Subspace identification of switching model
Komi Pekpe, Gilles Gasso, Gilles Mourot, José Ragot

To cite this version:
Komi Pekpe, Gilles Gasso, Gilles Mourot, José Ragot. Subspace identification of switching model. 13th IFAC Symposium on System Identification, SYSID, Aug 2003, Rotterdam, Netherlands. hal-01188371

HAL Id: hal-01188371
https://hal.science/hal-01188371
Submitted on 29 Aug 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers. L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Subspace identification of switching model

Komi Midzodzi PEKPE, Komi GASSO, Gilles MOUROT, José RAGOT
Centre de Recherche en Automatique de Nancy-CNRS UMR 7039
2, Avenue de la Forêt de Haye 54516 Vandœuvre-lès-Nancy Cedex France
Tel: (33) 3 83 59 57 04, Fax: (33) 3 83 59 56 44
E-mail: {kpekpe, gmourot, jragot}@ensem.inpl-nancy.fr

Abstract — Subspace identification of switching model is considered in this paper. Here the switching model is supposed to be a sum of weighted linear models. The method established uses recursive subspace identification to estimate the switching function and least squares method for local model Markov parameters estimation. To perform the computation of the weighting function a two-steps algorithm (switching times determination and model merging) is given. Finally the local model parameter estimation is based on the estimation of the Markov parameters.

Keywords — switching model, Markov parameters, least squares, weighting function, recursive subspace identification.

I. Introduction

The objective of this paper is to identify switching models. Here a switching model structure is considered as a sum of weighted linear systems (local models), but only one model is active at any time. In this paper, we use a weighting function which determines the active model. Our goal is to estimate the weighting function and identify the local model, using the knowledge of the inputs and outputs only. Thus, knowing the weighting function allows to determine which local model is active. The estimation of the switching function is made by a recursive subspace identification method. The proposed method is not sensible to the change in input dynamics because the recursive subspace method proposed in [4] is not sensible to this fact. Then the switching times correspond only to a change in the system dynamic. To identify the local models, we use least squares method. The parameter estimation is done by the extraction of the Markov parameter matrix. The advantage of the proposed method results in fact that there is not any stage of nonlinear optimization, and the parameter estimation is unbiased.

After the formulation of the problem, the notations are introduced and the estimation of the Markov parameters is achieved. In section 4 the determination of the weighting function is performed. The identification of the local models procedure is given. Finally an example illustrates the performance of the proposed method.

II. Formulation of the Model

The output of the switching model can be modeled as a weighted sum of outputs of h linear models as follows:

\[ y_k = \sum_{s=1}^{h} \omega_{s,k} y_{s,k}, \]  \( y_{s,k} \) \( k \)

where \( y_k \in \mathbb{R}^r \), any weight \( \omega_{s,k} \in \{0, 1\} \) and \( \sum_{s=1}^{h} \omega_{s,k} = 1 \), \( \forall k \in [1, q] \) (if we have q measurements of the inputs and outputs). \( \omega_{s,k} \) represents the weighting function and \( y_{s,k} \) the output of the \( s^{th} \) local model.

Any local model is supposed linear of order n_s and can be described by the equation:

\[
\begin{align*}
    x_{s,k+1} &= A_s x_{s,k} + B_s u_k, \\
    y_{s,k} &= C_s x_{s,k} + D_s u_k + e_k
\end{align*}
\]

Here the output error \( e_k \in \mathbb{R}^r \) is assumed to be a zero mean white noise sequence and uncorrelated with \( u_k \in \mathbb{R}^m \) and has covariance matrix

\[ \mathbb{E}e_k e_t = \begin{cases} R > 0, k=t \\
0, \text{ otherwise.} \end{cases} \]

We suppose each local model is stable.

The inputs \( u_k \) and outputs \( y_k \) of the switching model are supposed to be known. The object is to determine:

- the weighting function \( \omega_{s,k} \) for each model,
- the order \( n_s \) and the \( h \) linear local model parameters.

The parameters to be determined are \( A_s, B_s, C_s \) and \( D_s \).

