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

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

\[ y_k = \sum_{i=1}^{s} \phi_i x_k + c_k \]

where:

• \( y_k \in \mathbb{R} \) is the system output

• \( x_k \in \mathbb{R}^n \) is the regression vector.

  e.g. \( x_k = [y_{k-1} \ldots y_{k-n_y} \ u_{k-1} \ldots u_{k-n_u}]' \)

• \( c_k \in \mathbb{R} \) is the error

• \( \{x_i\}_{i=1}^s \), \( x_i = \{x : H_i x \leq 0\} \), is a polyhedral partition of the regressor set \( x \subseteq \mathbb{R}^n \)

unknowns: \( \{H_i, \phi_i, s\} \), \( i = 1, \ldots, s \)

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

\[ y_k = \begin{cases} 0.4 \ y_k + c_k & \text{if } 4 \ y_k + c_k < 0 \\ 0.5 \ -1 \ -0.5 \ y_k + c_k & \text{if } -4 \ 1 \ -10 \ y_k \leq 0 \\ -0.3 \ 0.5 \ -1.7 \ y_k + c_k & \text{if } -5 \ -1 \ -6 \ y_k \leq 0 \end{cases} \]

with \( y_k = [y_{k-1} \ u_{k-1}]' \), \( |y_k| \leq 5 \) and \( |c_k| \leq 0.1 \)
### PWA Identification Problem

\[
\begin{align*}
\min_{\theta_j, H_j} & \frac{1}{2N} \sum_{i=1}^{N} \left( \sum_{j=1}^n \| y_i - \varphi_j(\theta_j, x_i) \| \right) \\
\text{subject to} & \quad J_j(\varphi_i) = \begin{cases} 
1 & \text{if } H_j \varphi_i \leq 0 \\
0 & \text{otherwise}
\end{cases} \\
& + \text{linear bounds over } \theta_j, H_j
\end{align*}
\]

| $\| e \|_2$ | Euclidean norm |
| $\| e \|_\infty$ | $\max|e_i|$ $\infty$-norm |
| $\| e \|_1$ | $\sum|e_i|$ 1-norm |

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

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

### Approaches to PWA Identification

- **Mixed-integer linear or quadratic programming**

- **Bounded error & partition of infeasible set of inequalities**

- **K-means clustering in a feature space**

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

### Mixed-Integer Approach

\[
y_i = \varphi_j(\theta_j, x_i) + \sum_{i=1}^{S} \pm \max\{\varphi_j(\theta_i, 0)\} + e_i
\]

**Hinging Hyperplane hybrid models**

**Example:**

\[
y(t) = y(t-1) + 0.2u(t-1) + \max\{-y(t-1) + 2u(t-1), 0\} + \max(2u(t-1) + u(t-1), 0)
\]
Mixed-Integer Approach

\[ \hat{y}(t) = \varphi'(t)\theta_0 + \sum_{i=1}^{N} \pm \max\{\varphi'(t)\theta_i, 0\} \]

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

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.

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

\[
\min_{x, \delta} \begin{bmatrix} x', \delta' \end{bmatrix} \begin{bmatrix} Q & P \\ P & 0 \end{bmatrix} \begin{bmatrix} x \\ \delta \end{bmatrix} + \begin{bmatrix} x \\ \delta \end{bmatrix} \\
\text{s.t. } C \begin{bmatrix} x \\ \delta \end{bmatrix} \leq d \\
\delta \in \{0, 1\}^m \\
(x \in \mathbb{R}^n)
\]

(if Q=0 the problem is an MILP)

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

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

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

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

Mixed-Integer Approach

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

\[ z_i(t) = \max\{\varphi'(t)\theta_i, 0\} = \varphi'(t)\theta_i \delta_i(t) \]

(if-then-else condition)

3. Get linear mixed-integer constraints:

\[ \varphi'(t)\theta_i \leq M\delta_i(t) \]
\[ \varphi'(t)\theta_i \geq \epsilon + (m - \epsilon)(1 - \delta_i(t)) \]
\[ -M\delta_i(t) + z_i(t) \leq 0 \]
\[ m\delta_i(t) - z_i(t) \leq 0 \]
\[ -M(1 - \delta_i(t)) - z_i(t) \leq -\varphi'(t)\theta_i \]
\[ m(1 - \delta_i(t)) + z_i(t) \leq \varphi'(t)\theta_i \]

- \(\epsilon\) 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 \(\theta_i\)).

The identification problem is an MIQP!

