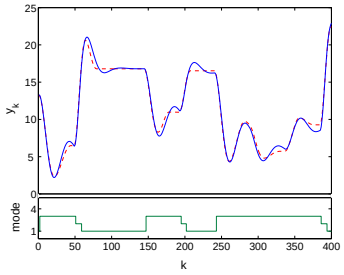


Figure 5.8 Identification of the mounting head: Data sets used for estimation (upper plot) and validation (lower plot). The solid line is the system output, and the dashed line is the scaled input

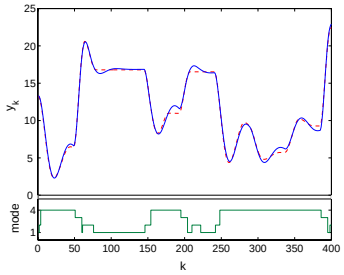
dry friction damping are evident in both data sets, *e.g.*, in the upper plot of Figure 5.8 on the interval $(500, 750)$. A PWARX model structure with orders $n_a = 2$ and $n_b = 2$ was considered. By choosing $\delta = 0.06$, $\delta = 0.05$, and $\delta = 0.04$, models with $s = 2$, $s = 3$, and $s = 4$ discrete modes, respectively, were identified. $N = 1000$ estimation data were used. For $s = 3$ and $s = 4$, M-RLP linear separation techniques (see Section 4.4) were applied in the region estimation step so as to avoid "holes" in the partition. Validation was then carried out by computing the fit between the measured and the simulated response using $N_v = 400$ validation data. The values (1.27) of the percentage of variance accounted for (*VAF*), are shown in Table 5.5 for the three identified models. These values demonstrate that the fit improves as the number of submodels increases, *i.e*, as smaller and smaller values of δ are chosen in the PWA system identification procedure. In Figure 5.9, the plots of the simulated responses are graphically compared to the measured response. Figure 5.9(a) clearly shows that only two affine submodels are not sufficient for accurately reproducing the system dynamics. Very good accordance between the measured and the simulated response is instead obtained with $s = 3$ and $s = 4$ submodels. Difficulties of the identified models in reproducing the nonlinear phenomena on the interval $(210, 240)$ are likely



(a)



(b)



(c)

Figure 5.9 Identification of the mounting head: Simulation results of the identified PWARX models with (a) $s = 2$, (b) $s = 3$, and (c) $s = 4$ submodels (solid line - simulated output, dashed line - system output). The lower plot in each figure shows the evolution of the discrete mode

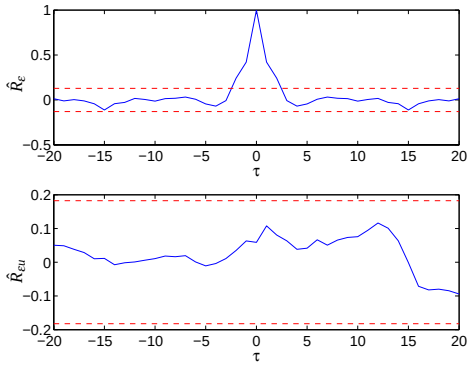


Figure 5.10 Identification of the mounting head: Results from the residual analysis of the identified PWARX model with $s = 4$ discrete modes. Validation data are used

to be due to incomplete information provided by the estimation data. Indeed, in the estimation data set (upper plot of Figure 5.8), all significant transitions of the output from low to high values show an overshoot. Consequently, an overshoot shows up in the simulated responses on the intervals $(60,140)$ and $(210,240)$, that are both generated by the same sequence of affine submodels, and are caused by large variations of the input signal. It is interesting to note that the identified model with $s = 4$ discrete modes is able to reproduce very faithfully the peak in the interval $(60,140)$. The discrete mode evolution in the lower plot of Figure 5.9(a) clearly shows that one of the two submodels is active in situations of high incoming velocity of the mounting head (*i.e.*, rapid transitions from low to high values of the mounting head position). One submodel modelling the same situation is also present in the identified models with $s = 3$ and $s = 4$ discrete modes.

