Identification of Hybrid Systems

Alberto Bemporad

Dip. di Ingegneria dell'Informazione
Università degli Studi di Siena

bemporad@dii.unisi.it
http://www.dii.unisi.it/~bemporad

Università degli Studi di Siena
Facoltà di Ingegneria

Goal

- Sometimes a *hybrid model* of the process (or of a part of it) cannot be derived manually from available knowledge.
- Therefore, a model must be either
  - Estimated from data (model unknown)
  - or *hybridized* before it can be used for control/analysis (model known but nonlinear)
- If a linear model is enough, no problem: several algorithms are available (e.g.: use Ljung's ID TBX)
- If switching modes are known and data can be generated for each mode, no problem: we identify one linear model per mode (e.g.: use Ljung's ID TBX)
- If modes & dynamics must be identified together, we need

hybrid system identification

PWARX Models

Consider PieceWise Affine autoRegressive eXogenous (PWARX) models of the form

$$y_k = \theta_i' \underbrace{\begin{bmatrix} x_k \\ 1 \end{bmatrix}}_{\varphi_k} + e_k \quad \text{if } x_k \in \mathcal{X}_i \text{ for some } i = 1, \ldots, s$$

where:

- $y_k \in \mathbb{R}$ is the *system output*
- $x_k \in \mathbb{R}^n$ is the *regression vector*,

  e.g. $x_k = [y_{k-1} \ldots y_{k-n_a}\, u_{k-1} \ldots u_{k-n_b}]'$
- $e_k \in \mathbb{R}$ is the *error*
- $\{\mathcal{X}_i\}_{i=1}^s$, $\mathcal{X}_i = \{x : \ H_i x \leq 0\}$, is a *polyhedral partition* of the regressor set $\mathcal{X} \subseteq \mathbb{R}^n$

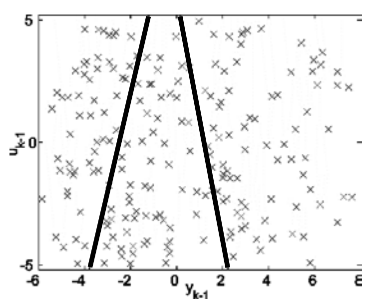
unknowns: $\{H_i,\ \theta_i,\ s\},\ i = 1, \ldots, s$

PWA Identification Problem

Estimate from data **both** the parameters of the affine submodels **and** the partition of the PWA map

Example Let the data be generated by the PWARX system

$$y_k = \begin{cases} \begin{bmatrix} -0.4 & 1 & 1.5 \end{bmatrix} \varphi_k + \varepsilon_k \\ \quad \text{if } \begin{bmatrix} 4 & -1 & 10 \end{bmatrix} \varphi_k < 0 \\[1ex] \begin{bmatrix} 0.5 & -1 & -0.5 \end{bmatrix} \varphi_k + \varepsilon_k \\ \quad \text{if } \begin{bmatrix} -4 & 1 & -10 \\ 5 & 1 & -6 \end{bmatrix} \varphi_k \leq 0 \\[1ex] \begin{bmatrix} -0.3 & 0.5 & -1.7 \end{bmatrix} \varphi_k + \varepsilon_k \\ \quad \text{if } \begin{bmatrix} -5 & -1 & 6 \end{bmatrix} \varphi_k < 0 \end{cases}$$



with $\varphi_k = [y_{k-1}\ u_{k-1}\ 1]'$, $|u_k| \leq 5$ and $|\varepsilon_k| \leq 0.1$

PWA Identification Problem

$$\min_{\theta_j, H_j} \frac{1}{2N} \sum_{t=1}^{N} \left(\sum_{j=1}^{s} \|y_t - \varphi_t' \theta_j\| \mathcal{J}_j(\varphi_t) \right)$$

$$\text{subj. to } \mathcal{J}_j(\varphi_t) = \begin{cases} 1 & \text{if } H_j \varphi_t \leq 0 \\ 0 & \text{otherwise} \end{cases}$$

$+$ linear bounds over θ_j, H_j

$\|v\|_2^2$	$=$	$v'v$	Euclidean norm
$\|v\|_\infty$	$=$	$\max \|v_i\|$	∞-norm
$\|v\|_1$	$=$	$\sum \|v_i\|$	1-norm

A. Known Guardlines (partition H_j known, θ_j unknown): ordinary least-squares problem (or linear/quadratic program if linear bounds over θ_j are given) **EASY PROBLEM**

B. Unknown Guardlines (partition H_j and θ_j unknown): generally non-convex, local minima **HARD PROBLEM!**

Approaches to PWA Identification

- Mixed-integer linear or quadratic programming

  J. Roll, A. Bemporad and L. Ljung, "*Identification of hybrid systems via mixed-integer programming*", Automatica, to appear.