Mixed-Integer Approach

Example: Identify the following system

\[ y_t = 0.8y_{t-1} + 0.4u_{t-1} - 0.1 + \max(-0.3y_{t-1} + 0.6u_{t-1} + 0.3, 0) \]

MLP: 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

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

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
\[ y_k = f(x_k) + v(k) \]
\[ f(x_k) = \theta_i \frac{x_k}{x_0} \quad \text{if } x_k \in X_i \text{ for some } i = 1, \ldots, s \]

**Bounded-error**: select a bound \( \delta > 0 \) and require that the identified model satisfies the condition
\[ |y_k - f(x_k)| \leq \delta, \quad \forall k = 1, \ldots, N \]

**Role of \( \delta \)**: trade off between quality of fit and model complexity

**Problem**
Given \( N \) datapoints \((y_k, x_k), k=1, \ldots, N\), estimate the min integer \( s \), a partition \( X_1, \ldots, X_s \), and params \( \theta_1, \ldots, \theta_s \) 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 \( \delta > 0 \) and the (possibly infeasible) system of \( N \) linear inequalities
\[ |y_k - \phi_i \theta_k| \leq \delta, \quad k = 1, \ldots, N, \]
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 \( \theta_i \) 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 \( \theta \) that satisfies the largest number of ineqs
\[ |y_k - \phi_i \theta| \leq \delta, \quad k = 1, \ldots, N \]
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


### 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 \( \theta_i \) is similar, or
   b. remove clusters that contain too few points
4. **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.

Step #2: Refinement Procedure

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

   $$\hat{\theta}_{(i)}^{(t+1)} = \arg \min_{\theta} \max_{(y_{k},x_{k}) \in \mathcal{D}_{(i)}^{(t)}} |y_{k} - \varphi(\theta)| \quad \text{projection estimate}$$

   (linear programming)

2. **Datapoint assignment.** For each datapoint $(y_{k},x_{k})$:

   - If $|y_{k} - \varphi(\hat{\theta}_{(i)}^{(t)})| \leq \delta$ for only one $i$, then assign $(y_{k},x_{k})$ to cluster $\mathcal{D}_{(i)}^{(t)}$
   - If $|y_{k} - \varphi(\hat{\theta}_{(i)}^{(t)})| > \delta$ for all $i$, then mark $(y_{k},x_{k})$ as infeasible
   - Otherwise, mark $(y_{k},x_{k})$ as undecidable

3. **Termination**

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

Example (cont’d)

Consider again the PWARX system

$$\begin{align*}
    y_{t} &= \begin{cases}
        -0.4 & 1 & 1.5 & \omega_{t} + x_{t} & \text{if } 4 \leq -1 & 10 & \omega_{t} < 0 \\
        0.5 & -1 & -0.5 & \omega_{t} + x_{t} & \text{if } -4 \leq 1 & -10 & \omega_{t} \leq 0 \\
        -0.3 & 0.5 & -1.7 & \omega_{t} + x_{t} & \text{if } -5 \leq -1 & 5 & \omega_{t} < 0
    \end{cases}
\end{align*}$$

with $\omega_{t} = [\omega_{t-1} \ u_{t-1} \ 1]'$

$\mathcal{X} = \{0.5 \ -1 \ -0.5 \ \omega_{t} + x_{t} \}$

Step #2: Comments

Comments about the iterative procedure

- Why the projection estimate?
  - No feasible datapoint at refinement $t$ becomes infeasible at refinement $t+1$

$$\max_{(y_{k},x_{k}) \in \mathcal{D}_{(i)}^{(t)}} |y_{k} - \varphi(\hat{\theta}_{(i)}^{(t+1)})| \leq \max_{(y_{k},x_{k}) \in \mathcal{D}_{(i)}^{(t)}} |y_{k} - \varphi(\hat{\theta}_{(i)}^{(t)})| \leq \delta$$

- Why the distinction among infeasible, undecidable, and feasible datapoints?
  - Infeasible datapoints are not consistent with any submodel, and may be outliers $\Rightarrow$ neglecting them helps improving the quality of the fit
  - Undecidable datapoints are consistent with more than one submodel $\Rightarrow$ neglecting them helps to reduce misclassifications
Step #3: Reduce Number of Submodels

- Similarity of the parameter vectors \( \mu(\theta^{(i)}, \theta^{(j)}) \)
  
  If \( \mu(\theta^{(i)}, \theta^{(j)}) \leq \alpha \), then merge models \( i^* \) and \( j^* \)
  
  (e.g.: \( \mu(\theta^{(i)}, \theta^{(j)}) = \frac{1}{N} \sum_i \min \{|\theta^{(i)}_{\cdot} - \theta^{(j)}_{\cdot}|\} \))