Figure 5.10 shows a whiteness test of the residuals, and a cross-correlation test between the residuals and the input, for the identified PWARX model with $s = 4$ discrete modes. Values of the auto-correlation sequence at lags $\tau = 1$ and $\tau = 2$ fall outside the confidence region. This can be explained by observing that there are large intervals in Figure 5.11 where the prediction error is almost constant.

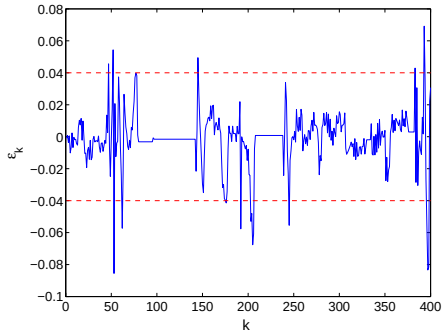


Figure 5.11 Identification of the mounting head: Plot of the prediction error using the identified PWARX model with $s = 4$ discrete modes. Validation data are used

Hence, auto-correlation at small lags is high. Comparison of Figures 5.8 and 5.11 shows that constant intervals of the prediction error are likely a consequence of the input signal of piecewise constant type used to excite the system. No significant cross-correlation between the residuals and the input is shown by the lower plot of Figure 5.10.

In order to compare PWARX model structures with different orders, a PWARX model with orders $n_a = 2$ and $n_b = 1$, and $s = 3$ discrete modes was also identified and validated using the same data sets of Figure 5.8. The plot of the simulated response of this model is shown in Figure 5.12. The corresponding *VAF* was 88.20%, which is 2% lower than that corresponding to the identified PWARX model with orders $n_a = 2$ and $n_b = 2$, and the same number $s = 3$ of discrete modes (see Table 5.5). However, this difference is not significant enough to accord clear preference to the latter model over the former.

This case study showed that suitable PWARX models, which are able to describe relevant aspects of the dynamics of a real process, can be obtained by the use of the proposed PWA system identification procedure. It also demonstrated that the bound δ of the procedure can effectively be used as a tuning knob to trade off between model complexity and quality of fit.

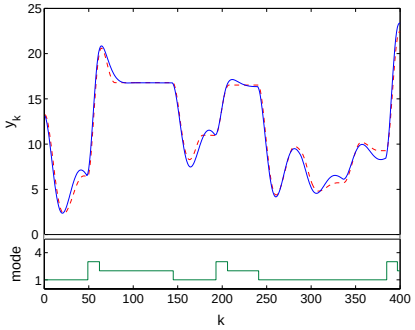


Figure 5.12 Identification of the mounting head: Simulation results of the identified PWARX model with orders $n_a = 2$ and $n_b = 1$, and $s = 3$ discrete modes (solid line - simulated output, dashed line - system output). The lower plot shows the evolution of the discrete mode

6

Conclusions

In recent years, PWA system identification has deserved increasing attention, motivated by the development of always more advanced tools for the analysis, verification and control of hybrid systems, and by the equivalence between PWA systems and several classes of hybrid systems.

In this thesis, a novel procedure for the identification of PWARX models from input-output data was presented and discussed. The proposed two-stage procedure, described in Chapter 3, consists of a first step, where data classification and parameter estimation are performed simultaneously, and a second step, which deals with the estimation of the regions. The key approach in the first step was the selection of a bound δ on the fitting error. This made it possible to formulate the problem of data classification and parameter estimation as an extension of the MIN PFS problem for infeasible systems of linear inequalities. The major capability of this formulation is that it also provides an estimate of the minimum number of submodels needed to fit the data, which therefore is not fixed a priori. A refinement procedure for improving both data classification and parameter estimation was also proposed. It alternates between data point reassignment and parameter update. Outliers can be detected and discarded in this phase. Moreover, the number of submodels is allowed to vary from iteration to iteration by exploiting parameter similarities and cluster cardinalities. The ambiguity related to the classification of the undecidable data points was

pointed out, and it was suggested to assign them to submodels by exploiting spatial localization. Region estimation was finally carried out by resorting to linear separation techniques. Chapter 4 was dedicated to an overview of several of these techniques, such as (Multicategory) Support Vector Machines and (Multicategory) Robust Linear Programming. According to the bounded error description, the identified PWA model associates to each submodel a set of feasible parameters, thus allowing for evaluation of the related parametric uncertainty.

