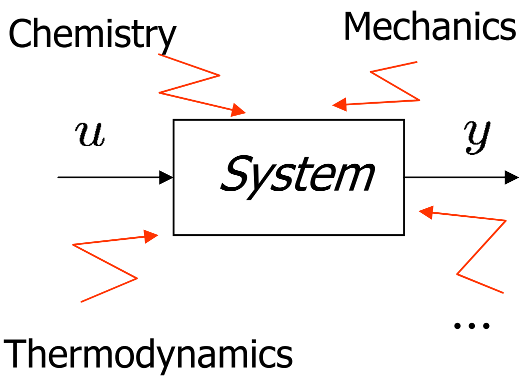
Identification algorithms for hybrid systems

Giancarlo Ferrari-Trecate

Modeling paradigms

White box



Drawbacks:

- Parameter values of components must be known
- Symplifying assumptions
- Not feasible if first-principles are not available (e.g. economics, biology,...)

Black box

Experimental data:

$$(u(k), y(k)), \; k = 1, \ldots N$$

⇓

Identification algorithm

⇓

Mathematical model

Huge literature on identification of linear and smooth, nonlinear systems

Hybrid identification

What about identification of hybrid models ? Is it really a problem ?

First guess:

- Each mode of operation is a linear/nonlinear system
- Resort to known identification methods for each mode !

Not always feasible !

Motivating example

Identification of an electronic component placement process

Fast component mounter (courtesy of Assembleon)

- 12 mounting heads working in parallel
- Maximum throughput: 96.000 components per hour

Placement of the electronic component on the Printed Circuit Board

Experimental setup

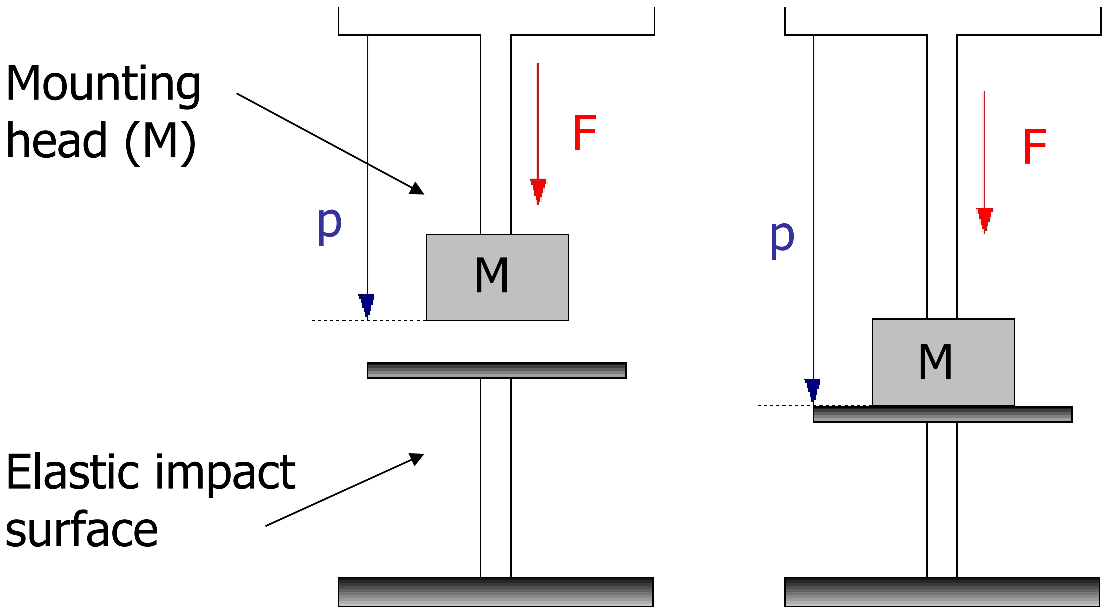
Mounting head

Moving impacting surface

Ground connection

Schematic representation

2 basic modes of operation :
- free mode
- impact mode



Input:
- Motor force F

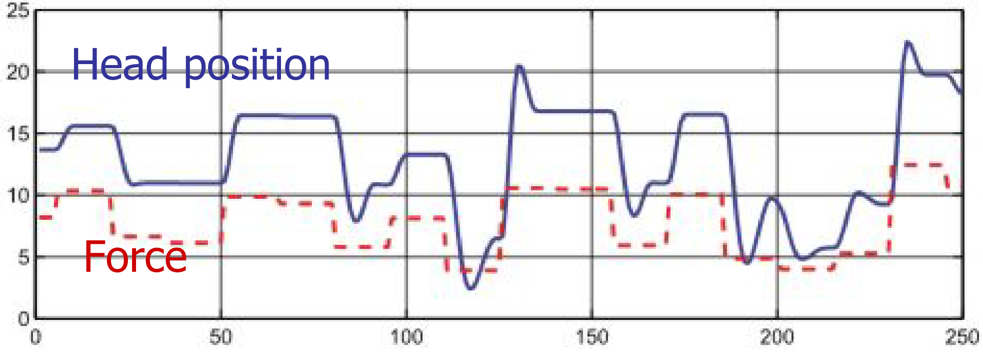
Output:
- Head position p

Problem: the position of the impact surface is not measured

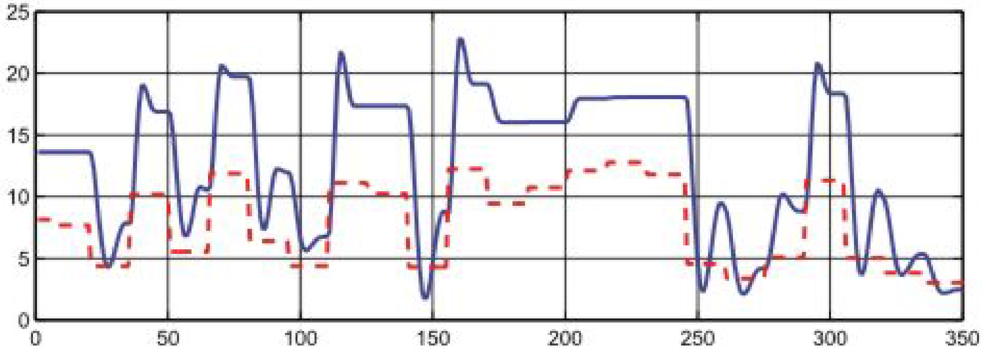
The switch between modes is not measured

Experimental data

Identification data



Validation data



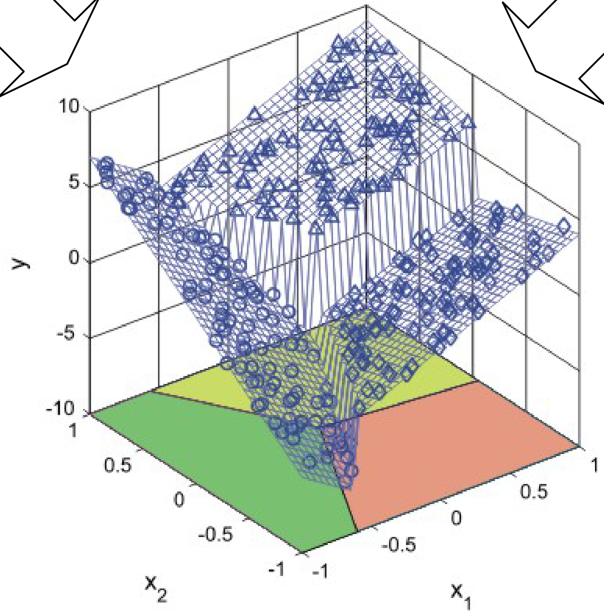
Which are the data generated in the free and impact modes ?
How to reconstruct the switching rule ?

Hybrid identification

Data could be naturally labeled according to finitely many modes of operation

Each mode has unknown quantitative dynamics

3 modes



Each mode has a linear behavior

Goals:

extract, at the same time, the switching mechanism and the mode dynamics

Hybrid identification: applications

Domains:

- Engineering: mechanical systems with contact phenomena, current transformers, traction control
- Computer vision: motion segmentation (Wang & Adelson, 1993), (Vidal et *al.,* , 2006-2007)
- Signal processing: signal segmentation (Heredia & Gonzalo, 1996)
- Biology and medicine: sleep apneas detection form ECG, pulse detection from hormone concentration, ...

Tomorrow: identification of genetic regulatory networks
(HYGEIA PhD School on Hybrid Systems Biology)

Outline of the lectures

1. Preliminaries: polytopes, PWA maps, PWARX models

2. Identification of PWARX models

3. The key difficulty: classification of the data points

4. Three identification algorithms:
   - Clustering-based procedure
   - Algebraic procedure
   - Bounded-error procedure

5. Back to the motivating example: identification results

Preliminaries: polytopes, PWA maps, PWARX models

Preliminaries: polytopes

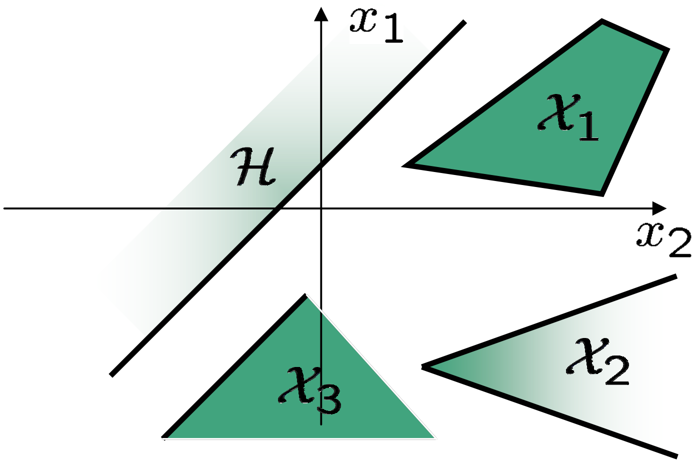
Let $x \in \mathbb{R}^n$

Hyperplane: $\{x : a'x = \beta\}$

Half-space: $\mathcal{H} = \{x : a'x \leq \beta\}$

Polyhedron: $\mathcal{X} = \{x : Ax \leq b\}$

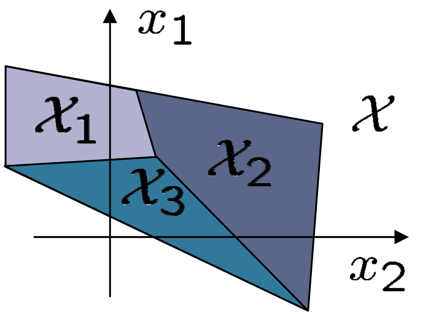
$$A = \begin{bmatrix} a'_1 \\ \vdots \\ a'_h \end{bmatrix} \quad b = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_h \end{bmatrix}$$



- Polyhedra are convex and closed sets

Polytope: bounded polyhedron

Not Necessarily Closed (NNC) polyhedron:
$\mathcal{X} \subset \mathbb{R}^n$ convex and s.t. Cl$(\mathcal{X})$ is a polyhedron

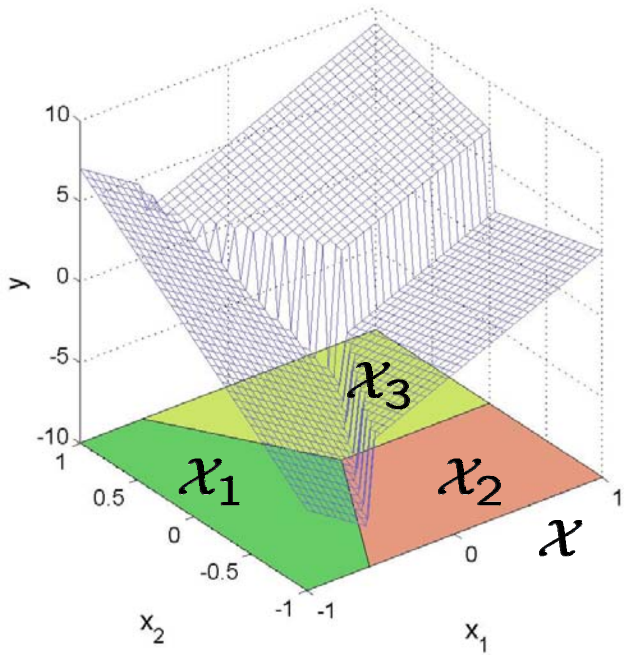


Polyhedral partition of the polyhedron $\mathcal{X}$:
finite collection of NNC polyhedra $\mathcal{X}_i$ such that $\cup_i \mathcal{X}_i = \mathcal{X}$ and $\mathcal{X}_i \cap \mathcal{X}_j = \emptyset, \ i \neq j$