Note that, to obtain the \( g^{th} \) weighting outputs, it is sufficient to make the product of the global outputs \( y_k \) by the \( g^{th} \) weighting function:

\[ \omega_{g,k} y_k = \omega_{g,k} \sum_{s=1}^{h} \omega_{s,k} y_{s,k} = \omega_{g,k} y_{g,k}, \forall k \in [1, q] \]

because

\[ (\omega_{g,k})^2 = \omega_{g,k} \text{ and } \omega_{g,k} \times \omega_{s,k} = 0, \text{ if } g \neq s, \forall k \in [1, q] \]  \( \omega_{g,k} \) \( k \)
In the following we use a recursive subspace identification technique to obtain the weighting function \( \omega_{s,k} \). In order to obtain local Markov parameters, least squares tools will be used. In the next section the matrices used later are defined.

### III. The System Matrices

For the \( s^{th} \) model, we can set the following definitions (note that, for simplicity we use \( i \) instead of \( i_s \) in this article).

Output weighted block Hankel matrix of the \( s^{th} \) model is defined as \((i>n_s)\):

\[
Y_{s,\omega}^i = \\
\begin{pmatrix}
\omega_{s,i} y_{s,i} & \omega_{s,i+1} y_{s,i+1} & \cdots & \omega_{s,i+j-1} y_{s,i+j-1}
\end{pmatrix}
\]  
(5)

Input block Hankel matrix is defined as:

\[
U = \\
\begin{pmatrix}
u_1 & u_2 & \cdots & u_j \\
u_2 & u_3 & \cdots & u_{j+1} \\
\vdots & \vdots & \ddots & \vdots \\
u_i & u_{i+1} & \cdots & u_{i+j-1}
\end{pmatrix}
\]  
(6)

The same definition is made for the Hankel block matrix \( E \) for the noise \( e_k \).

The state sequence matrix \( X_s \) and the weighting matrix of the \( s^{th} \) model \( \Omega_s \) are defined as:

\[
X_s = (x_{s,1}^T x_{s,2}^T \cdots x_{s,j}^T)^T
\]  
(7)

and

\[
\Omega_s = \\
\begin{pmatrix}
\omega_{s,i} & 0 & 0 & 0 \\
0 & \omega_{s,i+1} & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \omega_{s,i+j-1}
\end{pmatrix}
\]  
(8)

Then we have:

\[
Y_{s,\omega}^i = Y_i \omega_{s,i} \Omega_s
\]  
(9)

The extended observability matrix \( \Gamma_{i,s} \) of the \( s^{th} \) model is defined as:

\[
\Gamma_{i,s} = \\
\begin{pmatrix}
C_s & C_s A_s & \cdots & C_s A_{s}^{i-1}
\end{pmatrix} \in \mathbb{R}^{q \times n_s}
\]  
(10)

The extended controllability matrix \( C_{i,s} \) of the \( s^{th} \) model is defined as:

\[
C_{i,s} = (B_s A_s B_s \cdots A_{s}^{i-1} B_s)
\]  
(11)

The Markov parameter matrix for the deterministic part \( H^d_s \) and for the stochastic part \( H^\nu_s \) of the system are defined as\(^1\):

\[
H_{s,i}^d = (C_s A_{s}^{i-2} B_s \quad C_s A_{s}^{i-3} B_s \quad \cdots \quad C_s B_s \quad D_s)
\]  
(12)

\[
H_{s,i}^\nu = (0 \quad 0 \quad \cdots \quad 0 \quad I)
\]  
(13)

### IV. Estimation of the Markov Parameter Matrix for the Local Model

**Local matrix input-output equation**

It is essential to write the matrix input-output relation, that allows the extraction of the Markov parameter matrix. The following theorem summarizes this relation.

**Theorem 1:**

\[
Y_{s,\omega}^i = C_s A_{s}^{i-1} X_s \Omega_s + H_{s,i}^d U \Omega_s + H_{s,i}^\nu E \Omega_s
\]  
(14)

The proof of the theorem is established in appendix 1.

Now, the objective is to eliminate the term depending on the state \( X_s \) and the noise \( E \), in order to obtain the Toeplitz matrix \( H_{s,i}^d \). The proposed method uses least squares and may be summarize in the next theorem.