The greedy algorithm (Amaldi and Mattavelli, 2002) used to solve the MIN PFS problem with complementary inequalities was modified in this thesis in order to obtain improved solutions. The choice of the bound δ on the fitting error was also discussed. It was shown that a suitable choice of δ is typically close to the knee of the curve expressing the number of submodels versus δ. Interesting results were obtained by applying the identification procedure to experimental data from an electronic component placement process. These results demonstrated that the bound δ can effectively be used as a tuning knob to trade off between model complexity and quality of fit. It is pointed out that considering a common error bound was not restrictive, because the case of different bounds associated with each data point can be handled by scaling the data so that $\delta = 1$. The refinement procedure proposed in thesis as a part of the overall PWA system identification procedure, could be combined with other procedures available in the literature, so as to yield new identification methods. For instance, the rank constraint on the data derived by Vidal *et al.* (2003*a*) could be used to estimate the number of submodels, and then the *k*-plane algorithm proposed in (Bradley and Mangasarian, 2000) could be exploited to provide initial data classification and parameter estimation.

Future research will concern the possibility to include in the identification procedure the knowledge available a priori on the system to be identified (*e.g.*, saturations, thresholds, dead-zones), as well as to identify submodels of different orders for each discrete mode. Techniques for efficiently detecting and handling non-convex regions, or non-connected regions where the parameter vector is the same, will be investigated. Effort will be addressed to deriving rules for automatic

selection and update of the thresholds α and β in the refinement step, in order to completely automatize the identification procedure.

It would be interesting to define suitable criteria for evaluating the quality of the identified PWA models. Classical criteria like residual analysis might be misleading for this class of models. Since the partition of the PWA map cannot be determined exactly from a given finite estimation data set, even small errors in estimating the boundaries of the regions could determine large residuals, if the PWA map is discontinuous. In this respect, it would be also useful to provide bounds on the errors when reconstructing the regions. Experiment design and order selection are open issues in system identification, that hold also for the class of PWA models. In particular, the choice of the input signal for identification should be such that not only all the affine dynamics are sufficiently excited, but also accurate shaping of the boundaries of the regions will be possible. This is especially important when the system dynamics is discontinuous. Other open questions related to PWA system identification are the identification of state space models, and the development of on-line algorithms.

Bibliography

Agmon, S. (1954). The relaxation method for linear inequalities. *Canadian Journal of Mathematics* **6**, 382–392.

Amaldi, E. and M. Mattavelli (2002). The MIN PFS problem and piecewise linear model estimation. *Discrete Applied Mathematics* **118**, 115–143.

Amaldi, E. and R. Hauser (2001). Randomized relaxation methods for the maximum feasible subsystem problem. Technical Report 2001-90. DEI, Politecnico di Milano. Italy.

Amaldi, E. and V. Kann (1995). The complexity and approximability of finding maximum feasible subsystems of linear relations. *Theoretical Computer Science* **147**(1-2), 181–210.

Antsaklis, P.J. and A. Nerode, eds. (1998). Special issue on hybrid systems. *IEEE Transactions on Automatic Control* **43**(4), 457–579.

Bai, E.-W. (2002). Identification of linear systems with hard input nonlinearities of known structure. *Automatica* **38**(5), 853–860.

Batruni, R. (1991). A multilayer neural network with piecewise-linear structure and back-propagation learning. *IEEE Transactions on Neural Networks* **2**(3), 395–403.