- Bounded error & partition of infeasible set of inequalities

  A. Bemporad, A. Garulli, S. Paoletti and A. Vicino, "*A Greedy Approach to Identification of Piecewise Affine Models*", HSCC'03

- K-means clustering in a feature space

  G. Ferrari-Trecate, M. Muselli, D. Liberati, and M. Morari, "*A clustering technique for the identification of piecewise affine systems*", Automatica, 2003

- Other approaches:
  - Polynomial factorization (algebraic approach) (R. Vidal, S. Soatto, S. Sastry, 2003)
  - Hyperplane clustering in data space (E. Münz, V. Krebs, *IFAC* 2002)

Mixed-Integer Approach

Mixed-Integer Approach

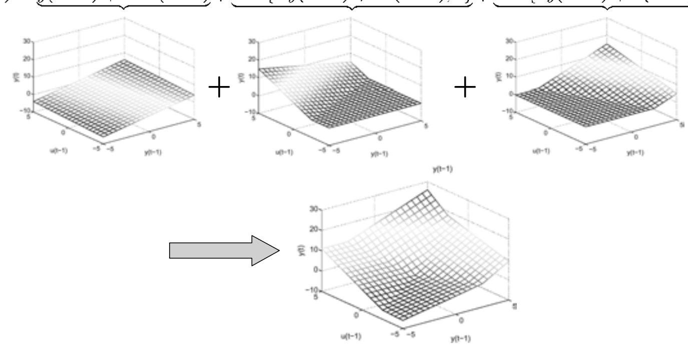
$$y_t = \varphi_t' \theta_0 + \sum_{i=1}^{S} \pm \max\{\varphi_t' \theta_i, 0\} + e_t$$

Hinging Hyperplane hybrid models

(Breiman, 1993)

Example:

$$y(t) = \underbrace{y(t-1) + 0.2u(t-1)}_{} + \underbrace{\max\{-y(t-1) + 2u(t-1), 0\}}_{} + \underbrace{\max\{2y(t-1) + u(t-1), 0\}}_{}$$



Mixed-Integer Approach

$$\hat{y}(t|\theta) = \varphi'(t)\theta_0 + \sum_{i=1}^{S} \pm \max\{\varphi'(t)\theta_i, 0\}$$

one-step ahead predicted output (t=0, 1, ..., N-1)

optimization problem: $$\min_{\theta} \frac{1}{2N} \sum_{t=0}^{N-1} [y_t - \hat{y}(t|\theta)]^2$$

- Could be solved using numerical methods such as the Gauss-Newton method. (Breiman, 1993)
- Problem: Local minima.
- We want to find a method that finds the global minimum

Mixed-Integer Approach

- A general Mixed-Integer Quadratic Program (MIQP) can be written as

$$\begin{aligned} \min_{x,\delta} \quad & \begin{bmatrix} x' & \delta' \end{bmatrix} Q \begin{bmatrix} x \\ \delta \end{bmatrix} + p' \begin{bmatrix} x \\ \delta \end{bmatrix} \\ \text{s.t.} \quad & C \begin{bmatrix} x \\ \delta \end{bmatrix} \leq d \\ & \delta \in \{0,1\}^m \qquad (x \in \mathbb{R}^n) \end{aligned}$$

(if Q=0 the problem is an MILP)

1. If we set $z_i(t) = \max\{\varphi'(t)\theta_i, 0\}$, we get

$$\hat{y}(t|\theta) = \varphi'(t)\theta_0 + \sum_{i=1}^{S} \pm z_i(t)$$

the cost function becomes quadratic in $(\theta_i, z_i(t))$:

$$\sum_{t=0}^{N-1} [y_t - \hat{y}(t|\theta)]^2 = \sum_{t=0}^{N-1} [y_t - \varphi'(t)\theta_0 - \sum_{i=1}^{S} \pm z_i(t)]^2$$

Mixed-Integer Approach

2. Introduce binary variables $\delta_i(t) = \begin{cases} 1 & \text{if } \varphi'(t)\theta_i \geq 0 \\ 0 & \text{otherwise} \end{cases}$

$$z_i(t) = \max\{\phi'(t)\theta_i, 0\} = \varphi'(t)'\theta_i\delta_i(t) \qquad \text{(if-then-else condition)}$$

3. Get linear mixed-integer constraints:

$$\begin{aligned} & \varphi'(t)\theta_i \leq M\delta_i(t) \\ & \varphi'(t)\theta_i \geq \varepsilon + (m - \varepsilon)(1 - \delta_i(t)) \\ \\ & -M\delta_i(t) + z_i(t) \leq 0 \\ & m\delta_i(t) - z_i(t) \leq 0 \\ & -M(1 - \delta_i(t)) - z_i(t) \leq -\varphi(t)'\theta_i \\ & m(1 - \delta_i(t)) + z_i(t) \leq \varphi(t)'\theta_i \end{aligned}$$

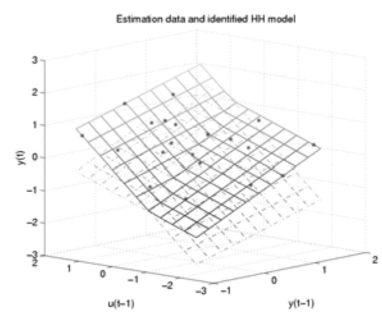
- ε is a small positive scalar (e.g., the machine precision),
- M and m are upper and lower bounds on $\varphi'(t)\theta_i$ (from bounds on θ_i)

The identification problem is an MIQP !