Preliminaries: PWA maps

- $\{\mathcal{X}_i\}_{i=1}^s$ is a polyhedral partition of the polytope $\mathcal{X} \subset \mathbb{R}^n$
- Switching function: $\lambda(x) = i \Leftrightarrow x \in \mathcal{X}_i$

$$f(x) = \begin{cases} \theta_1' \begin{bmatrix} x \\ 1 \end{bmatrix} & \text{if } \lambda(x) = 1 \\ \vdots & \\ \theta_s' \begin{bmatrix} x \\ 1 \end{bmatrix} & \text{if } \lambda(x) = s \end{cases}$$



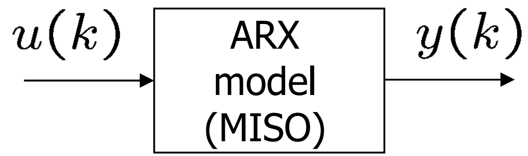
Ingredients:

- domain: $\mathcal{X}$
- number of modes: s
- modes: $(\theta_i, \mathcal{X}_i), \;\; i = 1, \ldots, s$
  - Parameter Vectors (PVs): θ_i
  - Regions: $\mathcal{X}_i$

PWARX models

AutoRegressive eXogenous (ARX) model of orders (n_a, n_b):

$$y(k) = \theta' x(k)$$



Vector of regressors:

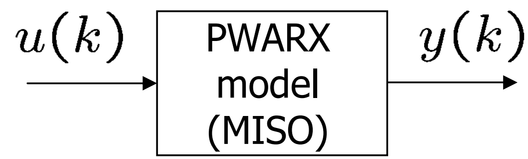
$$x(k) = \left[\begin{array}{cccccc} u'(k-1) & \dots & u'(k-n_a) & y(k-1) & \dots & y(k-n_b) \end{array}\right]'$$

$$u(k) \in \mathbb{R}^{n_u}, \;\; x(k) \in \mathbb{R}^{n}, \;\; n = n_u \cdot n_a + n_b$$

PieceWise ARX (PWARX) models of orders (n_a, n_b) :

$$y(k) = f(x(k))$$

- $f(\cdot)$ is a PWA map



PWA map: the interaction between logic/continuous components is modeled through discontinuities and the regions shape

Identification of PWARX models

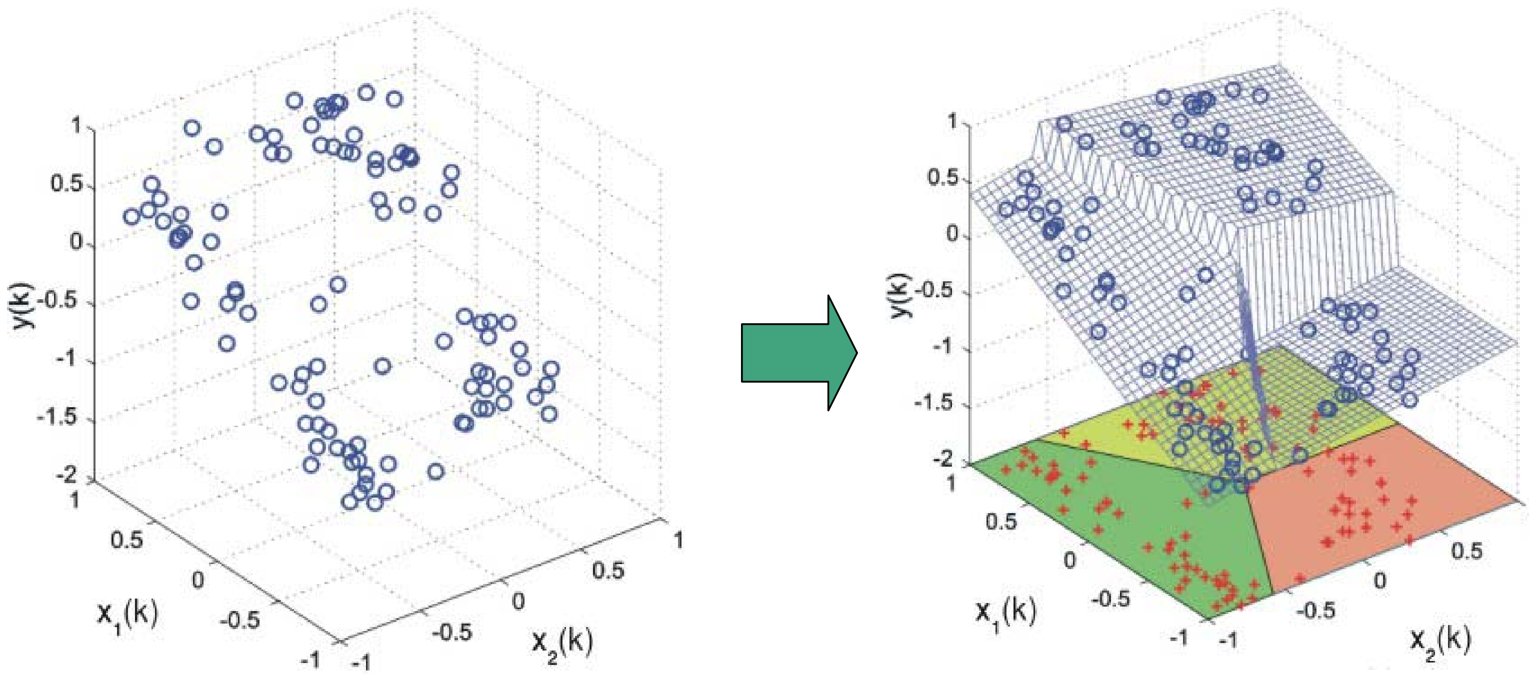
Identification of PWARX models

Dataset (noisy samples of a PWARX model) :

$$y(k) = f(x(k)) + \eta(k) \quad \eta : \text{noise}$$

$$\mathcal{N} = \{(x(k), y(x)),\ k = 1, \ldots, N\}$$

Identification: reconstruct the PWA map $f(\cdot)$ from $\mathcal{N}$



Identification of PWARX models

Dataset (noisy samples of a PWARX model) :

$$y(k) = f(x(k)) + \eta(k) \qquad \eta : \text{noise}$$

$$\mathcal{N} = \{(x(k), y(x)),\ k = 1, \ldots, N\}$$

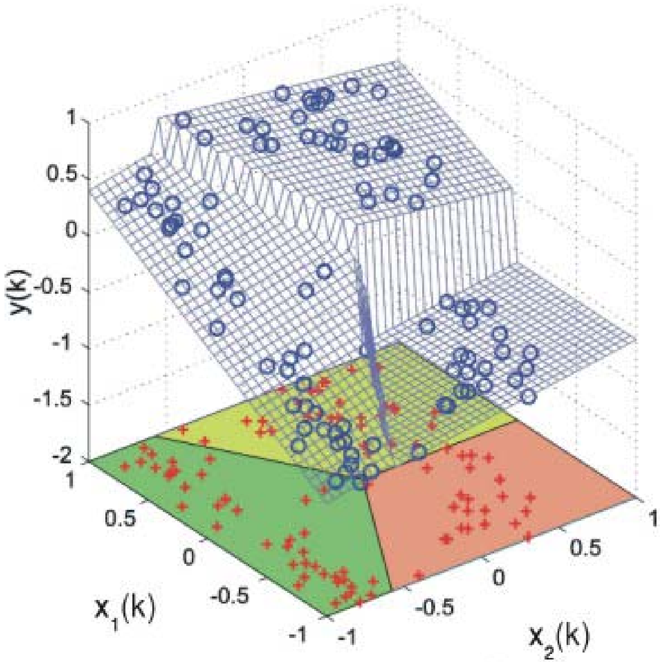
Identification: reconstruct the PWA map $f(\cdot)$ from $\mathcal{N}$

Standing assumptions:

1) Known model orders
2) Known regressor set $\mathcal{X} \subset \mathbb{R}^n$ (physical constraints)

Estimate:

- The number s of modes
- The PVs $\theta_i,\ i = 1, \ldots, s$
- The regions $\mathcal{X}_i,\ i = 1, \ldots, s$

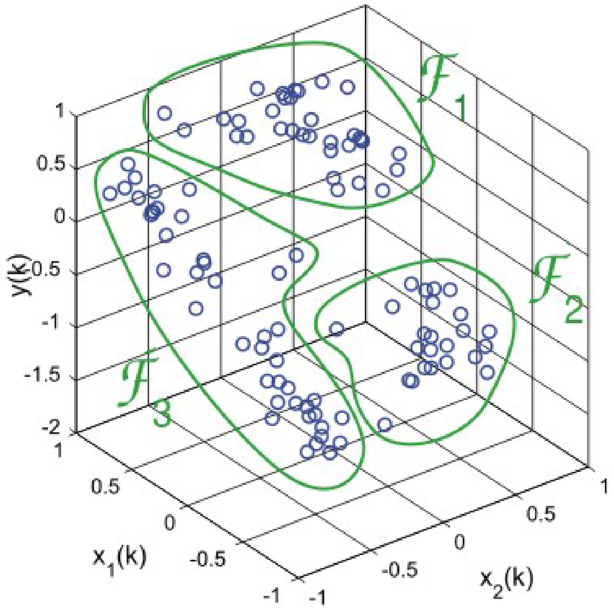


The key difficulty: data classification

Sub-problem: classification

Known switching sequence $\lambda(x(k))$

$$i^{th}\text{-mode dataset: } \mathcal{F}_i = \{(x(k), y(k)) : \lambda(x(k)) = i\}$$

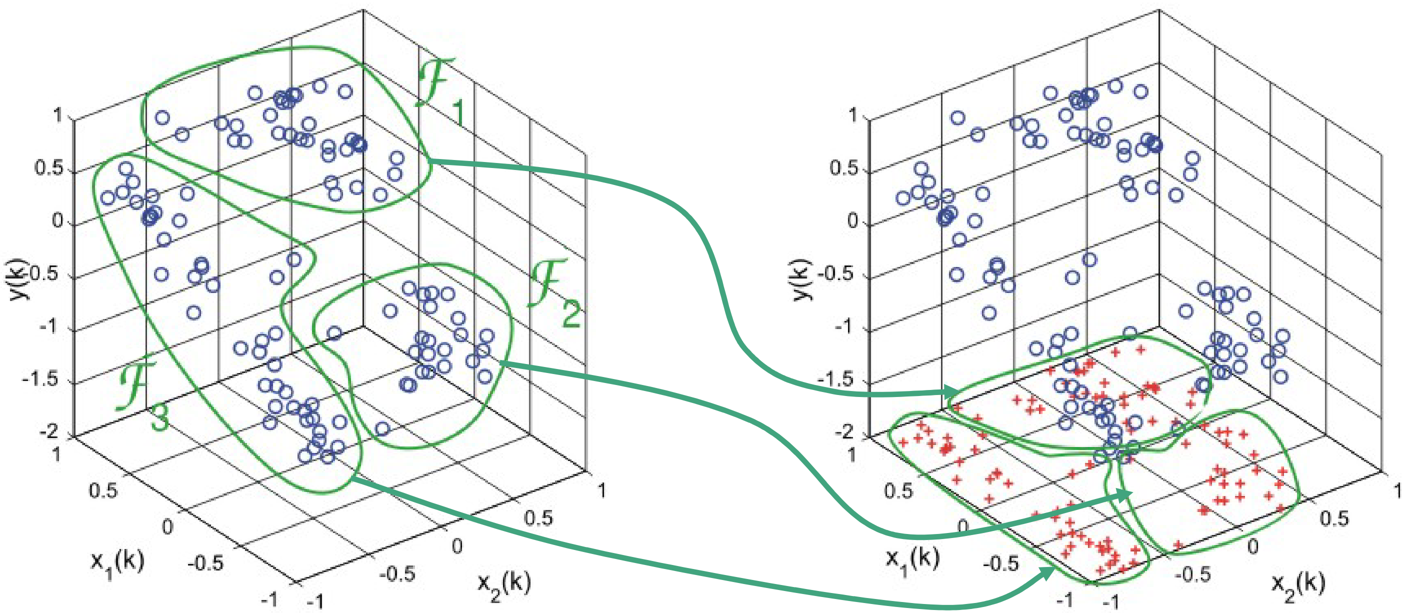


Sub-problem: classification

Known switching sequence $\lambda(x(k))$

$$i^{th}\text{-mode dataset: } \mathcal{F}_i = \{(x(k), y(k)) : \lambda(x(k)) = i\}$$