Bemporad, A., A. Garulli, S. Paoletti and A. Vicino (2003*a*). A greedy approach to identification of piecewise affine models. In: *Hybrid Systems: Computation*

and Control (O. Maler and A. Pnueli, Eds.). Vol. 2623 of *Lecture Notes in Computer Science*. pp. 97–112. Springer Verlag.

Bemporad, A., A. Garulli, S. Paoletti and A. Vicino (2003*b*). Set membership identification of piecewise affine models. In: *Proceedings of the 13th IFAC Symposium on System Identification*. Rotterdam, The Netherlands. pp. 1826–1831.

Bemporad, A. and M. Morari (1999). Control of systems integrating logic, dynamics, and constraints. *Automatica* **35**(3), 407–427.

Bemporad, A., D. Mignone and M. Morari (1999). Moving horizon estimation for hybrid systems and fault detection. In: *Proceedings of the American Control Conference*. Chicago, IL. pp. 2471–2475.

Bemporad, A., F.D. Torrisi and M. Morari (2000*a*). Optimization-based verification and stability characterization of piecewise affine and hybrid systems. In: *Hybrid Systems: Computation and Control* (N.A. Lynch and B.H. Krogh, Eds.). Vol. 1790 of *Lecture Notes in Computer Science*. pp. 45–58. Springer Verlag.

Bemporad, A., G. Ferrari-Trecate and M. Morari (2000*b*). Observability and controllability of piecewise affine and hybrid systems. *IEEE Transactions on Automatic Control* **45**(10), 1864–1876.

Bennett, K.P. and O.L. Mangasarian (1992). Robust linear programming discrimination of two linearly inseparable sets. *Optimization Methods and Software* **1**, 23–34.

Bennett, K.P. and O.L. Mangasarian (1994). Multicategory discrimination via linear programming. *Optimization Methods and Software* **3**, 27–39.

Billings, S.A. and W.S.F. Voon (1987). Piecewise linear identification of nonlinear systems. *International Journal of Control* **46**(1), 215–235.

Blondel, V.D. and J.N. Tsitsiklis (1999). Complexity of stability and controllability of elementary hybrid systems. *Automatica* **35**(3), 479–489.

Bradley, P.S. and O.L. Mangasarian (2000). *k*-plane clustering. *Journal of Global Optimization* **16**, 23–32.

Branicky, M.S. (1998). Multiple Lyapunov functions and other analysis tools for switched and hybrid systems. *IEEE Transactions on Automatic Control* **43**(4), 475–482.

Branicky, M.S., V.S. Borkar and S.K. Mitter (1998). A unified framework for hybrid control: model and optimal control theory. *IEEE Transactions on Automatic Control* **43**(1), 31–45.

Bredensteiner, E.J. and K.P. Bennett (1999). Multicategory classification by support vector machines. *Computational Optimization and Applications* **12**, 53–79.

Breiman, L. (1993). Hinging hyperplanes for regression, classification, and function approximation. *IEEE Transactions on Information Theory* **39**(3), 999–1013.

Broman, V. and M.J. Shensa (1990). A compact algorithm for the intersection and approximation of n-dimensional polytopes. *Mathematics and Computers in Simulation* **32**(5-6), 469–480.

Chan, K.S. and H. Tong (1986). On estimating thresholds in autoregressive models. *Journal of Time Series Analysis* **7**(3), 179–190.

Chisci, L., A. Garulli, A. Vicino and G. Zappa (1998). Block recursive parallelotopic bounding in set membership identification. *Automatica* **34**(1), 15–22.

Choi, C.-H. and J. Y. Choi (1994). Constructive neural networks with piecewise interpolation capabilities for function approximations. *IEEE Transactions on Neural Networks* **5**(6), 936–944.

Chua, L.O. and A.-C. Deng (1988). Canonical piecewise-linear representation. *IEEE Transactions on Circuits and Systems* **35**(1), 101–111.

Chua, L.O. and R.L.P. Ying (1983). Canonical piecewise-linear analysis. *IEEE Transactions on Circuits and Systems* **30**(3), 125–140.