Mixed-Integer Approach

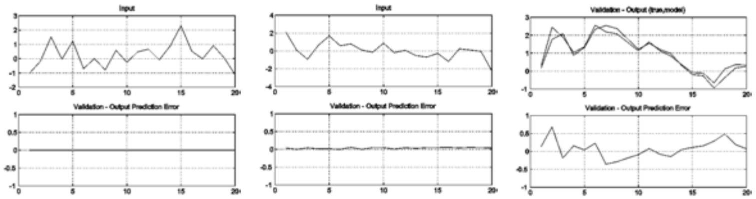
Example: Identify the following system

$$y_t = 0.8y_{t-1} + 0.4u_{t-1} - 0.1 + \max\{-0.3y_{t-1} + 0.6u_{t-1} + 0.3, 0\}$$



MILP: 66 variables (of which 20 integers) and 168 constraints.
Problem solved using Cplex 6.5 (1014 LP solved in 0.68 s)

Mixed-Integer Approach



System identified using noiseless data

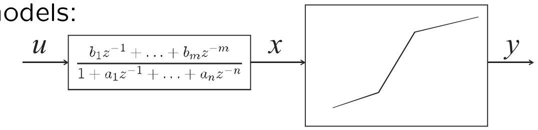
Using data with Var($e(t)$)=0.1

Fitting an ARX model to same data (Var($e(t)$)=0.1)

Problem: Worst-case complexity is exponential in the number of hinge functions and in the number of data.

Mixed-Integer Approach

Wiener models:



- Linear system $G(z)$ followed by a one dimensional static nonlinearity f.
- Assumptions: f is piecewise affine, continuous, invertible $\Rightarrow$ the system is piecewise affine.

Result:

- The identification problem can be again solved via MIQP or MILP
- Complexity is polynomial in worst-case in the number of data and number of max function
- Still the complexity depends heavily on the number of data

Mixed-Integer Approach

Comments:

- Global optimal solution can be obtained
- A 1-norm objective function gives an MILP problem a 2-norm objective function gives an MIQP problem
- Worst-case performance is exponential in the number functions and quite bad in the number of data!

Need to find methods that are suboptimal but computationally more efficient !

Bounded-Error Approach

Bounded Error Condition

Consider again a PWARX model of the form

$$\begin{aligned} y_k &= f(x_k) + e(k) \\ f(x_k) &= \theta_i' \underbrace{\begin{bmatrix} x_k \\ 1 \end{bmatrix}}_{\varphi_k} \quad \text{if } x_k \in \mathcal{X}_i \text{ for some } i = 1, \ldots, s \end{aligned}$$

Bounded-error: select a bound δ>0 and require that the identified model satisfies the condition

$$|y_k - f(x_k)| \le \delta, \quad \forall k = 1, \ldots, N$$

Role of δ: trade off between *quality of fit* and *model complexity*

Problem Given N datapoints (y_k,x_k), k=1, ..., N, estimate the min integer s, a partition $\mathcal{X}_1, \ldots, \mathcal{X}_s$, and params θ_1, ... ,θ_s such that the corresponding PWA model satisfies the bounded error condition

MIN PFS Problem

Problem restated as a MIN PFS problem:
(MINimum Partition into Feasible Subsystems)

Given δ>0 and the (possibly infeasible) system of N linear complementary inequalities

$$\left| y_k - \varphi_k' \theta \right| \le \delta, \quad k = 1, \ldots, N,$$

find a partition of this system of inequalities into a minimum number s of feasible subsystems of inequalities

- The partition of the complementary ineqs provides data classification (=clusters)
- Each subsystem of ineqs defines the set of linear models θ_i that are compatible with the data in cluster $\#i$
- MIN PFS is an NP-hard problem

A Greedy Algorithm for MIN PFS

A. Starting from an infeasible set of inequalities, choose a parameter θ that satisfies the largest number of ineqs

$$\left| y_k - \varphi_k' \theta \right| \le \delta, \quad k = 1, \ldots, N,$$

and classify those satisfied ineqs as the first cluster *(MAXimum Feasible Subsystem,* **MAX FS***)*

B. Iteratively repeat the MAX FS problem on the remaining ineqs

- The MAX FS problem is still NP-hard
- Amaldi & Mattavelli propose to tackle it using a randomized and thermal relaxation method

(Amaldi & Mattavelli, Disc. Appl. Math., 2002)