- Pattern recognition algorithms ⇨ Estimates of the regions

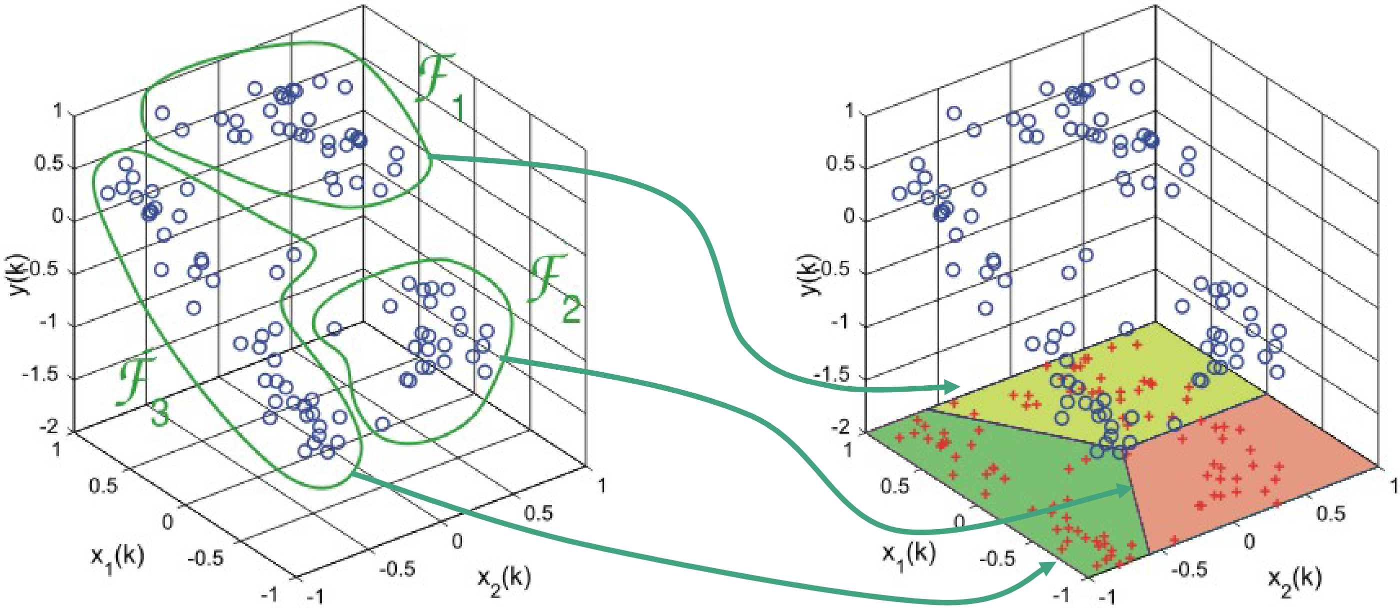


Sub-problem: classification

Known switching sequence $\lambda(x(k))$

$$i^{th}\text{-mode dataset: } \mathcal{F}_i = \{(x(k), y(k)) : \lambda(x(k)) = i\}$$

- Pattern recognition algorithms ⇨ Estimates of the regions

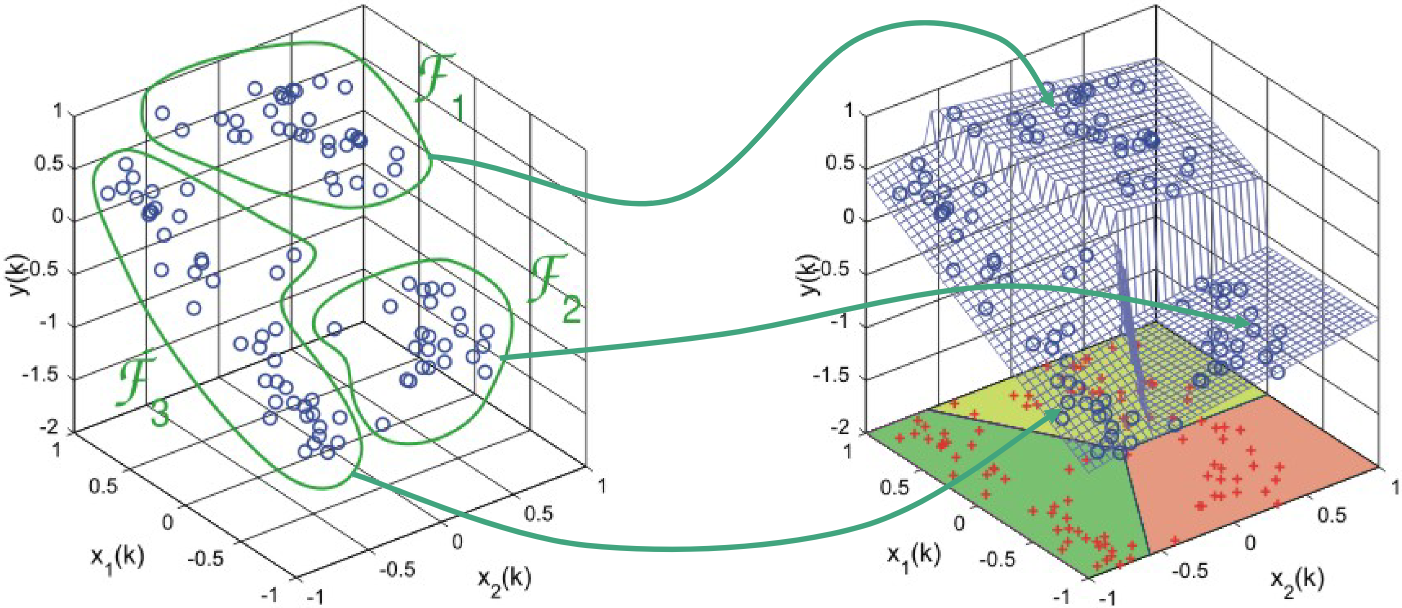


Sub-problem: classification

Known switching sequence $\lambda(x(k))$

i^{th}-mode dataset: $\mathcal{F}_i = \{(x(k), y(k)) : \lambda(x(k)) = i\}$

- Pattern recognition algorithms ⇨ Estimates of the regions
- Least squares on $\mathcal{F}_i$ ⇨ Estimates of the PVs

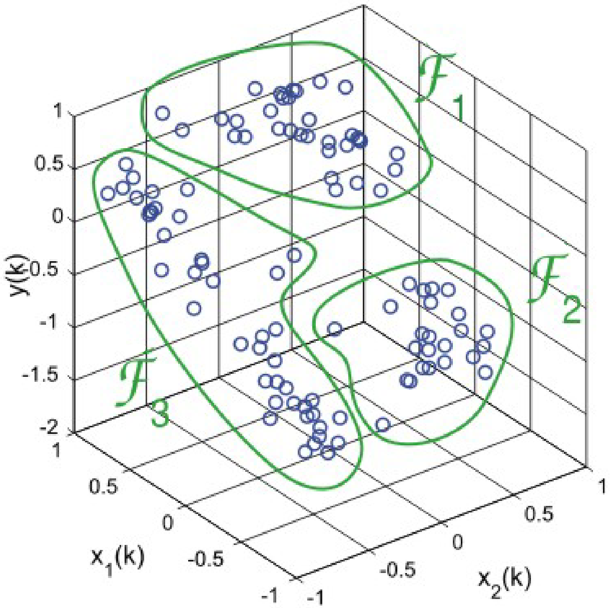


Sub-problem: classification

Known switching sequence $\lambda(x(k))$

$$i^{th}\text{-mode dataset: } \mathcal{F}_i = \{(x(k), y(k)) : \lambda(x(k)) = i\}$$

- Pattern recognition algorithms ⇨ Estimates of the regions
- Least squares on $\mathcal{F}_i$ ⇨ Estimates of the PVs



Classification problem:

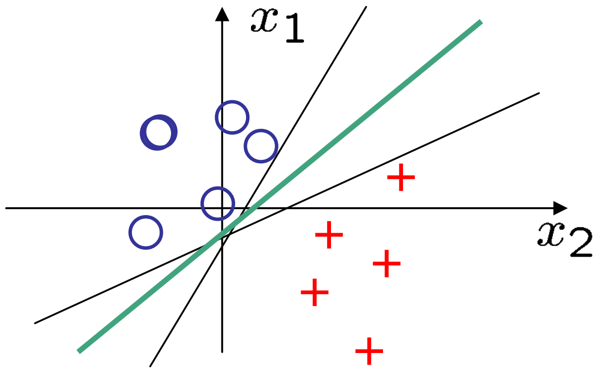
estimation of the switching sequence

All algorithms for the identification of PWARX models solve, implicitly or explicitly, the classification problem !

An introduction to pattern recognition

Pattern recognition: the two-class problem

Data: finite set of labeled points $x(k) \in \mathbb{R}^n$



- class 1: $\lambda(x(k)) = 1$
- class 2: $\lambda(x(k)) = 2$

Problem:

find a hyperplane $\{x : a'x = \beta\}$ that separates the two classes, i.e.

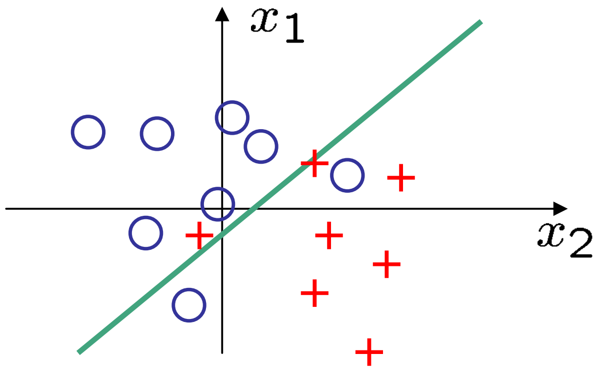
$$a'x(k) < \beta \quad \text{if} \quad \lambda(x(k)) = 1$$
$$a'x(k) > \beta \quad \text{if} \quad \lambda(x(k)) = 2$$

- If such a hyperplane exists, the classes are *linearly separable*
  - The separating hyperplane is not unique

The *optimal* (in a statistical sense) separating hyperplane is unique and can be computed by solving a quadratic program.
Sub-optimal hyperplanes can be computed through linear programs

Pattern recognition: the two-class problem

The inseparable case



Problem:
find an hyperplane that minimizes an error measure related to misclassified data points

Again:
The *optimal* separating hyperplane is unique and can be computed by solving a quadratic program (Support Vector Classification)

(Vapnik, 1998)

Sub-optimal hyperplanes can be computed through linear programs (Robust Linear Programming) (Bennet & Mangasarian, 1993)

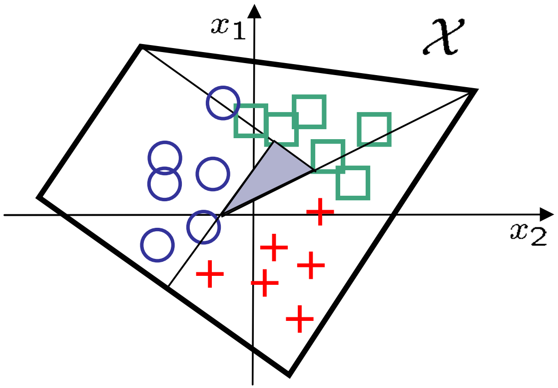
Estimation of the regions

The region $\mathcal{X}_i$ is defined by the $s-1$ hyperplanes separating $\{x(k):\lambda(x(k))=i\}$ from $\{x(k):\lambda(x(k))=j\},\ \forall j\neq i$

Two strategies:

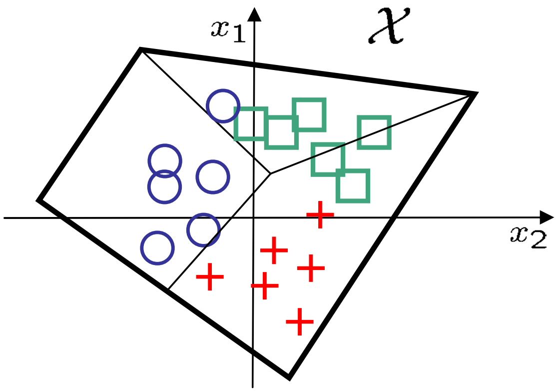
1) Pairwise separation:

- Many QPs/LPs of "small size"
  (the number of variables scales linearly with the number of data involved in each separation problem)
- Problem: possible "holes" in the set of regressors !