Chua, L.O., M. Hasler, J. Neirynck and P. Verburgh (1982). Dynamics of a piecewise-linear resonant circuit. *IEEE Transactions on Circuits and Systems* **29**(8), 535–547.

Chutinan, A. and B.H. Krogh (2003). Computational techniques for hybrid system verification. *IEEE Transactions on Automatic Control* **48**(1), 64–75.

Cortes, C. and V. Vapnik (1995). Support-vector networks. *Machine Learning* **20**, 273–297.

Cristianini, N. and J. Shawe-Taylor (2000). *An Introduction to Support Vector Machines*. Cambridge University Press.

De Schutter, B. (2000). Optimal control of a class of linear hybrid systems with saturation. *SIAM Journal on Control and Optimization* **39**(3), 835–851.

De Schutter, B. and T.J.J. Van den Boom (2001). Model predictive control for max-min-plus-scaling systems. In: *Proceedings of the American Control Conference*. Arlington, VA. pp. 319–324.

Ernst, S. (1998). Hinging hyperplane trees for approximation and identification. In: *Proceedings of the 37th IEEE Conference on Decision and Control*. Vol. 2. Tampa, FL. pp. 1266–1271.

Ferrari-Trecate, G. and M. Muselli (2003). Single-linkage clustering for optimal classification in piecewise affine regression. In: *Proceedings of the IFAC Conference on Analysis and Design of Hybrid Systems*. Saint Malo, France.

Ferrari-Trecate, G., M. Muselli, D. Liberati and M. Morari (2003). A clustering technique for the identification of piecewise affine systems. *Automatica* **39**(2), 205–217.

Fogel, E. and Y.F. Huang (1982). On the value of information in system identification-bounded noise case. *Automatica* **18**(2), 229–238.

Gad, E. F., A. F. Atiya, S. Shaheen and A. El-Dessouki (2000). A new algorithm for learning in piecewise-linear neural networks. *Neural Networks* **13**(4-5), 485–505.

Heemels, W.P.M.H., B. De Schutter and A. Bemporad (2001). Equivalence of hybrid dynamical models. *Automatica* **37**(7), 1085–1091.

Heemels, W.P.M.H., J.M. Schumacher and S. Weiland (2000). Linear complementarity systems. *SIAM Journal of Applied Mathematics* **60**(4), 1234–1269.

Heredia, E.A. and G.R. Arce (1996). Piecewise linear system modeling based on a continuous threshold decomposition. *IEEE Transactions on Signal Processing* **44**(6), 1440–1453.

Hush, D. R. and B. Horne (1998). Efficient algorithms for function approximation with piecewise linear sigmoidal networks. *IEEE Transactions on Neural Networks* **9**(6), 1129–1141.

Johansson, M. and A. Rantzer (1998). Computation of piecewise quadratic Lyapunov functions for hybrid systems. *IEEE Transactions on Automatic Control* **43**(4), 555–559.

Juditsky, A., H. Hjalmarsson, A. Benveniste, B. Delyon, L. Ljung, J. Sjöberg and Q. Zhang (1995). Nonlinear black-box models in system identification: mathematical foundations. *Automatica* **31**(12), 1725–1750.

Julián, P., A. Desages and O. Agamennoni (1999). High level canonical piecewise linear representation using a simplicial partition. *IEEE Transactions on Circuits and Systems – I: Fundamental Theory and Applications* **46**(4), 463–480.

Julian, P., M. Jordan and A. Desages (1998). Canonical piecewise-linear approximation of smooth functions. *IEEE Transactions on Circuits and Systems – I: Fundamental Theory and Applications* **45**(5), 567–571.

Juloski, A.Lj., W.P.M.H. Heemels and G. Ferrari-Trecate (2003). Identification of an experimental hybrid system. In: *Proceedings of the IFAC Conference on Analysis and Design of Hybrid Systems*. Saint Malo, France.

Kang, S.M. and L.O. Chua (1978). A global representation of multidimensional piecewise-linear functions with linear partitions. *IEEE Transactions on Circuits and Systems* **25**(11), 938–940.