**Theorem 2:** Under the assumptions that:

1. the \( s^{th} \) local model is stable,
2. the matrix \( U \Omega_s \) has full rank,
3. with an adequate value of \( i \), \( A_{s}^{i-1} \) is neglected, we have:

\[
Y_{s,\omega}^i (U \Omega_s)^T (U \Omega_s (U \Omega_s)^T)^{-1} \to H_{s,i}^d
\]  
(15)

The proof of the theorem can be found in appendix 2.

Now, from equation 15, we have an estimation of the Markov parameter matrix \( H_{s,i}^d \).

**Remark 1:** to estimate the Markov parameter matrix \( H_{s,i}^d \) by equation 15, we need the knowledge of the weighting function \( \Omega_s \). We use recursive subspace identification to compute this weighting function in the next section.

### V. Determination of the Weighting Function

It is important to note that the previous result involves the state matrix of the system which depends on the active local model.

The recursive subspace approach is used to determine the switching time between local models. As the local models are stable, the recursive subspace identification can be applied. This identification method does not allow to detect the change in inputs dynamics; therefore if a change

---

\(^1\)The superscript ‘d’ and ‘st’ stand for ‘deterministic’ and ‘stochastic’ respectively.
is detected it corresponds only to change in the system dynamics. We recommend to the reader, to refer to [4] for the details of the recursive subspace identification procedure.

To begin with, we make a first computation of the weighting function with the recursive subspace algorithm. Then we merge the models which are closed to each other, and we recompute the new weighting functions.

A. The first computation of the weighting functions

The switching time is determined by the recursive subspace identification algorithm. Here we summarize the main stages of the algorithm of Oku and al.[4].

Let us define:

\[ \Phi_j = \begin{pmatrix} \varphi_1(1) & \varphi_2(2) & \ldots & \varphi_{i}(j-i) \end{pmatrix} \]  

(16)

\[ \varphi_i(k) = u_i(k) = \begin{pmatrix} u_k \\ u_{i+k-1} \end{pmatrix}, k \in [1, j-i]. \]  

(17)

\[ U_j = \begin{pmatrix} u_i(i) & u_i(i+1) & \ldots & u_i(j) \end{pmatrix} \]  

(18)

\[ Y_j \] is defined similarly to (18). 

\[ \Theta_j \] which is the product of the extended observability matrix and the extended controllability matrix, can be estimated by the formula:

\[ \hat{\Theta}_j = Y_j \Pi U_j^T \Phi_j^T \Psi_j \] with

\[ \Psi_j = \begin{pmatrix} \Phi_j \Pi U_j^T \Phi_j^T \end{pmatrix}^{-1} \] and

\[ \Pi U_j = I - F(T)(FT)^{-1}F, \] where I is the identity matrix with the appropriate size and F is a matrix.

If we define \( P_j \) as \( P_j = (U_j U_j^T)^{-1} \), \( \hat{\Theta}_j \) can be obtained also by a usual formulation of the recursive least squares method with a forgetting factor \( \gamma (\gamma < 1, \text{see [4]}): \)

\[ \hat{\Theta}_j = \hat{\Theta}_{j-1} - \beta_j (e_j + \hat{\Theta}_{j-1} q_j^T \Psi_j), \]  

(19)

\[ \Psi_j = \frac{1}{\gamma} (\Psi_{j-1} - \beta_j \Psi_{j-1} q_j^T \Psi_{j-1}), \]  

(20)

\[ P_j = \frac{1}{\gamma} (P_{j-1} - \alpha_j P_{j-1} u_i(j) u_i(j)^T P_{j-1}), \]  

(21)

\[ Y_j U_j^T = \gamma Y_{j-1} U_j^T + y_i(j) u_i(j)^T, \]  

(22)

\[ \Phi_j U_j^T = \gamma \Phi_{j-1} U_j^T + \varphi_i(j-i) u_i(j)^T, \]  

(23)

\[ \alpha_j = \left( \gamma + u_i(j)^T P_{j-1} u_i(j) \right)^{-1}, \]  

(24)

\[ \beta_j = \left( \frac{1}{\alpha_j} + q_j^T \Psi_{j-1} \right)^{-1}, \]  

(25)

\[ e_j = y_i(j) - Y_{j-1} U_j^T P_{j-1} u_i(j), \]  

(26)

\[ q_j = \Phi_{j-1} U_j^T P_{j-1} u_i(j)^T - \varphi_i(j-i), \]  

(27)