2) Multi-class separation:

- Separate *simultaneously* all classes
- Problem: existing algorithms amount to QPs/LPs of "big size"
  (the number of variables scales linearly with the *total* number of data)



Three identification algorithms

- Clustering-based procedure
- Algebraic procedure
- Bounded-error procedure

The clustering-based procedure

(Ferrari-Trecate et *al.*, 2003)

Clustering-based procedure: introduction

Standing assumptions:

1) known number of modes
2) $\theta_i \neq \theta_j, \; i \neq j$ (just for sake of simplicity)

Key idea:

PWA maps are locally linear. If the local models around two data points are similar, it is likely that the data points belong to the same mode of operation

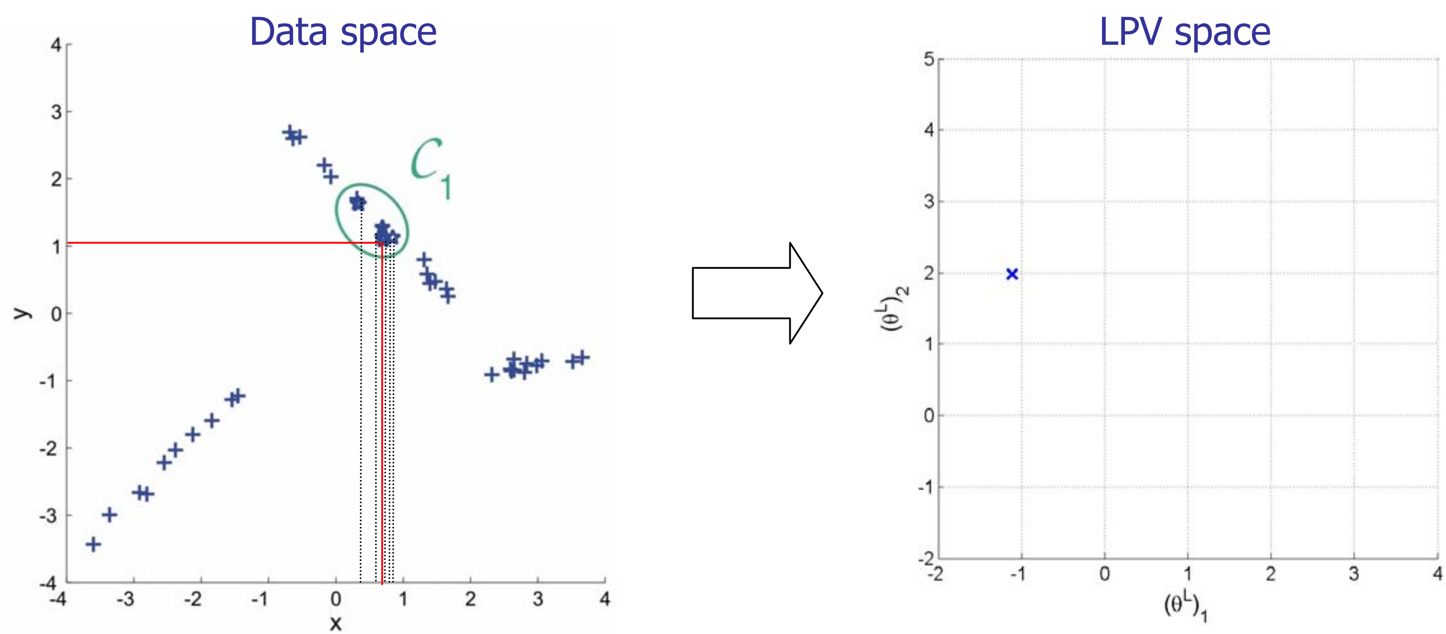
Steps of the algorithm:

1) associate to each data point a local affine model
2) aggregate local models with similar features into clusters
3) classify in the same way data points corresponding to local models in the same cluster

Clustering-based procedure - Step 1

Extract local models

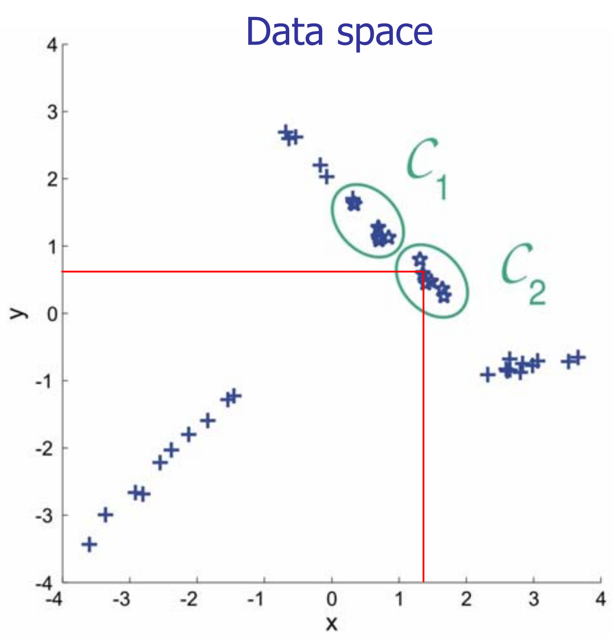
1. For each data point $(x(k), y(k))$ build a Local Dataset (LD) $\mathcal{C}_k$ collecting $(x(k), y(k))$ and its first $c-1$ neighboring points $(c > n)$
2. Fit a local linear model on each $\mathcal{C}_k$ through Least Squares
   - Local Parameter Vector (LPV) θ_k^L and associated variance V_k

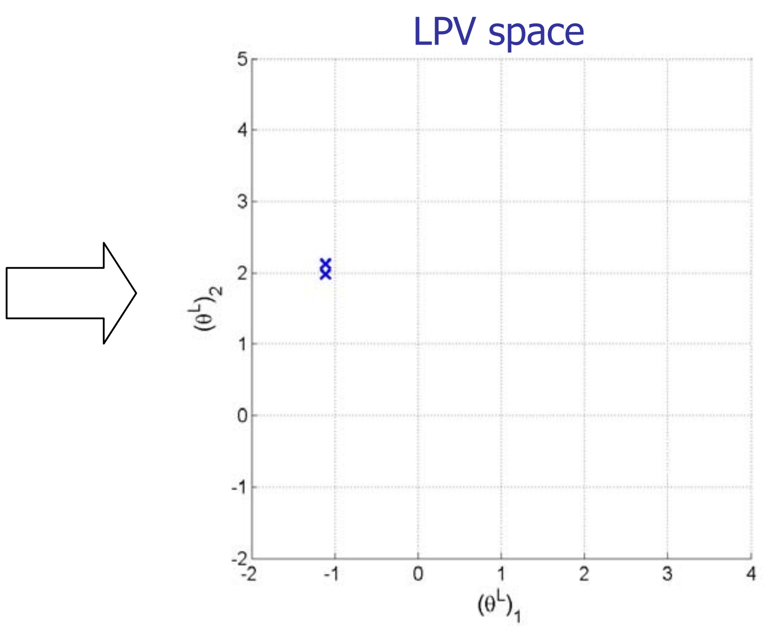


Clustering-based procedure - Step 1

Extract local models

1. For each data point $(x(k), y(k))$ build a Local Dataset (LD) $\mathcal{C}_k$ collecting $(x(k), y(k))$ and its first $c-1$ neighboring points $(c > n)$
2. Fit a local linear model on each $\mathcal{C}_k$ through Least Squares
   - Local Parameter Vector (LPV) θ_k^L and associated variance V_k