Leenaerts, D.M.W. and W.M.G. Van Bokhoven (1998). *Piecewise Linear Modeling and Analysis*. Kluwer Academic Publishers.

Liberzon, D. and A.S. Morse (1999). Basic problems in stability and design of switched systems. *IEEE Control Systems Magazine* **19**(5), 59–70.

Lin, J.-N. and R. Unbehauen (1992). Canonical piecewise-linear approximations. *IEEE Transactions on Circuits and Systems – I: Fundamental Theory and Applications* **39**(8), 697–699.

Ljung, L. (1999). *System Identification : Theory for the User*. 2nd ed.. Prentice Hall.

Ljung, L. (2003). *System Identification Toolbox User's Guide*. 6 ed.. The MathWorks, Inc.

Lunze, J. (2000). Diagnosis of quantized systems based on a timed discrete-event model. *IEEE Transactions on Systems, Man & Cybernetics, Part A* **30**(3), 322–335.

Lygeros, J., C. Tomlin and S. Sastry (1999). Controllers for reachability specifications for hybrid systems. *Automatica* **35**(3), 349–370.

Mangasarian, O.L. (1994). Misclassification minimization. *Journal of Global Optimization* **5**(4), 309–323.

Mangasarian, O.L. (1999). Arbitrary-norm separating plane. *Operations Research Letters* **24**(1-2), 15–23.

Medeiros, M.C., A. Veiga and M.G.C. Resende (2002). A combinatorial approach to piecewise linear time series analysis. *Journal of Computational and Graphical Statistics* **11**(1), 236–258.

Milanese, M. and A. Vicino (1991). Optimal estimation theory for dynamic systems with set membership uncertainty: an overview. *Automatica* **27**(6), 997–1009.

Milanese, M. and R. Tempo (1985). Optimal algorithms theory for robust estimation and prediction. *IEEE Transactions on Automatic Control* **30**(8), 730–738.

Milanese, M., J.P. Norton, H. Piet-Lahanier and E. Walter (eds.) (1996). *Bounding Approaches to System Identification*. Plenum Press. New York.

Morse, A.S., C.C. Pantelides, S.S. Sastry and J.M. Schumacher, eds. (1999). Special issue on hybrid systems. *Automatica* **35**(3), 347–535.

Motzkin, T.S. and I.J. Schoenberg (1954). The relaxation method for linear inequalities. *Canadian Journal of Mathematics* **6**, 393–404.

Münz, E. and V. Krebs (2002). Identification of hybrid systems using apriori knowledge. In: *Proceedings of the 15th IFAC World Congress*. Barcelona, Spain.

Ninness, B. and G.C. Goodwin (1995). Estimation of model quality. *Automatica* **31**(12), 1771–1797.

Pearson, R.K. (1988). Block-sequential algorithms for set-theoretic estimation. *SIAM Journal on Matrix Analysis and Apllications* **9**(4), 513–527.

Pfetsch, M.E. (2002). The Maximum Feasible Subsystem Problem and Vertex-Facet Incidences of Polyhedra. PhD thesis. Technischen Universität Berlin.

Piet-Lahanier, H. and E. Walter (1993). Polyhedric approximation and tracking for bounded-error models. In: *Proceedings of the IEEE International Symposium on Circuits and Systems*. Chicago, IL,. pp. 782–785.

Pucar, P. and J. Sjöberg (1998). On the hinge-finding algorithm for hinging hyperplanes. *IEEE Transactions on Information Theory* **44**(3), 1310–1319.

Pupeikis, R. (2003). Identification of piecewise affine Wiener systems using data partition. Technical Report LiTH-ISY-R-2523. Department of Electrical Engineering, Linköping University. Linköping, Sweden.

Pupeikis, R., D. Navakauskas and L. Ljung (2003). Identification of Wiener systems with hard and discontinuous nonlinearities. Technical Report LiTH-ISY-R-2501. Department of Electrical Engineering, Linköping University. Linköping, Sweden.