- Cardinality of the clusters \( |\mathcal{D}^{(i)}| \) has too few points
  
  If \( \beta \leq \min \frac{\text{card}(\mathcal{D}^{(i)})}{N} \), then discard the \( i \)-th submodel

Thresholds \( \alpha \) and \( \beta \) 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

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)

Step #4: Estimation of the Partition

Generalized separating hyperplane and MAX FS

- Given two clusters \( F_i \) and \( F_j \), a separating hyperplane \( x^T a + b = 0 \) is such that

\[
\begin{align*}
    x^T a + b &\leq -1 \quad \forall x_k \in F_i \\
    x^T a + b &\geq 1 \quad \forall x_k \in F_j
\end{align*}
\]

- A solution of the MAX FS problem of the above system of inequ 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

\[ y_k = \begin{cases} 
-0.4 & 1 & 1.5 \varphi_0 + \varphi_1 + \tau_k \quad \text{if} \quad [4 \ -1 \ 10] \varphi_0 < 0 \\
0.5 & -1 & -0.5 \varphi_0 + \varphi_1 + \tau_k \quad \text{if} \quad [-4 \ 1 \ -10] \varphi_0 \leq 0 \\
-0.3 & 0.5 & -1.7 \varphi_0 + \varphi_1 + \tau_k \quad \text{if} \quad [-5 \ -1 \ 6] \varphi_0 \leq 0 
\end{cases} \]

with \( \varphi_0 = [w_{k-1} \ u_{k-1} \ 1]' \)...

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, \( \delta = 0.05, \alpha = 10\%, \beta = 1\% \)

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

Feature Space Clustering Approach
(Thanks to G. Ferrari-Trecate for providing this material)

Assumptions (PWARX Model)

- 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

- **Regression** \( f : \mathbb{R}^n \rightarrow \mathbb{R} \)
- **Pattern Recognition**
  - **Clustering** \( f : \mathbb{R}^n \rightarrow \{0, 1, \ldots, p\} \)
- **Reconstruction of continuous behaviors (dynamics)**
- **Reconstruction of discrete behaviors (switching)**

Hybrid Identification

---

**Toy Example**

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

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

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 \( C_j \) 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 \( C_j \) and the datapoint \( (x(j), y(j)) \)

Sets collecting points belonging to a single subsystem: Pure sets (e.g., \( C_1 \))

Sets collecting points belonging to different subsystems: Mixed sets (e.g., \( C_2 \))

---

**Step #2: Identification of Local Models**

For each local dataset \( C_j \), identify an affine model through least squares

\( \hat{\theta}^{LS,j} \)

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

"High" S/N ratio and few mixed sets \( \Rightarrow \) Clusters of vectors \( \hat{\theta}^{LS,j} \) + 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 $\xi_j = \left( \frac{\partial m_j}{\partial y_j} \right) m_j = \frac{1}{2} \sum_{x,y \in C_j} x$

The vector $\xi_j$ 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 $\xi_j$ is based on a mixed local dataset

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

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

**Step #4: Classification of the Datapoints**

Build the sets $F_i$, $i=1,\ldots,s$ of classified datapoints according to the rule

$\xi_j \in D_i$, then $(x(j), y(j)) \in F_i$
Step #5: Identification of the Submodels

Use the data in each set $F_i$ for estimating both the affine submodels and the regions.

Submodel coefficients:
- Weighted Least Squares exploiting the confidence measures
- Linear Support Vector Machines (Vapnik, 1998)
- Multicategory Robust Linear Programming (Bennet & Mangasarian, 1992)

Shape of the polyhedral regions:
- Hysteresis and saturations occur for currents of high intensity
- Derive a model for simulation
- Sampling time: 5.0000e-005 s.

Example: Industrial Transformer

Industrial transformers used in a protection system:
- 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:
- The measurement of $i_1(t)$ is difficult and costly
- Derive a model for simulation
- Sampling time: 5.0000e-005 s.

Toy Problem: Identification Results

True model
\[
y(u(k-1) + 2 + s(k)) = u(k-1) \in [-4, -1) \]

Identified model
\[
y(u(k-1) + 1.90 + s(k)) = u(k-1) \in [-4, -0.08] \]

True model
\[
y(u(k-1) + 0.01 + s(k)) = u(k-1) \in [-0.89, -0.8] \]

Identified model
\[
y(u(k-1) + 1.92 + s(k)) = u(k-1) \in [-3.8, -4.8] \]

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

Identified PWARX model

PWARX model (five regions)
\[
i_j(k) = a_{j,1}i_2(k-1) + a_{j,2}\Delta i_2(k-1) \quad \Delta i_2(k-1) \in \mathcal{X}_j
\]

$\mathcal{X}_j$, $j = 1, \ldots, 5$

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 !)