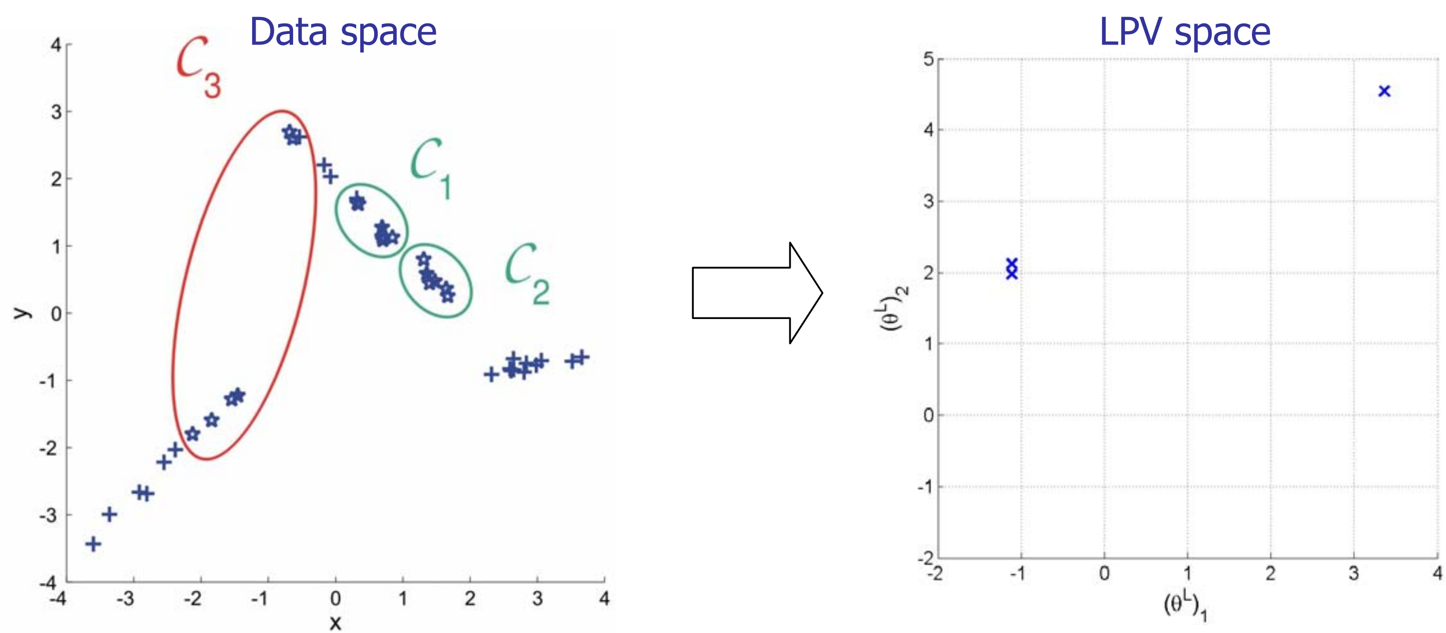




Clustering-based procedure - Step 1

Extract local models

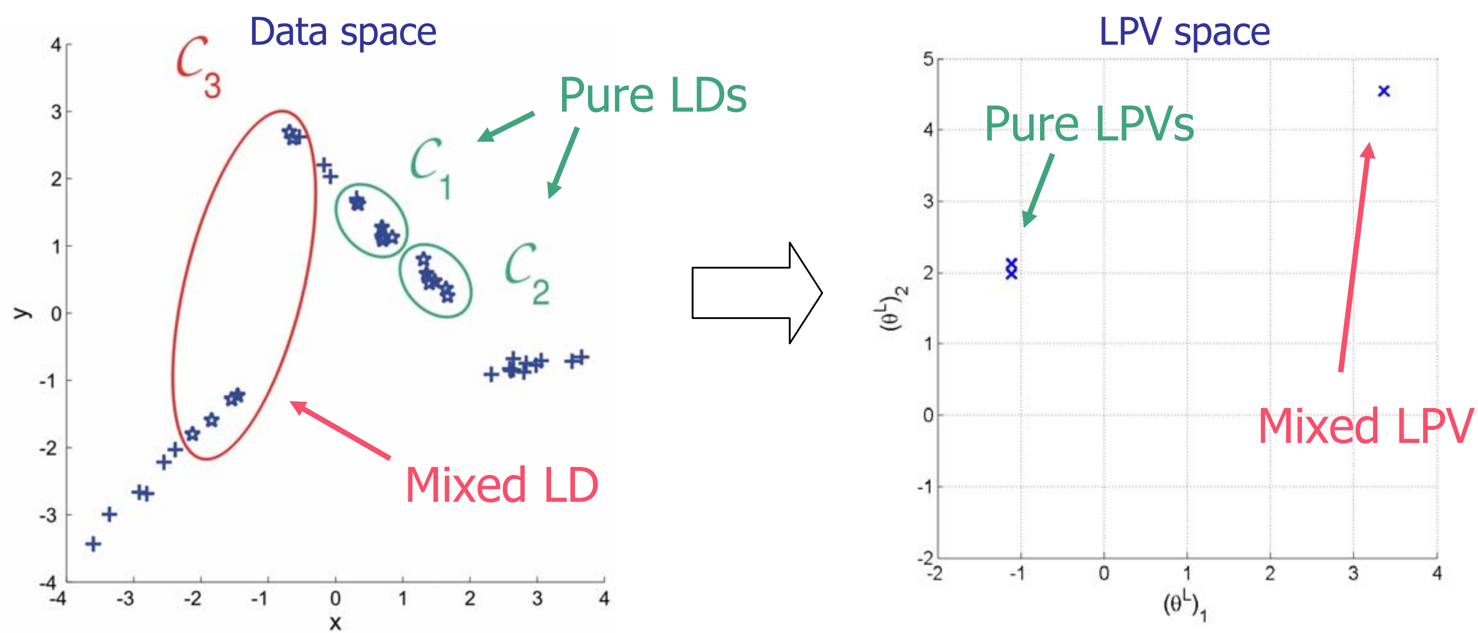
1. For each data point $(x(k), y(k))$ build a Local Dataset (LD) $\mathcal{C}_k$ collecting $(x(k), y(k))$ and its first $c-1$ neighboring points $(c > n)$
2. Fit a local linear model on each $\mathcal{C}_k$ through Least Squares
   - Local Parameter Vector (LPV) θ_k^L and associated variance V_k



Clustering-based procedure - Step 1

Extract local models

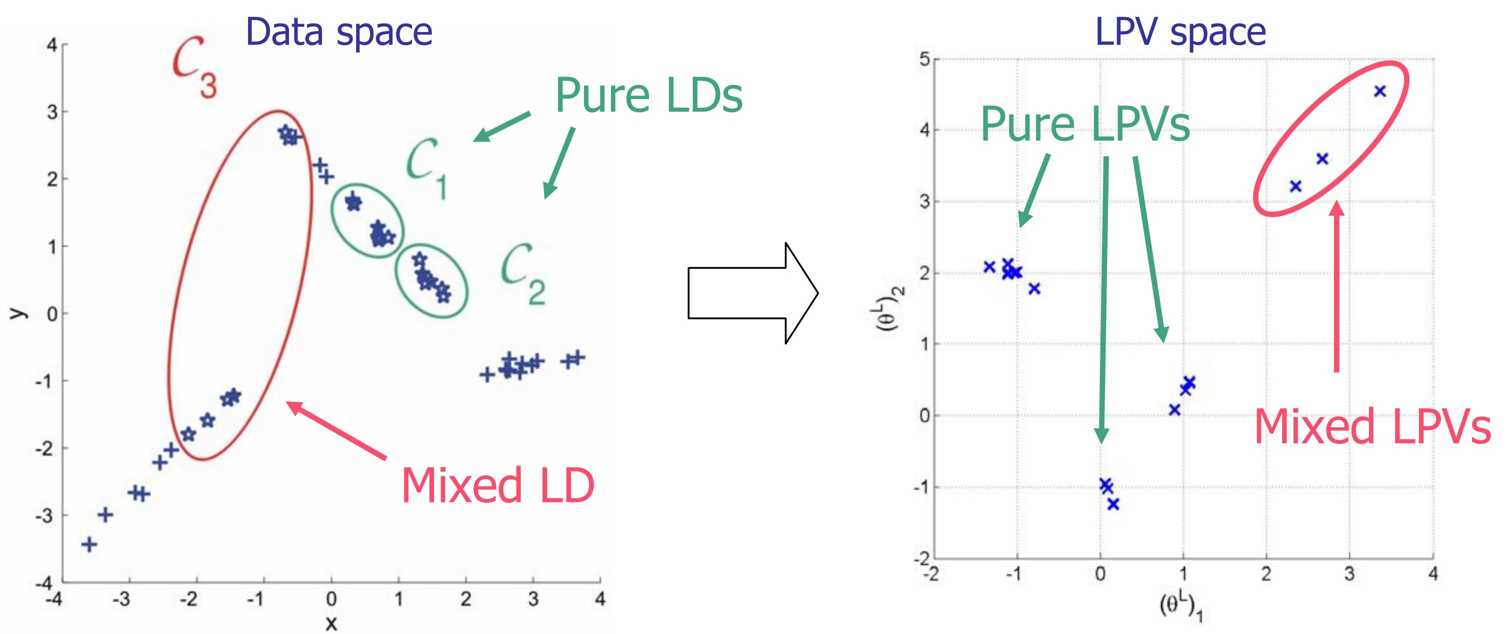
1. For each data point $(x(k), y(k))$ build a Local Dataset (LD) $\mathcal{C}_k$ collecting $(x(k), y(k))$ and its first $c-1$ neighboring points $(c > n)$
2. Fit a local linear model on each $\mathcal{C}_k$ through Least Squares
   - Local Parameter Vector (LPV) θ_k^L and associated variance V_k



Clustering-based procedure - Step 1

Extract local models

1. For each data point $(x(k), y(k))$ build a Local Dataset (LD) $\mathcal{C}_k$ collecting $(x(k), y(k))$ and its first $c-1$ neighboring points $(c > n)$
2. Fit a local linear model on each $\mathcal{C}_k$ through Least Squares
   - Local Parameter Vector (LPV) θ_k^L and associated variance V_k

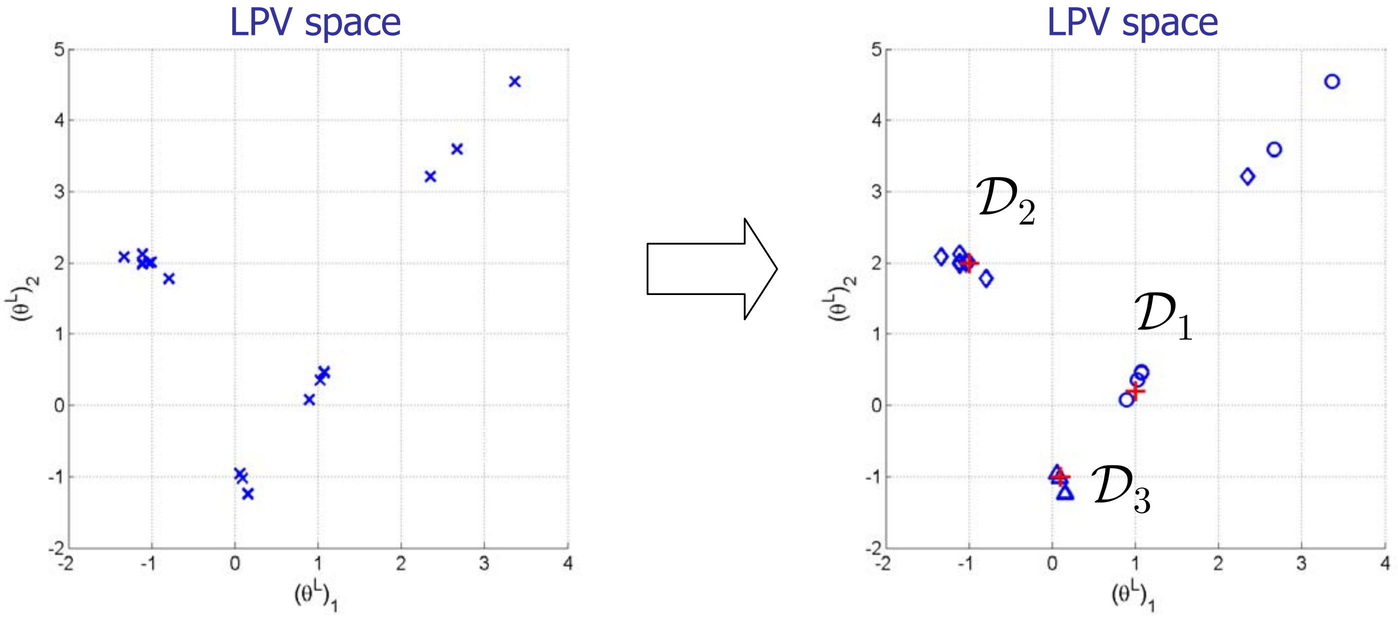


Clustering-based procedure - Step 2

Clustering of the LPVs

Find the clusters $\{\mathcal{D}_m\}_{m=1}^s$ and their centers $\{\mu_m\}_{m=1}^s$ that minimize

$$J = \sum_{m=1}^s \sum_{\theta_j^L \in \mathcal{D}_m} \|\theta_j^L - \mu_m\|_{V_j^{-1}}$$



Clustering-based procedure - Step 2

Clustering of the LPVs

Find the clusters $\{\mathcal{D}_m\}_{m=1}^s$ and their centers $\{\mu_m\}_{m=1}^s$ that minimize

$$J = \sum_{m=1}^s \sum_{\theta_j^L \in \mathcal{D}_m} \|\theta_j^L - \mu_m\|_{V_j^{-1}}$$

- *K-means strategy. Iterative procedure (fast but sub-optimal)*
  1. Fix the centers and compute the clusters
  2. Fix the clusters and compute the centers
  3. Go to 1 (if the cost decreased)

- If the centers are updated in a suitable way:
  - the cost decreases at each iteration
  - guaranteed termination in finitely many iterations
- Mixed LPVs have "high" variance V_k ⇨ Little influence on the final clusters
- K-means is a *supervised* clustering algorithm (the number of clusters must be specified)

Clustering-based procedure - Step 3

Classification of the data points

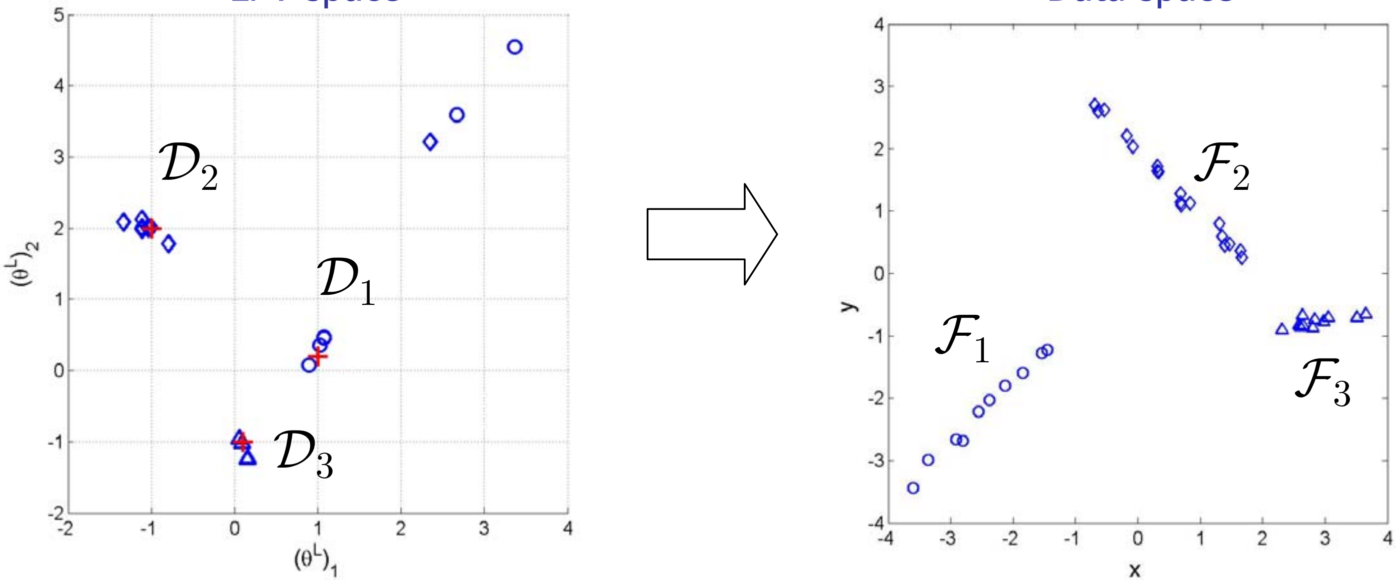
By construction we have the one-to-one map $\theta_k^L \leftrightarrow (x(k), y(k))$