PWA Identification Algorithm

1. *Initialize*: exploit a greedy strategy for partitioning an infeasible system of linear inequalities into a minimum number of feasible subsystems
2. *Refine the estimates:* alternate between datapoint reassignment and parameter update
3. *Reduce the number of submodels*:
   a. join clusters whose model θ_i is similar, or
   b. remove clusters that contain too few points
3. *Estimate the partition*: compute a separating hyperplane for each pair of clusters of regression vectors (alternative: use multi-category classification techniques)

Step #1: Greedy Algorithm for MIN-PFS

Comments on the greedy algorithm

- The greedy strategy is not guaranteed to yield a minimum number of partitions (it solves MIN PFS only suboptimally)
- Randomness involved for tackling the MAX FS problem
- The cardinality and the composition of the clusters may depend on the order in which the feasible subsystems are extracted
- Some datapoints might be consistent with more than one submodel

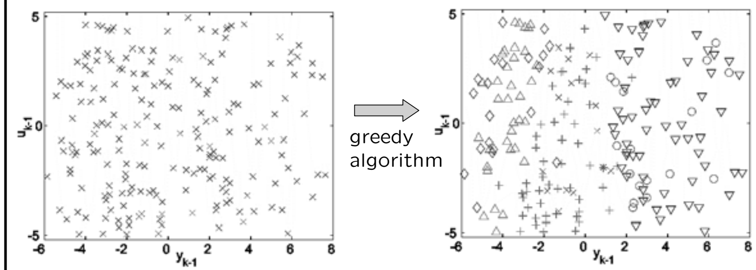
The greedy strategy can only be used for **initialization** of the clusters. Then we need a procedure for the **refinement** of the estimates

Example (cont'd)

Consider again the PWARX system

$$y_k = \begin{cases} \begin{bmatrix} -0.4 & 1 & 1.5 \end{bmatrix} \varphi_k + \varepsilon_k & \text{if } \begin{bmatrix} 4 & -1 & 10 \end{bmatrix} \varphi_k < 0 \\ \begin{bmatrix} 0.5 & -1 & -0.5 \end{bmatrix} \varphi_k + \varepsilon_k & \text{if } \begin{bmatrix} -4 & 1 & -10 \\ 5 & 1 & -6 \end{bmatrix} \varphi_k \leq 0 \\ \begin{bmatrix} -0.3 & 0.5 & -1.7 \end{bmatrix} \varphi_k + \varepsilon_k & \text{if } \begin{bmatrix} -5 & -1 & 6 \end{bmatrix} \varphi_k < 0 \end{cases}$$

with $\varphi_k = [y_{k-1}\ u_{k-1}\ 1]'$, $|u_k| \leq 5$ and $|\varepsilon_k| \leq 0.1$



initial classification
(6 clusters → quite rough)

Step #2: Refinement Procedure

1. **Parameter update.** For all i, compute $\hat{\theta}_i^{(t+1)}$ as:

$$\hat{\theta}_i^{(t+1)} = \arg\min_{\theta} \max_{(y_k,x_k)\in\mathcal{D}_i^{(t)}} \left| y_k - \varphi_k' \theta \right|$$

projection estimate (linear programming)

2. **Datapoint reassignment.** For each datapoint (y_k, x_k):
   - If $\left| y_k - \varphi_k' \hat{\theta}_i^{(t)} \right| \leq \delta$ for only one i, then assign (y_k, x_k) to cluster $\mathcal{D}_i^{(t)}$
   - If $\left| y_k - \varphi_k' \hat{\theta}_i^{(t)} \right| > \delta$ for all i, then mark (y_k, x_k) as *infeasible*
   - Otherwise, mark (y_k, x_k) as *undecidable*

3. **Termination**

   If $\left\| \hat{\theta}_i^{(t+1)} - \hat{\theta}_i^{(t)} \right\| / \left\| \hat{\theta}_i^{(t)} \right\| \leq \gamma$ for all $i = 1, \ldots, \hat{s}$, then exit. Otherwise, set $t = t+1$ and go to step 1 ($\gamma > 0$ is a given termination threshold)

Step #2: Comments

Comments about the iterative procedure

- Why the projection estimate?

  No feasible datapoint at refinement t becomes infeasible at refinement $t+1$

$$\max_{(y_k,x_k)\in\mathcal{D}_i^{(t)}} \left| y_k - \varphi_k' \hat{\theta}_i^{(t+1)} \right| \leq \max_{(y_k,x_k)\in\mathcal{D}_i^{(t)}} \left| y_k - \varphi_k' \hat{\theta}_i^{(t)} \right| \leq \delta$$

- Why the distinction among *infeasible*, *undecidable*, and *feasible* datapoints?