Let \( \lambda \) be a threshold designed according to the \( \chi^2 \) distribution with \( \ell_i \) degrees of freedom. The distance from the estimated parameter \( \hat{\Theta}_j \) to the true parameter \( \Theta_j \), \( D(\hat{\Theta}_j, \Theta_j) \) is defined as:

\[ D(\hat{\Theta}_j, \Theta_j) = \frac{1}{\sigma^2} \left( \Phi_j \Pi U_j \Phi_j^T \right) (\hat{\Theta}_j - \Theta_j)^T, \]  

(28)

where \( e_j = Y_j - \hat{Y}_j \), \( \hat{Y}_j \) is an estimation of \( Y_j \). An estimation of the variance of the modeling error \( \sigma^2 \) is given by (4):

\[ \sigma^2_j = \frac{1}{n(j-i+1) - 2m[i]} \times \tr (Y_j - \hat{\Theta}_j \Phi_j \Pi U_j \hat{\Theta}_j^T), \]  

(29)

The change test is defined as (see [4]):

if \( D(\hat{\Theta}_j, \Theta_j) < \lambda \) : no change has occurred,

if \( D(\hat{\Theta}_j, \Theta_j) > \lambda \) : a change has occurred.

Note that in the implementation, the true parameter \( \Theta_j \) is replaced by an approximation: \( \hat{\Theta}_j \) (see [4]) computed as a least squares estimation of \( \Theta_j \) over a sliding window:

\[ \hat{\Theta}_j = \arg \min_{\Theta_j} \tr (\hat{Y} - \Theta \Phi_j \Pi U_j \hat{\Theta}_j^T), \]  

(30)

with

\[ \hat{Y}_j = \begin{pmatrix} y_i(j - L + 2i) & \ldots & y_i(j) \end{pmatrix}, \]  

(31)

\[ \Phi_j = \begin{pmatrix} \varphi_i(j-L+i) & \ldots & \varphi_i(j-i) \end{pmatrix} \]  

(32)

Once a change at time instant \( t \) is detected, the recursive update equation are re-initialized at time instant \( t \), the re-initialization technique is proposed in [4].

The first computation of the weighting functions is described by the algorithm 1 exposed below.

**Algorithm 1** (first computation of the weighting functions):

**Step 1:** (initial conditions)

set \( s = 1 \) and \( k = 0 \), where \( s \) is the local model index and \( k \) is the time index\(^2\).

**Step 2:** (change has not occurred)

if change doesn’t occur (i.e. \( D(\hat{\Theta}_j, \Theta_j) < \lambda \)) then:

\( k = k+1 \), go to step 4.

**Step 3:** (change has occurred)

if change occurs (i.e. \( D(\hat{\Theta}_j, \Theta_j) > \lambda \)) then:

\( s = s+1 \) and \( k = k+1 \).

**Step 4:** (computation of the weighting function)

model "s" is active: \( \omega_{s,k} \leftarrow 1 \) and \( \omega_{r,k} \leftarrow 0 \), \( \forall r \neq s \).

**Step 5:** (test of stop)

\( h \leftarrow s \) (h is the number of identified models)

if \( k < q \) then go to step 2,

if \( k = q \) then stop.

B. The fusion of model

After the first computation of the weighting function by the preceding algorithm, we estimate the Markov parameter matrix \( H_{s,i}^d \) (for the h local models ). Then, we seek the models which are ‘similar’. Firstly, we estimate the covariance of the Markov parameter matrix \( H_{s,i}^d \).

\(^2\) The left arrow "\( \leftarrow \)" denote the replacement of the value of the left hand side, by the right hand side.
B.1 Covariance estimate

The variance of the estimate error \( \sigma_y^2 \) can be unbiasedly estimated by:

\[
\hat{\sigma}^2_y = \frac{1}{i(j-m)} \text{trace} \left( \varepsilon \varepsilon^T \right),
\]

where:

\[
\varepsilon = Y_{s_i \omega} - \hat{Y}_{s_i \omega} \quad (\text{see } 9), \quad \hat{Y}_{s_i \omega} \text{ is an estimation of } Y_{s_i \omega}.
\]

The variance \( \hat{\Sigma}_{d_{s_i}} \) of the output estimated Markov parameter matrix \( \hat{H}_{d_{s_i}} \) is estimated by the following lemma.