Construct the mode data sets $\{\mathcal{F}_m\}_{m=1}^{s}$ *according to the rule:*

If $\theta_k^L \in \mathcal{D}_m$ then $\lambda(x(k)) = m$

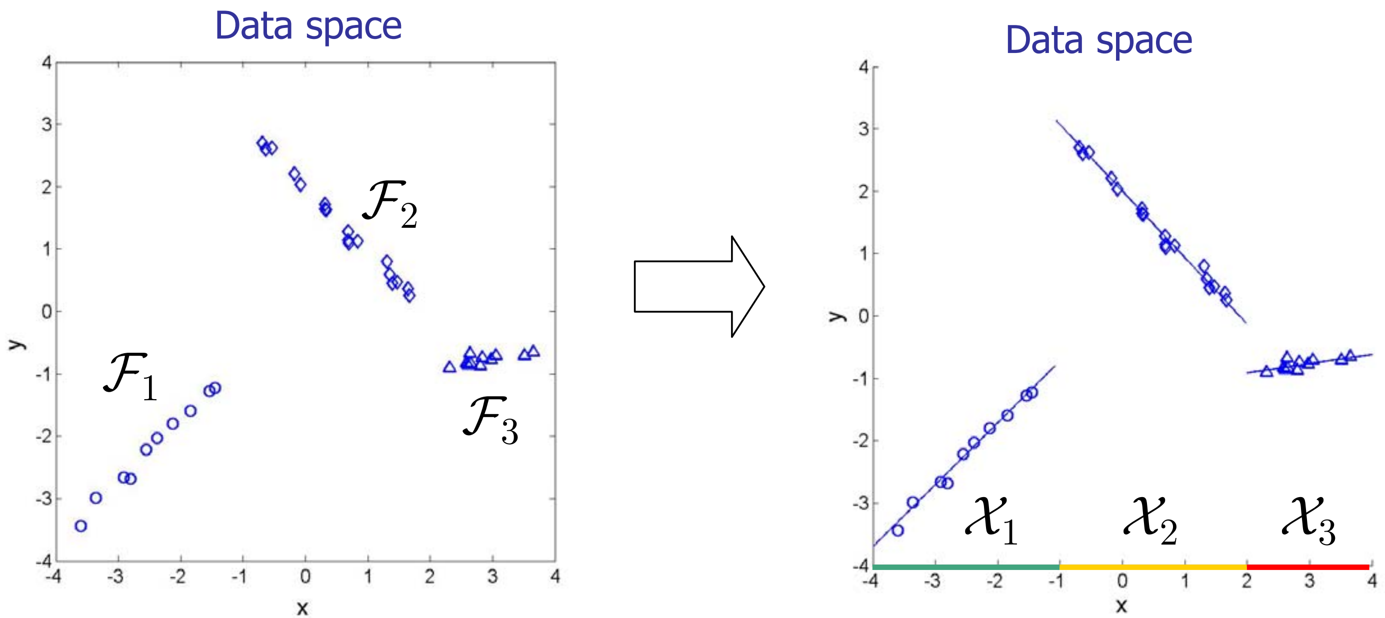
LPV space

Data space



Clustering-based procedure: final step

Easy step: find the modes (parameters and regions)



Computing mode PVs:

- Confidence measure on $(x(k), y(k))$: $w(k) = \frac{1}{\sqrt{(2\pi)^n \det(V_k)}}$
- Exploit $w(k)$ in weighted least squares (data points associated with mixed local models will have little impact on the estimates)

Clustering-based procedure: discussion

Parameters of the algorithm:

- Number of modes
- Size c of the LDs
  - c too big $\Rightarrow$ many mixed LDs
  - c too low $\Rightarrow$ poor noise averaging in computing the LPVs

Generalizations:

- The assumption $\theta_i \neq \theta_j,\ \ i \neq j$ can be removed by enhancing LPVs with measures of the spatial localization of LDs

(Ferrari-Trecate et *al.*, 2003)

- The number of modes can be automatically estimated by replacing K-means with an *unsupervised* clustering algorithm

(Ferrari-Trecate & Muselli, 2003)

  - estimation of the number of clusters (= number of modes)
  - warning: many unsupervised clustering algorithms depend on parameters that influence the number of clusters to be found !

The algebraic procedure

(Vidal et *al.*, 2003), (Ma & Vidal, 2005)

Algebraic procedure: introduction

Standing assumptions:

1) $\theta_i \neq \theta_j, \;\; i \neq j$

2) data are noiseless (will be relaxed at the end)

Key idea:

Recast the identification of a PWARX model into the identification of a "lifted" linear model whose coefficients can be computed without knowing the switching sequence.

Steps of the algorithm:

1) Find the number of modes

2) Compute the mode PVs

3) Classify the data points

The hybrid decoupling constraint

Noiseless data:

$$y(k) = \theta_i' \begin{bmatrix} x(k)' & 1 \end{bmatrix}' \text{ if } \lambda(x(k)) = i$$

Consider the extended PVs and regressor vectors:

- $\bar{\theta}_i = \begin{bmatrix} \theta_i' & 1 \end{bmatrix}'$ and $\bar{x}(k) = \begin{bmatrix} x(k)' & 1 & -y(k) \end{bmatrix}'$

Each $\bar{x}(k)$ verifies one of the equations

$$\bar{\theta}_i' \bar{x}(k) = \theta_i' \begin{bmatrix} x(k)' & 1 \end{bmatrix}' - y(k) = 0, \quad i \in \{1, \ldots, s\}$$

Then, $\forall \bar{x} \in \{\bar{x}(1), \ldots, \bar{x}(N)\}$ it holds

$$p(\bar{x}) = \prod_{i=1}^{s} \bar{\theta}_i' \bar{x} = 0$$

Hybrid decoupling constraint

The hybrid decoupling constraint

Hybrid decoupling constraint: $\begin{cases} p(\bar{x}) = \prod_{i=1}^{s} \bar{\theta}_i' \bar{x} = 0 \\ \forall \bar{x} \in \{\bar{x}(1), \ldots, \bar{x}(N)\} \end{cases}$

Example: $n = 2, \;\; s = 2$

[redacted]

- ν_s : Veronese map of degree s
  - monomials listed in the degree-lexicographic order

For the true mode number s, one has:

[redacted]

Data dependent

Unknowns

Estimation of the number of modes

For a generic mode number $m \in \mathbb{N}^+$ consider

$$L_m h_m = \begin{bmatrix} \nu_m(\bar{x}(1)) \\ \vdots \\ \nu_m(\bar{x}(N)) \end{bmatrix} h_m$$

Under mild assumptions on the data set, it holds:

$$s = \min \left\{ m : \text{rank}(L_m) = \binom{m+n+1}{m} - 1 \right\}$$

i.e. the system $L_m h_m = 0$ has a unique solution

⇩

Find h_s by solving $L_s h_s = 0$

Estimation of the PVs

Recall that $p(\bar{x})$ is the hybrid decoupling polynomial and $\bar{\theta}_i = \left[\begin{array}{cc} \theta_i' & 1 \end{array} \right]'$

- Let $Dp(\bar{x}) = \left[\begin{array}{ccc} \frac{\partial p(\bar{x})}{\partial \bar{x}_1} & \cdots & \frac{\partial p(\bar{x})}{\partial \bar{x}_{n+2}} \end{array} \right]'$
- It holds $\bar{\theta}_i = \frac{Dp(\bar{x})}{e' Dp(\bar{x})}, \; \forall \bar{x} : \lambda(x) = i$, where $e = \left[\begin{array}{cccc} 1 & 0 & \cdots & 0 \end{array} \right]'$

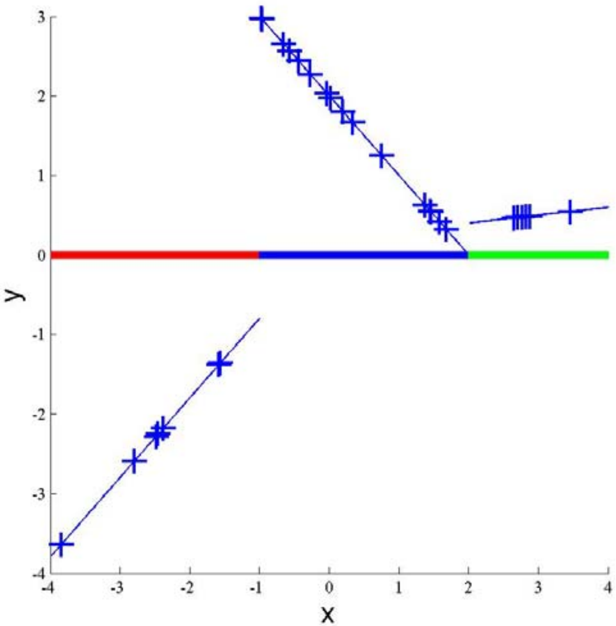
Problem: the switching function is unknown ...

... but for noiseless data:

$$\forall i \in \{1, \ldots, s\} \; \exists \bar{x}(k) : \lambda(x(k)) = i$$

⇩

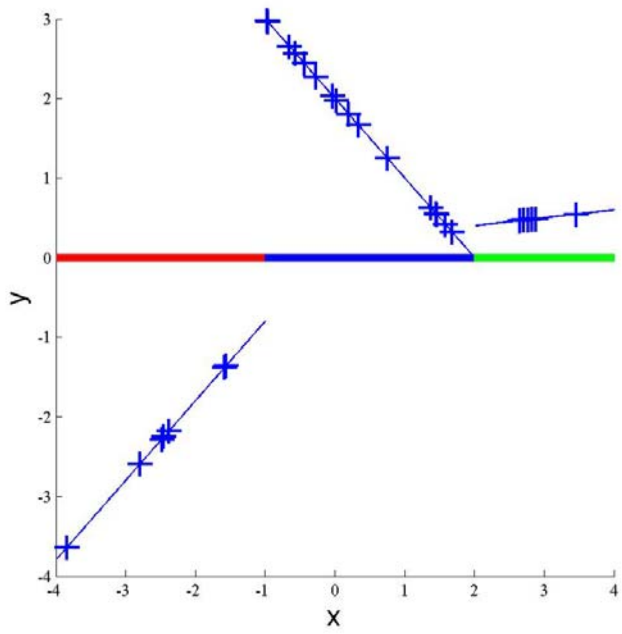
Compute $\bar{\theta}_i, \; i = 1, \ldots, s$ using the data points