- Infeasible datapoints are not consistent with any submodel, and may be **outliers** ⇒ neglecting them helps improving the quality of the fit

- Undecidable datapoints are consistent with more than one submodel ⇒ neglecting them helps to reduce misclassifications

Step #3: Reduce Number of Submodels

- **Similarity of the parameter vectors** $(\hat{\theta}_i^{(t)} \approx \hat{\theta}_j^{(t)})$

  If $\alpha_{i^*,j^*} = \min_{1 \le i < j \le \hat{s}} \mu(\hat{\theta}_i^{(t)}, \hat{\theta}_j^{(t)}) \le \alpha$, then merge submodels i^* and j^*

  $$\left(\text{e.g.: } \mu(\hat{\theta}_i^{(t)}, \hat{\theta}_j^{(t)}) \triangleq \left\|\hat{\theta}_i^{(t)} - \hat{\theta}_j^{(t)}\right\| / \min\left\{\left\|\hat{\theta}_i^{(t)}\right\|, \left\|\hat{\theta}_j^{(t)}\right\|\right\}\right)$$

- **Cardinality of the clusters** ($\mathcal{D}_i^{(t)}$ has too few points)

  If $\beta_{i^*} = \min_{1 \le i \le \hat{s}} \frac{\text{card}(\mathcal{D}_i^{(t)})}{N} \le \beta$, then discard the i^*-th submodel

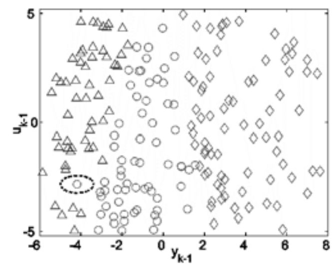
Thresholds α and β should be suitably chosen in order to reduce the number of submodels still preserving a good fit of the data

Example (cont'd)

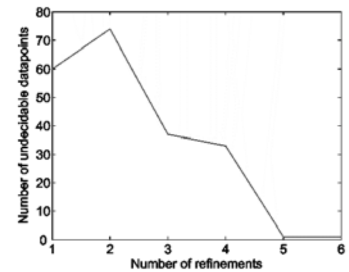
Consider again the PWARX system

$$y_k = \begin{cases} \begin{bmatrix} -0.4 & 1 & 1.5 \end{bmatrix} \varphi_k + \varepsilon_k & \text{if } \begin{bmatrix} 4 & -1 & 10 \end{bmatrix} \varphi_k < 0 \\ \begin{bmatrix} 0.5 & -1 & -0.5 \end{bmatrix} \varphi_k + \varepsilon_k & \text{if } \begin{bmatrix} -4 & 1 & -10 \\ 5 & 1 & -6 \end{bmatrix} \varphi_k \le 0 \\ \begin{bmatrix} -0.3 & 0.5 & -1.7 \end{bmatrix} \varphi_k + \varepsilon_k & \text{if } \begin{bmatrix} -5 & -1 & 6 \end{bmatrix} \varphi_k < 0 \end{cases}$$

with $\varphi_k = [y_{k-1}\ u_{k-1}\ 1]'$, $|u_k| \le 5$ and $|\varepsilon_k| \le 0.1$



Classification of the regression vectors after the refinement (3 clusters)



Number of undecidable datapoints vs number of refinements

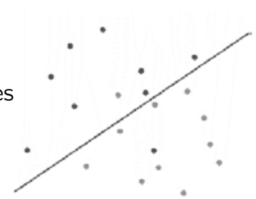
Step #4: Estimation of the Partition

Estimation of the partition of the regressor set

- This step can be performed by computing a *separating hyperplane* for each pair of final clusters F_i of regression vectors
- If two clusters F_i and F_j are not linearly separable, look for a hyperplane that minimizes the number of misclassified points (*generalized separating hyperplane*)
- Linear Support Vector Machines (SVMs) can be used to compute the *optimal generalized separating hyperplane* of two clusters

Alternative:

use multi-category classification techniques (computationally more demanding, but better results)
(Bennet and Mangasarian, 1992)

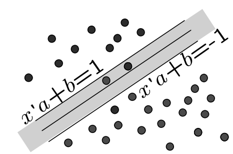


Step #4: Estimation of the Partition

Generalized separating hyperplane and MAX FS

- Given two clusters F_i and F_j, a separating hyperplane $x'a+b=0$ is such that

$$\begin{cases} x_k'a + b \le -1 & \forall x_k \in F_i \\ x_k'a + b \ge 1 & \forall x_k \in F_j \end{cases}$$



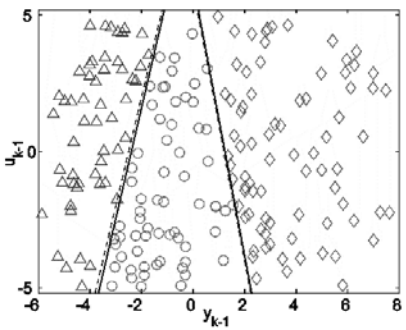
- A solution of the MAX FS problem of the above system of ineqs is a hyperplane that minimizes the number of misclassified points
- The misclassified points, if any, are removed from F_i and/or F_j
- Then, compute the optimal separating hyperplane of F_i and F_j via quadratic programming

Example (cont'd)