Lemma 1:

Unbiased estimation of \( \hat{\Sigma}_{d_{s_i}} \) is given by the formula:

\[
\hat{\Sigma}_{d_{s_i}} = \hat{\Theta} \hat{\Theta}^T,
\]

where:

\[
\hat{\Theta} = \frac{1}{ \sum_{s=1}^{\text{size} \left( B \right)} } \sum_{s=1}^{\text{size} \left( B \right)} \hat{\Theta}_s.
\]

Now we can make a test on the models to find the models which are close to each other.

B.2 The 'similar' models

The aim is to find the models close to each other (or "similar"). For that purpose, we define the distance from the estimated Markov parameters of model \( s \) to the parameters of the model \( r \) as:

\[
d(\hat{H}^d_{d_{s_i}}, \hat{H}^d_{d_{r_i}}) = \text{Trace} \left( (\hat{H}^d_{d_{s_i}} - \hat{H}^d_{d_{r_i}}) \hat{\Sigma}_{d_{s_i}} (\hat{H}^d_{d_{s_i}} - \hat{H}^d_{d_{r_i}})^T \right),
\]

with:

\[
\hat{\Sigma}_{d_{s_i}} = \frac{N_s}{N_s + N_r} \hat{\Sigma}_{d_{s_i}} + \frac{N_r}{N_s + N_r} \hat{\Sigma}_{d_{r_i}},
\]

where \( N_s = \sum_{k=1}^{\text{size} \left( B \right)} \omega_{s,k} \) and \( N_r = \sum_{k=1}^{\text{size} \left( B \right)} \omega_{r,k} \).

If \( \delta \) is a threshold designed according to the \( \chi^2 \) distribution with \( \ell \) degrees of freedom, then we perform the following 'similarity test'

B.21 'similarity test'

The models \( s \) and \( r \) are similar if:

1. if \( d(\hat{H}^d_{d_{s_i}}, \hat{H}^d_{d_{r_i}}) \geq \delta \): the model \( s \) and the model \( r \) are not 'similar',
2. if \( d(\hat{H}^d_{d_{s_i}}, \hat{H}^d_{d_{r_i}}) \leq \delta \): the model \( s \) and the model \( r \) are 'similar'.

B.3 Model merging

If \( B_{s_0} \) is the set of models which are "similar" to \( s_0 \), the models belonging to this set are replaced by the new model \( s_0^* \). The new weighting function for the new model \( s_0^* \) is:

\[
\omega_{s_0,k} = \sum_{s \in B_{s_0}} \omega_{s,k} \quad \text{for } k=1,...,q; \quad \text{moreover we estimate the Markov parameters of the new model } s_0^*.
\]

Remark 2: It is necessary to make a new "similarity test" (see B.21) for the merging of the new model (after a first merging), because it can exist two "similar" local models which have not been detected in the first step, because of the insufficiency of data used to identify the local models. But, by the new computation of the weighting functions and Markov parameters, the number of measurements used to identify the new local models is greater than those which are used to estimate the previous local models.

The merging of the local models is described by algorithm 2.

Algorithm 2 (models merging):

Step 1: (initialization)

\[ s_t = 0; \]

Step 2: (computation of Markov parameters)

compute the Markov parameters (by theorem 2)

Step 3: (test)

Check the "similarity test" (see B.21), and construct the set of models which are "similar".

Step 4: (replacement of the weighting functions)

For each set of models \( B_{s_t} \), if cardinal of \( B_{s_t} \) is greater than 1 then:

1. if \( s_t = 1 \) then go to 1,
2. if \( s_t = 0 \) then stop.

Now we have the final weighting function, we estimate the order and the parameters of each local model. The estimation procedure is described in next section.

VI. Estimation of the system parameters

The goal of this section is to estimate the system order \( n_s \) and matrices \( A_s, B_s, C_s, D_s \) for each local model.

- The system matrix \( D_s \) is directly obtained from the Markov parameter matrix \( H^d_{d_{s_i}} \) (see 12).
- Having the Markov parameters (by theorem 2), one can use the algorithm of Kung [3], Ho and Kalman [1], Era [2], Zeiger and McEwen [6] to estimate the system matrices \( A_s, B_s, C_s \).
- In this paper we set \( i > (n_s - 1) \). Let \( \nu = \text{integer}(i/2) \).
- Following we summarize the Era algorithm (minimal and balanced realization [2]).