Ragot, J., G. Mourot and D. Maquin (2003). Parameter estimation of switching piecewise linear systems. In: *Proceedings of the 42nd IEEE Conference on Decision and Control*. Maui, Hawaii. pp. 5783–5788.

Roll, J. (2003). Local and Piecewise Affine Approaches to System identification. PhD thesis. Department of Electrical Engineering, Linköping University. Linköping, Sweden.

Roll, J., A. Bemporad and L. Ljung (2004). Identification of piecewise affine systems via mixed-integer programming. *Automatica* **40**, 37–50.

Schweppe, F.C. (1968). Recursive state estimation: Unknown but bounded errors and system inputs. *IEEE Transactions on Automatic Control* **13**(1), 22–28.

Sjöberg, J., Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P. Glorennec, H. Hjalmarsson and A. Juditsky (1995). Nonlinear black-box modeling in system identification: a unified overview. *Automatica* **31**(12), 1691–1724.

Skeppstedt, A., L. Ljung and M. Millnert (1992). Construction of composite models from observed data. *International Journal of Control* **55**(1), 141–152.

Sontag, E.D. (1981). Nonlinear regulation: The piecewise linear approach. *IEEE Transactions on Automatic Control* **26**(2), 346–358.

Sontag, E.D. (1996). Interconnected automata and linear systems: A theoretical framework in discrete-time. In: *Hybrid Systems III - Verification and Control*

(R. Alur, T.A. Henzinger and E.D. Sontag, Eds.). Vol. 1066 of *Lecture Notes in Computer Science*. pp. 436–448. Springer-Verlag.

Strömberg, J.-E., F. Gustafsson and L. Ljung (1991). Trees as black-box model structures for dynamical systems. In: *Proceedings of the European Control Conference*. Grenoble, France. pp. 1175–1180.

Sun, Z., S.S. Ge and T.H. Lee (2002). Controllability and reachability criteria for switched linear systems. *Automatica* **38**(5), 775–786.

Tong, H. (1983). *Threshold Models in Non-linear Time Series Analysis*. Vol. 21 of *Lecture Notes in Statistics*. Springer Verlag.

Van der Schaft, A.J. and J.M. Schumacher (1998). Complementarity modelling of hybrid systems. *IEEE Transactions on Automatic Control* **43**(4), 483–490.

Van der Schaft, A.J. and J.M. Schumacher (2000). *An Introduction to Hybrid Dynamical Systems*. Vol. 251 of *Lecture Notes in Control and Information Sciences*. Springer Verlag.

Vapnik, V.N. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag. New York.

Vapnik, V.N. (1998). *Statistical Learning Theory*. John Wiley.

Veres, S.M. (1994). Polyhedron updating and relaxation for on-line parameter and state bounding. In: *Proceedings of the 10th IFAC Symposium on System Identification*. Vol. 3. Copenhagen, Denmark. pp. 371–376.

Vicino, A. and G. Zappa (1996). Sequential approximation of feasible parameter sets for identification with set membership uncertainty. *IEEE Transactions on Automatic Control* **41**(6), 774–785.

Vidal, R., A. Chiuso, S. Soatto and S. Sastry (2003*a*). Observability of linear hybrid systems. In: *Hybrid Systems: Computation and Control* (O. Maler and

A. Pnueli, Eds.). Vol. 2623 of *Lecture Notes in Computer Science*. pp. 526–539. Springer Verlag.

Vidal, R., S. Soatto, Y. Ma and S. Sastry (2003*b*). 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*. Maui, Hawaii. pp. 167–172.

Vörös, J. (1997). Parameter identification of discontinuous Hammerstein systems. *Automatica* **33**(6), 1141–1146.

Vörös, J. (2001). Parameter identification of Wiener systems with discontinuous nonlinearities. *Systems & Control Letters* **44**(5), 363–372.

Witsenhausen, H.S. (1968). Sets of possible states of linear systems given perturbed observations. *IEEE Transactions on Automatic Control* **13**(5), 556–558.