Consider again the PWARX system

$$y_k = \begin{cases} \begin{bmatrix} -0.4 & 1 & 1.5 \end{bmatrix} \varphi_k + \varepsilon_k & \text{if } \begin{bmatrix} 4 & -1 & 10 \end{bmatrix} \varphi_k < 0 \\ \begin{bmatrix} 0.5 & -1 & -0.5 \end{bmatrix} \varphi_k + \varepsilon_k & \text{if } \begin{bmatrix} -4 & 1 & -10 \\ 5 & 1 & -6 \end{bmatrix} \varphi_k \leq 0 \\ \begin{bmatrix} -0.3 & 0.5 & -1.7 \end{bmatrix} \varphi_k + \varepsilon_k & \text{if } \begin{bmatrix} -5 & -1 & 6 \end{bmatrix} \varphi_k < 0 \end{cases}$$

with $\varphi_k = [y_{k-1}\ u_{k-1}\ 1]'$, $|u_k| \leq 5$ and $|\varepsilon_k| \leq 0.1$

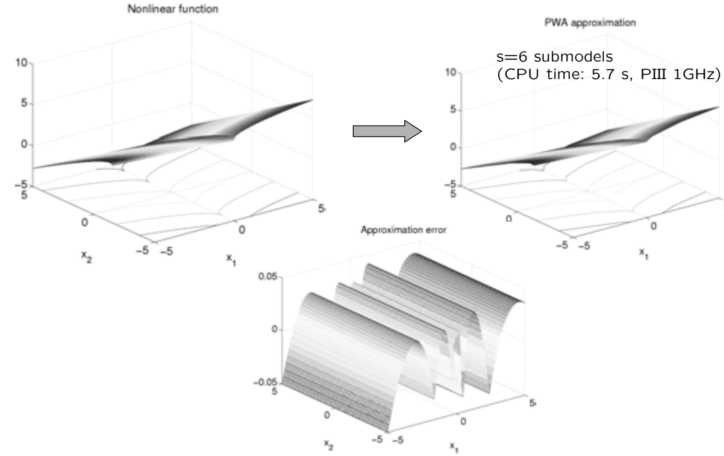


Final classification of the regression vectors, and true (*dashed lines*) and estimated (*solid lines*) partition of the regressor set

Example 2: Nonlinear Fnc Approx.

We want to *hybridize* the nonlinear function $y = \sqrt{|x_1|} - x_2$

N=1000 datapoints, δ=0.05, α=10%, β=1%



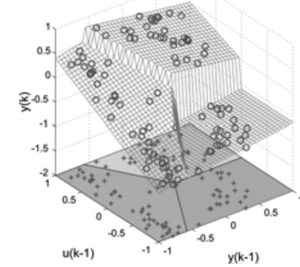
Feature Space Clustering Approach

(thanks to G.Ferrari-Trecate for providing this material)

Assumptions (PWARX Model)

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

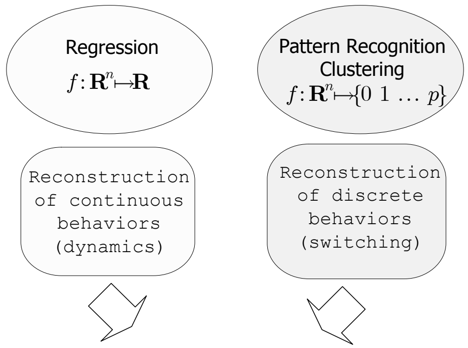
- Model orders n_a, n_b fixed
- The number s of submodels is known



The switching law is assumed unknown: Both the submodels and the shape of the regions must be estimated from the dataset

Hybrid Identification Algorithm

Learning from a finite dataset



Hybrid Identification

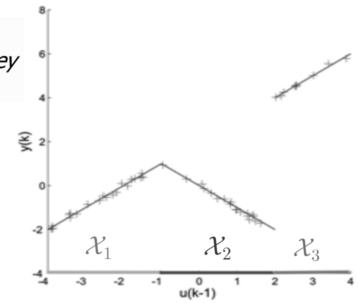
Toy Example

$$y(k) = \begin{cases} u(k-1) + 2 + \epsilon(k) & \text{if} \quad u(k-1) \in \mathcal{X}_1 \\ -u(k-1) + \epsilon(k) & \text{if} \quad u(k-1) \in \mathcal{X}_2 \\ u(k-1) + 2 + \epsilon(k) & \text{if} \quad u(k-1) \in \mathcal{X}_3 \end{cases}$$

The first and the third submodel have the same coefficients (but they are defined on different regions)

Dataset

- $N = 50$ datapoints
- $\epsilon_k \sim \mathcal{N}(0, 0.01)$



Step #1: Build Local Datasets

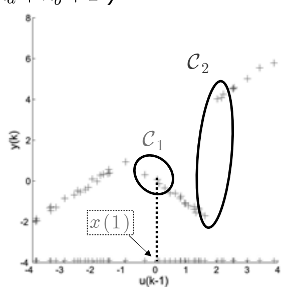
The PWARX model is locally linear:
Small sets of datapoints that are close to each other are likely to belong to the same submodel

