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:

\[
\text{If } \theta_k^L \in \mathcal{D}_m \text{ then } \lambda(x(k)) = m
\]
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) \( \theta_i \neq \theta_j, \quad i \neq j \)
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

\[ p(\bar{x}) = \prod_{i=1}^{s} \bar{\theta}_i \bar{x} = 0 \quad \forall \bar{x} \in \{x(1), \ldots, x(N)\} \]

Example: \( n = 2, \ s = 2 \)

\[ p(\bar{x}) = (\theta_{11} \bar{x}_1 + \bar{\theta}_{12} \bar{x}_2 + \bar{x}_3)(\bar{\theta}_{21} \bar{x}_1 + \bar{\theta}_{22} \bar{x}_2 + \bar{x}_3) = h'_s \nu_s(\bar{x}) \]

\[ h_s = \begin{bmatrix} \bar{\theta}_{11} \bar{\theta}_{21} & \bar{\theta}_{11} \bar{\theta}_{22} + \bar{\theta}_{12} \bar{\theta}_{21} & \bar{\theta}_{11} + \bar{\theta}_{21} & \bar{\theta}_{12} \bar{\theta}_{22} & \bar{\theta}_{12} + \bar{\theta}_{22} & 1 \end{bmatrix}' \]

\[ \nu_s(\bar{x}) = \begin{bmatrix} \bar{x}_1^2 & \bar{x}_1 \bar{x}_2 & \bar{x}_1 \bar{x}_3 & \bar{x}_2^2 & \bar{x}_2 \bar{x}_3 & \bar{x}_3^2 \end{bmatrix}' \]

- \( \nu_s \): Veronese map of degree \( s \)
- monomials listed in the degree-lexicographic order

For the true mode number \( s \), one has:

\[ \nu_s(\bar{x})' h_s = 0 \Rightarrow L_s h_s = \begin{bmatrix} \nu_s(\bar{x}(1))' \\ \vdots \\ \nu_s(\bar{x}(N))' \end{bmatrix} h_s = 0 \]

Data dependent

Unknows
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:

\[
s = \min \left\{ m : \text{rank}(L_m) = \left( \frac{m+n+1}{m} \right) - 1 \right\}
\]

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 \( \vec{\theta}_i = \left[ \begin{array}{c} \theta'_i \\ 1 \end{array} \right]' \)

- Let \( Dp(\bar{x}) = \left[ \frac{\partial p(\bar{x})}{\partial x_1} \quad \ldots \quad \frac{\partial p(\bar{x})}{\partial x_{n+2}} \right]' \)

- It holds \( \vec{\theta}_i = \frac{Dp(\bar{x})}{e'Dp(\bar{x})}, \, \forall \bar{x} : \lambda(x) = i \), where \( e = \left[ \begin{array}{c} 1 \\ 0 \quad \ldots \quad 0 \end{array} \right]' \)

**Problem:** the switching function is unknown ...

... but for noiseless data:

\[ \forall i \in \{1, \ldots, s\} \quad \exists \bar{x}(k) : \lambda(x(k)) = i \]

\[ \downarrow \]

Compute \( \vec{\theta}_i, \, i = 1, \ldots, s \) using the data points
Data classification

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

$$\lambda(x(k)) = \arg \min_{i \in \{1, \ldots, s\}} (y(k) - \theta_i^T x(k))^2$$
Algebraic procedure: the noisy case

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

- Estimation of the number of modes

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

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

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

  - Problem:
    - $\epsilon$ “big” $\Rightarrow$ few modes
    - $\epsilon$ “small” $\Rightarrow$ many modes
Algebraic procedure: the noisy case

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

- Estimation of the PVs
  \[
  \bar{\theta}_i = \frac{Dp(\bar{x})}{e' Dp(\bar{x})}, \quad \forall \bar{x} : \lambda(x) = i, \quad (1)
  \]

- 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 $\varepsilon$ in the noisy case
  - $\varepsilon$ too big $\Rightarrow$ the number of modes is over-estimated
  - $\varepsilon$ 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 $\theta_i$ and $\theta_j$ intersect over $X_i$
   - 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 \( \theta_i \) and \( \theta_j \) intersect over \( X_i \)
   - Wrongly classified data points make the sets \( X_i \) and \( X_j \) linearly inseparable

2) \textit{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 $\delta > 0$

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

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

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 $\delta$: 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

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

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

$$\begin{cases} &|y(i) - \theta_1'[x(i)' 1]'| \leq \delta, \ i \in \mathcal{I}_1 \\ &\vdots \\ &|y(i) - \theta_s'[x(i)' 1]'| \leq \delta, \ i \in \mathcal{I}_s \end{cases}$$

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

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

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 $\theta$ that verifies the largest number of LCIs

$$|y(k) - \theta' \begin{bmatrix} x'(k) & 1 \end{bmatrix}' | \leq \delta, \quad k \in \mathcal{I}$$

MAXimum Feasible Subsystem (MAX FS) problem

Let $\mathcal{I}_m = \{k : |y(k) - \theta' \begin{bmatrix} x'(k) & 1 \end{bmatrix}' | \leq \delta\}$

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

3) Set $\mathcal{I} = \mathcal{I} \setminus \mathcal{I}_m$, $m = m + 1$ 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 \textit{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 \( \theta_j \) 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 $\|\theta_i^{(t+1)} - \theta_i^{(t)}\| \|\theta_i^{(t)}\| < \gamma$
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

Impact surface connected to the ground

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

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
Clustering-based procedure: results

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

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

Size of the LDs: \( c = 55 \)

Classified data points

Impact mode

Free mode

Validation results \textit{(simulation !)}

The “small” nonlinearity due to the dry friction is averaged out
Clustering-based procedure: results

PWARX model with three modes:

Regressors: \( x(k) = \begin{bmatrix} y(k-1) & y(k-2) & u(k-1) & u(k-2) \end{bmatrix} \)

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:

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:** subpace 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](http://sisdin.unipv.it/lab/personale/pers_hp/ferrari/HIT_toolbox.html)
  - Clustering-based procedures

  - Identification of hinging-hyperplane models
  - Bounded-error procedure