1. Build the Hankel matrices \( H^0_{\nu,s} \) and \( H^1_{\nu,s} \) which contains the Markov parameters and are defined by:

\[
H^0_{\nu,s} = \begin{pmatrix}
C_s A^k B_s & C_s A^{k+1} B_s & \cdots & C_s A^{k+n-1} B_s \\
C_s A^{k+1} B_s & C_s A^{k+2} B_s & \cdots & C_s A^{k+n} B_s \\
\vdots & \vdots & \ddots & \vdots \\
C_s A^{k+n-2} B_s & C_s A^{k+n-1} B_s & \cdots & C_s A^{k+2n-2} B_s
\end{pmatrix}
\]

2. Make a singular values decomposition of the matrix \( H^0_{\nu,s} \):

\[ \text{Integer}(i) \text{ is the integer part of } i. \]
\[ H^0_{\nu,s} = \begin{pmatrix} U_1 & V_1^T \\ 0 & S_2 \end{pmatrix} \begin{pmatrix} S_1 & 0 \\ 0 & S_2 \end{pmatrix} \begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix} \]

\[ \approx U_1 S_1 V_1^T \]  

(37)

where \( S_2 \) contains the neglected singular values.

- The system order is equal to the number of singular values in \( S_1 \).
- The extended observability matrix \( \Gamma_{\nu,s} = U_1 S_1^{1/2} \), and the controllability matrix \( C_{\nu,s} = S_1^{1/2} V_1^T \) are computed. Note that:
  - \( H^0_{\nu,s} = \Gamma_{\nu,s} C_{\nu,s} \).
  - The matrix \( C_s \) can be determined from the first \( \ell \) rows of \( \Gamma_{\nu,s} \).
  - The matrix \( A_s \) is computed. The index \( s \).
  - The matrix \( A_s \) is given by the formula:
    \[ A_s = S_1^{1/2} U_1^T H^0_{\nu,s} V_1 S_1^{1/2} \].

VII. THE SIMULATION EXAMPLE

We consider a third order system described as:

\[ x_{k+1} = A_s x_k + B_s u_k + K e_k \]

where \( s \) takes the value 1 on the intervals [1, 999] and [1800, 2499] and the value 2 on the intervals [1000, 1799] and [2500, 3500]; and we have:

\[ A_1 = \begin{pmatrix} 0.32 & 0.31 & 0 \\ -0.32 & 0.31 & 0 \\ 0 & 0 & -0.18 \end{pmatrix}, B_1 = \begin{pmatrix} 0.9 & -0.7 \\ 0.71 & -0.5 \\ 0.8 & 0.47 \end{pmatrix} \]

\[ C_1 = \begin{pmatrix} -0.55 & 0.2 & 0.8 \\ 0.45 & 0.3 & 0.58 \end{pmatrix}, D_1 = \begin{pmatrix} 0.97 & 0.63 \\ -0.32 & 0.95 \end{pmatrix} \]

\[ A_2 = \begin{pmatrix} -0.1 & -0.4 & 0 \\ 0.5 & -0.4 & 0 \\ 0 & 0 & 0.26 \end{pmatrix}, B_2 = \begin{pmatrix} 0.1 & -0.6 \\ 0.32 & -0.66 \\ 0.3 & 0.82 \end{pmatrix} \]

\[ C_2 = \begin{pmatrix} -0.8 & -0.1 & 0.7 \\ 0.3 & 0.48 & 0.9 \end{pmatrix}, D_2 = \begin{pmatrix} 0.5 & 0.3 \\ -0.2 & -0.5 \end{pmatrix} \]

Moreover \( u_k \in \mathbb{R}^2 \), and \( y_k \in \mathbb{R}^2 \), the noise \( e_k \in \mathbb{R}^2 \) is a zero mean white noise and \( K = \frac{\sqrt{2}}{\sqrt{3}} I_2 \).

We suppose that we have \( i+j+1=3500 \) measurements of the inputs and outputs. The inputs \( u_k \) are white noises.