For each datapoint $(x(j), y(j))$ build a set $\mathcal{C}_j$ collecting $(x(j), y(j))$ and the first $c-1$ neighboring points ($c > n_a + n_b + 1$)

There is a one-to-one map between each set $\mathcal{C}_j$ and the datapoint $(x(j), y(j))$

Sets collecting points belonging to a single subsystem: Pure sets (e.g. $\mathcal{C}_1$)

Sets collecting points belonging to different subsystems: Mixed sets (e.g. $\mathcal{C}_2$)

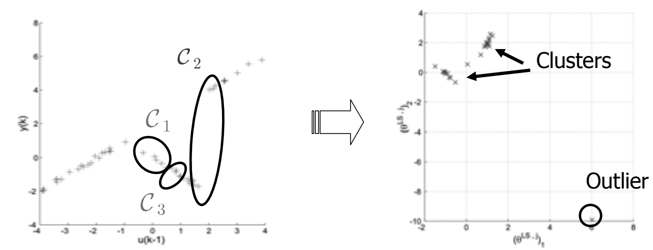


Step #2: Identification of Local Models

For each local dataset $\mathcal{C}_j$ identify an affine model through least squares (vector of coefficients: $\theta^{LS,j}$)

- Pure sets collecting datapoints belonging to the *same* submodel should produce *similar* $\theta^{LS,j}$
- Mixed sets should give *isolated* vectors of coefficients $\theta^{LS,j}$

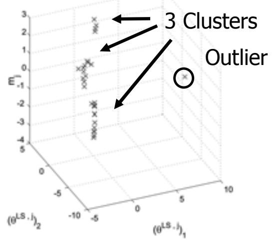
"High" S/N ratio and few mixed sets $\Rightarrow$ Clusters of vectors $\theta^{LS,j}$ + few outliers



The Feature Vectors

Problem: The same vector of coefficients can characterize submodels defined on different regions (Ferrari-Trecate et. *al*, HSCC 2001)

Consider the *feature vectors* $\xi_j = \begin{bmatrix} \theta^{LS,j} \\ m_j \end{bmatrix}$ $\quad m_j = \frac{1}{c}\sum_{(x,y)\in\mathcal{C}_j} x$



The vector ξ_j takes into account the spatial localization of the j-th local model

Step #3: Clustering the feature vectors

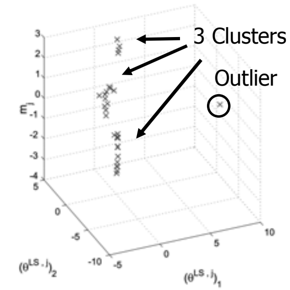
Next problem: find the clusters in the feature space

⇩

The accuracy must not be spoiled by the outliers

⇩

Introduce measures of the confidence one should have about the fact that ξ_j is based on a mixed local dataset



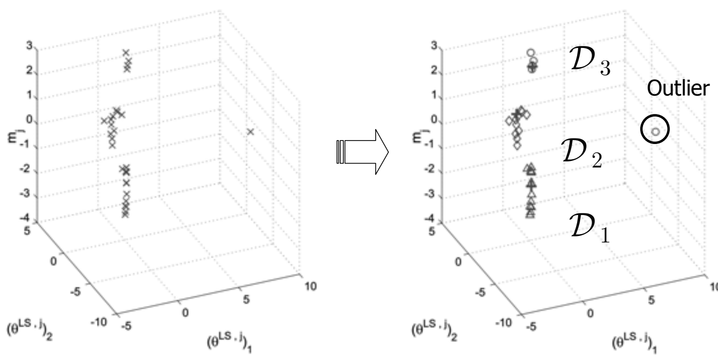
Exploit such measures in a "K-means like" algorithm that divides the feature vectors in subsets $\mathcal{D}_i,\ i=1,\ldots,s$

The clustering algorithm proposed in (Ferrari-Trecate et. *al*, HSCC 2001) guarantees convergence to a (possibly suboptimal) set of clusters in a finite number of iterations

Clustering Step (Toy Example)

Unclassified feature vectors

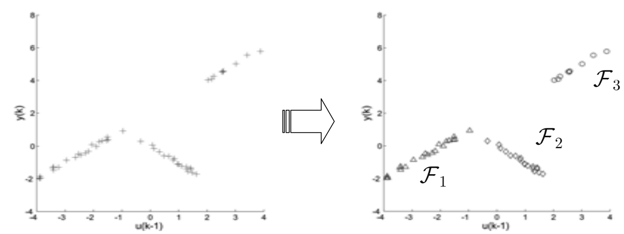
Classified feature vectors



Step #4: Classification of the Datapoints

Build the sets $\mathcal{F}_i\ \ i=1,\ldots,s$ of classified datapoints according to the rule

if $\xi_j \in \mathcal{D}_i$, then $(x(j), y(j)) \in \mathcal{F}_i$



Step #5: Identification of the Submodels

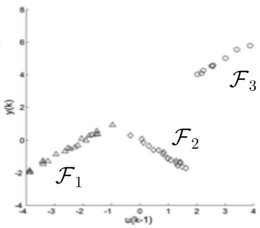
Use the data in each set $\mathcal{F}_i$ for estimating both the affine submodels and the regions