Data classification

For $k = 1, \ldots, N$ set

$$\lambda(x(k)) = \arg \min_{i \in \{1,\ldots s\}} (y(k) - \theta_i' x(k))^2$$



Algebraic procedure: the noisy case

Noisy data: $y(k+1) = f(x(k)) + \eta(k)$

- Estimation of the number of modes

Rank-deficiency condition

$$s = \min\left\{m : \mathrm{rank}(L_m) = \binom{m+n+1}{m} - 1\right\}$$

- Problem: L_m is always full-rank, $\forall m \in \mathbb{N}^+$

Singular values

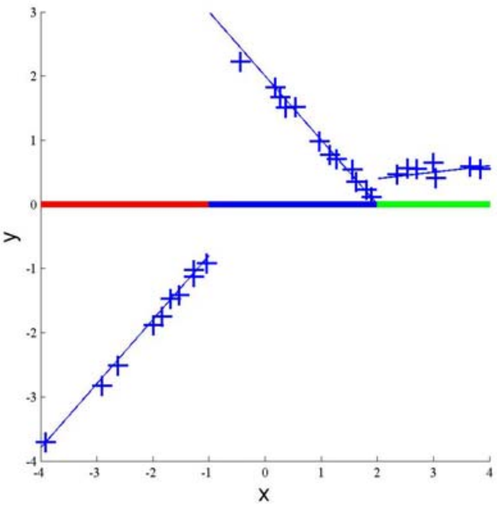
- Remedy: declare that $\mathrm{rank}(L_m) = r$ if $\sigma_{r+1}/\sigma_r < \epsilon$

User's knob

- Problem:
  - ϵ "big" $\Rightarrow$ few modes
  - ϵ "small" $\Rightarrow$ many modes

Algebraic procedure: the noisy case

Noisy data: $y(k+1) = f(x(k)) + \eta(k)$



- Estimation of the PVs

$$\bar{\theta}_i = \frac{Dp(\bar{x})}{e' Dp(\bar{x})}, \qquad \forall \bar{x} : \lambda(x) = i, \quad (1)$$

- Problem: no data point lies exactly on the mode hyperplanes

- Remedy: there are methods for finding the s data points closest to each hyperplane without knowing the switching sequence

(Ma & Vidal, 2005)

⇩

The rule (1) is still usable (but the quality of the estimates depends on the noise level)

Algebraic procedure: discussion

Parameters of the algorithm:

- No parameter in the noiseless case
- Tolerance ϵ in the noisy case
  - ϵ too big $\Rightarrow$ the number of modes is over-estimated
  - ϵ too small $\Rightarrow$ the number of modes is under-estimated

Generalizations:

- MIMO models can be considered (Vidal et *al.*, 2003)
- Automatic estimation of the model orders, possibly different for each mode of operation (Vidal, 2004)
- Recursive implementations for on-line identification (Vidal & Anderson, 2004), (Hashambhoy & Vidal, 2005)

A discussion on the assumption

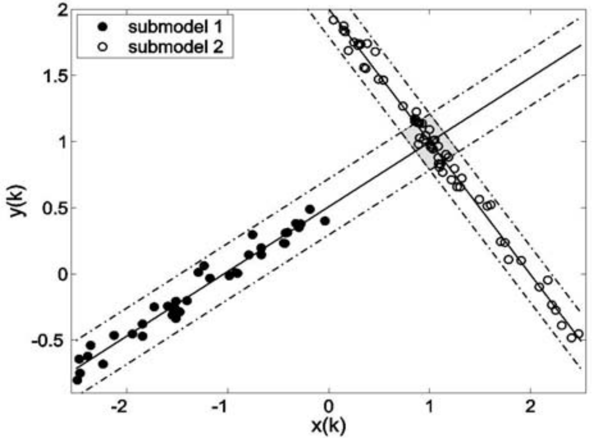
$$\theta_i \neq \theta_j, \;\; i \neq j$$

Modes of operation with virtual intersections

The assumption $\theta_i \neq \theta_j, \;\; i \neq j$ is critical in two cases:

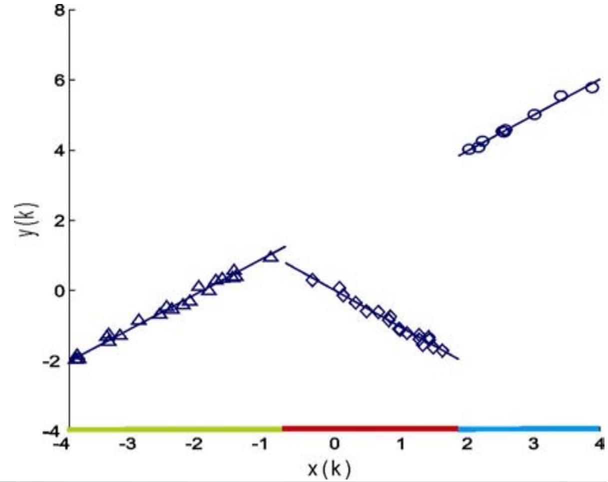
1) The hyperplanes defined by the PVs θ_i and θ_j intersect over $\mathcal{X}_i$

- They may fit equally well data close to the intersection
- These data points may be wrongly classified



2) *Same* PVs for *different* modes i.e. $\theta_i = \theta_j, \;\; i \neq j$

- Data belonging to different modes will be classified in the same way



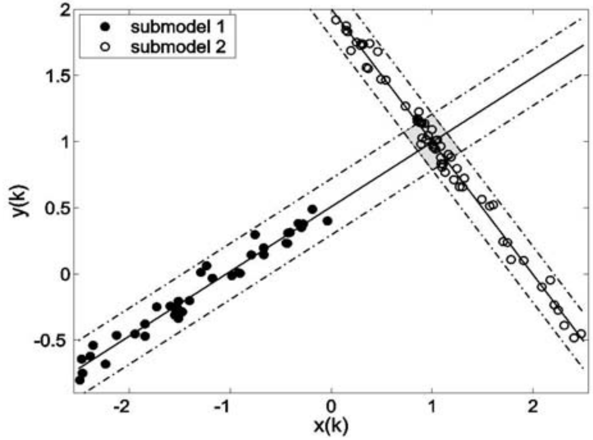
Modes of operation with virtual intersections

Consequences on region estimation: consider the sets

$$X_i = \{x(k) : \lambda(x(k)) = i\} \quad X_j = \{x(k) : \lambda(x(k)) = j\}$$

1) The hyperplanes defined by the PVs θ_i and θ_j intersect over $\mathcal{X}_i$

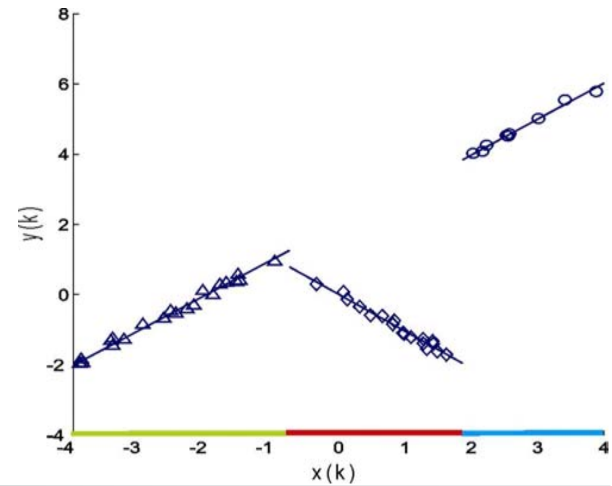
- Wrongly classified data points make the sets X_i and X_j linearly inseparable



2) *Same* PVs for *different* modes, i.e. $\theta_i = \theta_j,\ i \neq j$

- It may happen that no point in X_i is linearly separable from all points in X_j

The quality of the reconstructed regions may be extremely poor



The bounded-error procedure

(Bemporad et *al.*, 2003)

Bounded-error procedure: introduction

Standing assumption: $\theta_i \neq \theta_j, \; i \neq j$

Key idea:

Impose that all prediction errors are bounded by a given quantity $\delta > 0$
This allows one to recast the identification problem into the problem of finding the MINimum Partition into Feasible Subsystems (MIN-PFS) of a set of inequalities

Steps of the algorithm:

1) Initialization: solve the MIN-PFS problem and get a first estimate of the number of modes, PVs, and switching sequence
2) Refinement: final classification of the data points

Bounded-error condition

Impose that the prediction errors are bounded by a given quantity $\delta > 0$

The identified model must verify the
Linear Complementarity Inequalities (LCIs)

$$|y(k) - f(x(k))| \leq \delta, \quad \forall k \in \{1, \ldots, N\}$$

Each LCI can be split into two linear inequalities:

$$f(x(k)) \leq y(k) + \delta$$
$$f(x(k)) \geq y(k) - \delta$$

Role of δ : trade off between model *accuracy* and *complexity*

The MIN-PFS problem

Identification problem restated as MINimum Partition into Feasible Subsytem (MIN-PFS) problem

Given $\delta > 0$ and the (possibly infeasible) system of N LCIs

$$\begin{cases} |y(1) - \theta' \begin{bmatrix} x(1)' & 1 \end{bmatrix}'| \leq \delta \\ \vdots \\ |y(N) - \theta' \begin{bmatrix} x(N)' & 1 \end{bmatrix}'| \leq \delta \end{cases}$$

find a partition into a minimum number s of feasible subsystems of LCIs

$$\left\{ |y(i) - \theta_1' \begin{bmatrix} x(i)' & 1 \end{bmatrix}'| \leq \delta, \ i \in \mathcal{I}_1 \right\}$$

$$\vdots$$

$$\left\{ |y(i) - \theta_s' \begin{bmatrix} x(i)' & 1 \end{bmatrix}'| \leq \delta, \ i \in \mathcal{I}_s \right\}$$

$$\mathcal{I}_i \cap \mathcal{I}_j = \emptyset, \ \ i \neq j$$

$$\cup_{i=1}^{s} \mathcal{I}_i = \{1, \ldots, N\}$$

The MIN-PFS problem is NP hard

A suboptimal greedy algorithm was proposed in (Amaldi & Mattavelli, 2002)

Greedy algorithm for MIN-PFS problems

Set $\mathcal{I} = \{1, \ldots, N\}$ and $m = 1$

1) Choose θ that verifies the largest number of LCIs

$$|y(k) - \theta' \left[\begin{array}{cc} x'(k) & 1 \end{array} \right]'| \leq \delta, \quad k \in \mathcal{I}$$

MAXimum Feasible Subsystem (MAX FS) problem

Let $\mathcal{I}_m = \{k : |y(k) - \theta' \left[\begin{array}{cc} x'(k) & 1 \end{array} \right]'| \leq \delta\}$

2) Set $\theta_m = \theta, \ \lambda(x(k)) = m \Leftrightarrow k \in \mathcal{I}_m$

3) Set $\mathcal{I} = \mathcal{I} \backslash \mathcal{I}_m, \ m = m + 1$ and go to (1) if $\mathcal{I} \neq \emptyset$

Output: mode number m, switching sequence and PVs

The MAX FS problem is still NP hard

- A sub-optimal but computational efficient algorithm to solve it using a *randomized* method has been given in (Amaldi & Mattavelli, 2002)

Pitfalls of the greedy algorithm

Problems:

- The greedy algorithm is not guaranteed to yield a minimal partition (causes: sub-optimality and randomness)
- The mean number of extracted subsystems may be well above the minimum

In order to cope with these drawbacks, modifications to the original algorithms have been proposed (Bemporad et *al., 2003-2004-2005*)

- Still, the estimates of the number of modes and the switching sequence need improvements

↓

Refinement of the estimates

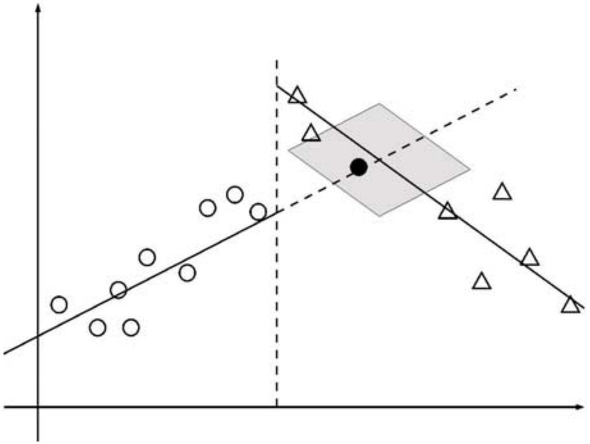
Virtual intersections

How to cope with virtual intersections ?