A. The first computation of the weights (use algorithm 1)

For the implementation of the recursive subspace method, we adopt \( i=15 \). Since the dimension of the outputs is \( \ell=2 \), the degree of freedom of the \( \chi^2 \) distribution is \( \ell i=30 \). The exponential forgetting factor \( \gamma \) is taken as 0.98 (see [4]).

B. The fusion of the weighting functions (algorithm 2)

The estimated poles show that the local models 1 and 3 are "similar" (figure 2). The same remark hold for local models 2 and 4. That is confirmed by the "similarity test". The distance \( d(H^d_{s,i}, H^d_{s,i}) \) of "similarity test" allows to merge the models without ambiguity. For hundred experiments we carried out the distance and we have:

\[ d(H^d_{s,i}, H^d_{s,i}) \in [2300, 2800] \]
The estimated poles of the 1st and 2nd new local models according to \( \omega_{1,k}^* \) and \( \omega_{2,k}^* \) respectively.

The figure 3 shows the 100 Monte Carlo realizations for the estimated poles of the new models obtained with \( \omega_{1,k}^* \) and \( \omega_{2,k}^* \). As we can see, the new weighting functions \( \omega_{1,k}^* \) and \( \omega_{2,k}^* \) improve the variance of the estimated poles in each case. The estimated poles are unbiased.

VIII. Conclusion

Switching model identification is considered in this paper. The technique is based on subspace formulation and uses least squares method for the parameter estimation. The switching model is supposed to be a weighted sum of a local models. A recursive subspace identification technique is used to determine the switching time and a merging algorithm is given to estimate the final weighting function. Finally we estimate the local systems order and parameters from the new estimation of the local models Markov parameters. An illustrating example shows the application of the method.

IX. Appendix 1

Proof: The proof of theorem 1 is established in free noise case, the stochastic case is obtained easily.

Form equation 5 we have:

\[
Y_i^{\dagger} = \left[ \begin{array}{cccc}
\omega_{s,i}y_{s,i+1} & \omega_{s,i+1}y_{s,i+1} & \ldots & \omega_{s,i+j-1}y_{s,i+j-1}
\end{array} \right]
\]

\[
\Rightarrow \ Y_i^{\dagger} = \left[ \begin{array}{c}
\omega_{s,i}(C_sA_s^{-1}x_{s,i}+C_sA_s^{i-2}B_su_1+\ldots+D_su_i) \\
\omega_{s,i+j-1}(C_sA_s^{i-1}x_{s,j}+C_sA_s^{i-2}B_su_j+\ldots+D_su_{i+j-1})
\end{array} \right]
\]

\[
\Rightarrow \ Y_i^{\dagger} = C_sA_s^{i-1}X_s\Omega_s +
\]

\[
H_{s,i}^{d}(\begin{array}{cccc}
\omega_{s,i}u_1 & \omega_{s,i+j-1}u_j \\
\omega_{s,i}u_2 & \omega_{s,i+j-1}u_j \\
\vdots & \vdots \\
\omega_{s,i}u_i & \omega_{s,i+j-1}u_i+j-1
\end{array})
\]

Finally we obtain :

\[
Y_{s,i} = C_sA_s^{i-1}X_s\Omega_s + H_{s,i}^{d}U\Omega_s
\]

X. Appendix 2

Proof:

\[
Y_{s,i} = C_sA_s^{i-1}X_s\Omega_s + H_{s,i}^{d}U\Omega_s + H_{s,i}^{st}E\Omega_s
\]

if \( A_s^{i-1} \) is neglected then:

\[
Y_{s,i} \simeq H_{s,i}^{d}U\Omega_s + H_{s,i}^{st}E\Omega_s
\]

since \( A_s \) is assumed to be asymptotically stable, the covariance matrix of the state sequence ( \( \lim_{j \to \infty} \frac{1}{j}(X_{s,i\Omega_s}(X_{s,i\Omega_s})^T) \)) is bounded. Note that, the elements of the weighting matrix \( \Omega_s \) take the values 0 and 1.

By the least squares method we can estimate the Markov parameter matrix \( H_{s,i}^{d} \):

\[
Y_{s,i}^{\dagger}(U\Omega_s)^T(U\Omega_s(U\Omega_s)^T)^{-1} \simeq H_{s,i}^{d}
\]