Submodel coefficients:

Weighted Least Squares exploiting the confidence measures

Shape of the polyhedral regions:

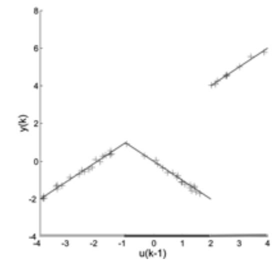
- Linear Support Vector Machines (Vapnik, 1998)
  - solved via linear/quadratic programming
- Multicategory Robust Linear Programming (Bennet & Mangasarian, 1992)



Toy Problem: Identification Results

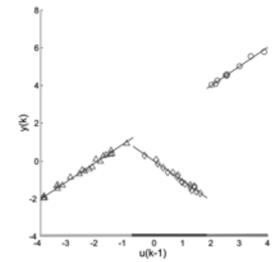
True model

$$y(k)=\begin{cases} u(k-1)+2+\epsilon(k) & u(k-1)\in[-4,-1] \\ -u(k-1)+\epsilon(k) & u(k-1)\in[-1,2] \\ u(k-1)+2+\epsilon(k) & u(k-1)\in[2,4] \end{cases}$$



Identified model

$$y(k)=\begin{cases} u(k-1)+1.99+\epsilon(k) & u(k-1)\in[-4,-0.8] \\ -1.05u(k-1)+0.01+\epsilon(k) & u(k-1)\in[-0.8,1.8] \\ 1.02u(k-1)+1.92+\epsilon(k) & u(k-1)\in[1.8,4] \end{cases}$$

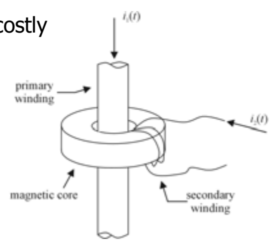


Computational time: 1.26 s. (on a Pentium 600 Mhz running Matlab 5.3)

Example: Industrial Transformer

Industrial transformers used in a protection system:
(S. Bittanti *et al.*, 2001)

- The measurement of $i_1(t)$ is difficult and costly
- The measurement of $i_2(t)$ is easy



Goal: Estimate $i_1(t)$ from $i_2(t)$

Problems:
- Hysteresis and saturations occur for currents of high intensity
- Derive a model for *simulation*
- Sampling time: 5.0000e-005 s.

Identified PWARX model

PWARX model (five regions)

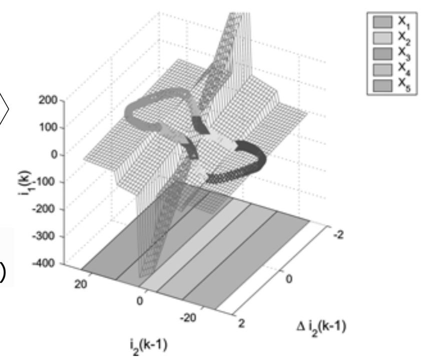
$$i_1(k)=a_{j,1}i_2(k-1)+a_{j,2}\triangle i_2(k-1) \qquad [i_2(k-1)\ \ \Delta i_2(k-1)]\in\mathcal{X}_j$$

$$j=1,\ldots,5$$

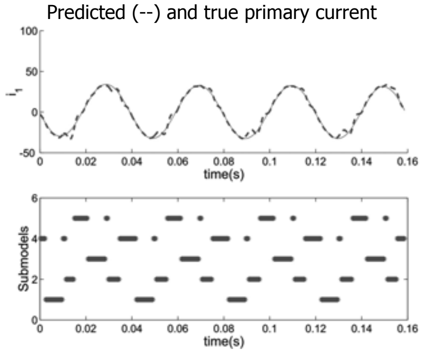
Identified submodels and classified datapoints (440 measurements)

Local datasets $\mathcal{C}_j$ of 50 points

Computational time: 15.76 s. (Pentium 600 Mhz, Matlab 5.3)



Validation Results



- There are nonlinear (ad hoc) simulators for industrial transformers

(S. Bittanti *et al.*, 2001)

Advantage of PWARX models: Simple enough for on-line implementation

Conclusions

- Main goal of hybrid systems identification:
  - **Develop simple switching models from data (or from more complex models) to be used for control/analysis purposes**
- Hybrid system identification is a hard problem
- Theory is still in its infancy
- Some algorithms are already available
- Applications:
  - Biomedical (Analysis of the EEG ⇒ Brain-Computer interface; Dialysis: early assessment of the therapy duration)
  - Ecological (trophic, oxygen and nutrient dynamics in aquatic systems)
  - ... (many others !)