The hyperplanes defined by the PVs θ_j and $\theta_i, \ \theta_j \neq \theta_i$ intersect over $\mathcal{X}_i$

Ideas:

- Classify as *undecidable* points that are consistent with more than one mode
- Use nearest neighbors rules for attributing undecidable data points to modes and reduce misclassification errors



Refinement of the estimates

Input: parameters $\theta_i^{(0)}$ from the initialization step

Set $t = 0$ (iteration counter)

1. Data classification. For each point $(x(k), y(k)),\ \ k = 1, \ldots, N$
   - If $|y(k) - \theta_i^{(t)'} \begin{bmatrix} x(k)' & 1 \end{bmatrix}'| \leq \delta$ for only one i set $\lambda(x(k)) = i$
   - If $|y(k) - \theta_i^{(t)'} \begin{bmatrix} x(k)' & 1 \end{bmatrix}'| > \delta$ for all i mark the point as **infeasible**
   - Otherwise mark the point as **undecidable**
2. Assignment of undecidable points (nearest neighbors rules)
3. Update the PVs obtaining $\theta_i^{(t+1)}$
4. Iterate until $||\theta_i^{(t+1)} - \theta_i^{(t)}||\backslash||\theta_i^{(t)}|| < \gamma$

given termination threshold

Other improvements

Reduce the number of submodels by

1. aggregating models with similar PVs
2. discarding modes of operation with few data points (they are likely to be artifacts caused by the greedy algorithm)

Bounded-error procedure: discussion

Parameters of the algorithm:

- error bound
- thresholds for taking decisions
  - when to merge two modes
  - when to stop the refinement
- parameters influencing the behavior of the randomized algorithm for the MIN-PFS problem (not critical to set in many practical cases)

Other applications:

- Useful when the noise corrupting the measurements is bounded (and the bound is known)
- Useful for obtaining PWA approximations of a nonlinear function with a *given* accuracy

Back to the motivating example

Placement of the electronic component on the Printed Circuit Board (PCB)

Experimental setup

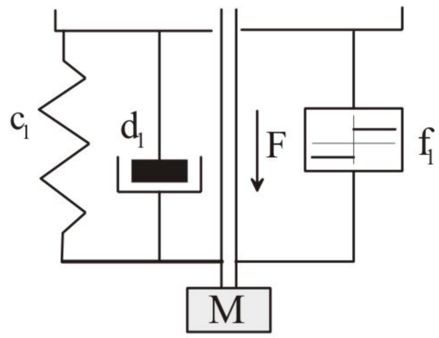
Mounting head

Moving impacting surface

Ground connection

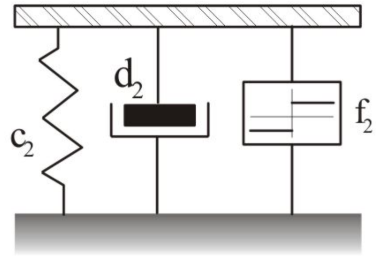
Conceptual representation

Mounting head (M) connected to the casing



F: Motor force (input)
d: Linear friction
c: Spring
f: Dry friction
Output: Head position
(Upper saturation=0)

Impact surface connected to the ground



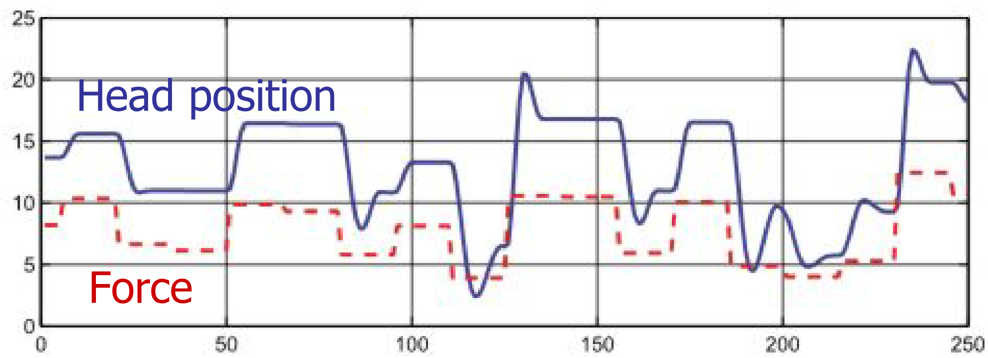
PWA system with 4 modes:

- upper saturation
- free mode
- impact mode
- lower saturation

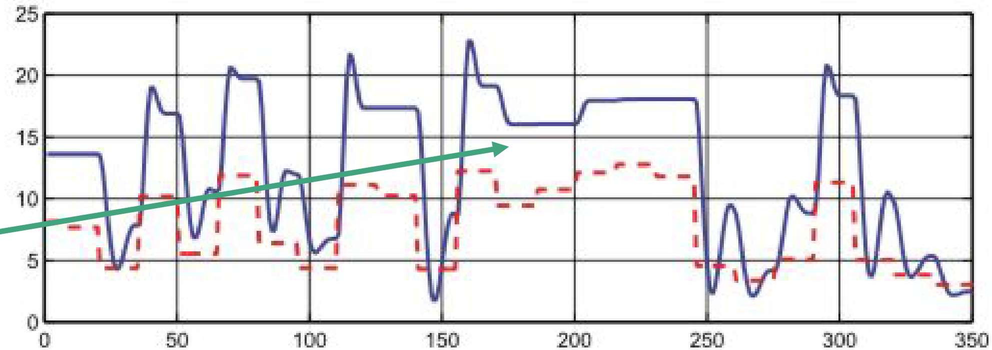
Bimodal learning experiment

- We focus on two modes: free and impact
- Saturations are avoided through a suitable input signal

Identification data (250 points)



Validation data



Effect of the dry friction

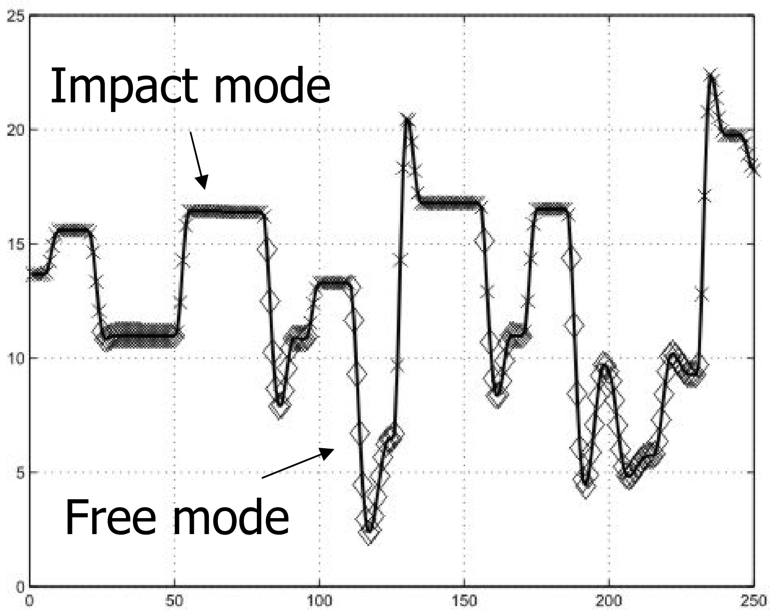
Clustering-based procedure: results

PWARX model with two modes: (Juloski et *al.*, 2004)

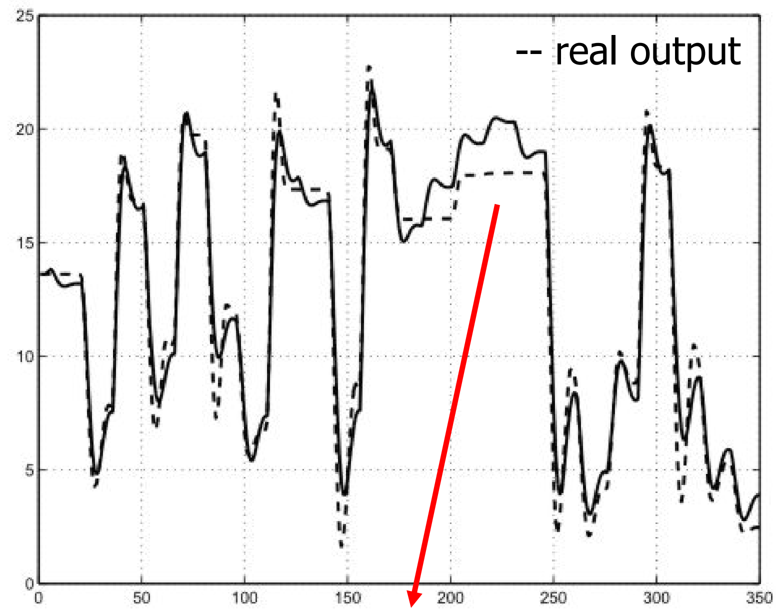
Regressors: $x(k) = [y(k-1) \quad y(k-2) \quad u(k-1)]'$

Size of the LDs: $c = 55$

Classified data points



Validation results *(simulation !)*



The "small" nonlinearity due to the dry friction is averaged out

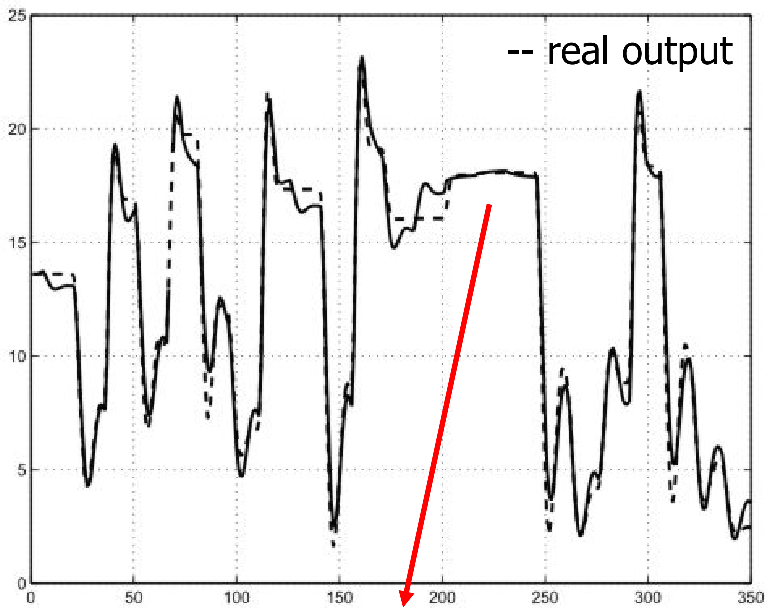
Clustering-based procedure: results

PWARX model with three modes: (Juloski et *al.*, 2004)

Regressors: $x(k) = [y(k-1) \quad y(k-2) \quad u(k-1) \quad u(k-2)]'$

Size of the LDs: $c = 35$

Validation results *(simulation!)*



The dry friction effect is captured by the new mode

Conclusions

Identification of PWARX models: the overall complex behavior decomposed in simple modes of operation

Main challenge of hybrid identification: the classification problem

- Three algorithms have been discussed
  - Detailed comparison in (Juloski et *al.* , HSCC05, 2005)

- Other algorithms for hybrid identification are available !

Tutorial paper on hybrid identification:

S. Paoletti, A.Lj. Juloski, G. Ferrari-Trecate, R. Vidal (2007). Identification of hybrid systems: a tutorial. *European Journal of Control,* **13**, 242–260.

Other methods for hybrid identification

(incomplete list ...)

PWARX models

• Optimization-based approaches: the classification problem is recast into an optimization problem

- e.g. Hinging-hyperplane model ⇒ Mixed-Integer programming (Roll et *al.*, 2005)

• Bayesian procedures: a priori knowledge on PVs is embodied in prior probabilities (Juloski et *al.*, 2005)

- Classification and PV estimation are based on Bayes rule
- By-product: confidence measures on the correct classification of data points that can used to improve the estimation of the regions

Switched ARX models

• Account for stochasticity in switching between modes (Verriest, 2001)

Other methods for hybrid identification

(incomplete list ...)

PWA models in state-space form

Tools: subspace identification techniques

(Borges et *al.*, 2006), (Verdult and Verhaegen, 2004)

GNU-licensed MatLab software

- HIT toolbox http://sisdin.unipv.it/lab/personale/pers_hp/ferrari/HIT_toolbox.html
  - Clustering-based procedures

HIT Hybrid Identification Toolbox

- PWAID toolbox http://www.rt.isy.liu.se/~roll/PWAID/
  - Identification of hinging-hyperplane models
  - Bounded-error procedure