

A Recursive Sparse Learning Method: Application to Jump Markov Linear Systems

Dulin Chen, Laurent Bako and Stéphane Lecœuche

* Univ Lille Nord de France, F-59000 Lille, France

** EMDouai, IA, F-59500 Douai, France

Abstract: This paper addresses the problem of identifying linear multi-variable models from the input-output data which is corrupted by an unknown, non-centered, and sparse vector error sequence. This problem is sometimes referred to as error correcting problem in coding theory and robust estimation problem in statistics. By taking advantage of some recent developments in sparse optimization theory, we present here a recursive approach. We then show that the proposed identification method can be adapted to estimate parameter matrices of Jump Markov Linear Systems (JMLS), that is, switched linear systems in which the discrete state sequence is a stationary Markov chain. Some numerical simulation results illustrate the potential of the new method.

Keywords: Sparse estimation; Recursive identification; Error correction; Robust identification; Jump Markov Linear Systems (JMLS).

1. INTRODUCTION

In this paper, we consider the problem of identifying a linear multi-variable model, subject to sparse vector errors. More specifically, given a collection $\mathcal{D}_N = \{x(t), y(t)\}_{t=1}^N$ (with N possibly infinite) of data generated by a model of the following form,

$$y(t) = \mathbf{H}x(t) + f(t) + e(t), \quad (1)$$

the problem is to estimate the matrix $\mathbf{H} \in \mathbb{R}^{m \times n}$. Here, $y(t) \in \mathbb{R}^m$ is the output of the model, $x(t) \in \mathbb{R}^n$ is referred to as an explanatory variable and sometimes as a regressor vector; $\{e(t)\}$ is an unknown i.i.d noise sequence with a normal distribution $\mathcal{N}(0, \sigma_e^2 I_m)$.

In equation (1), the sequence $\{f(t)\} \subset \mathbb{R}^m$ is assumed to be an unknown but sparse sequence of possibly gross error vectors. By sparse sequence of vectors, it is meant here that the cardinality of the set $\{t : f(t) = \mathbf{0}\}$, is very large compared to the total number N of data. In other words, only a few number of vectors $f(t), t = 1, \dots, N$, are nonzero vectors. Those nonzero vectors in the sequence $\{f(t)\}$ can in principle have arbitrarily large magnitudes. Typically, the sequence $\{f(t)\}$ can be thought of as occasional errors occurring in data measuring devices. For example, it can be viewed as accounting for data transmission/reception errors over communication networks [1] or sensor/actuator faults in linear dynamical systems. A particularly interesting application of the above problem concerns, as will be suggested later in the paper, the identification of switched linear models. In this case, if one focuses on the problem of recovering a specific linear multi-variable submodel, then the data generated by the other submodels can be regarded as a sparse sequence of errors [2–4]. We will elaborate more on this point in Section 3.

The objective of this paper is to provide a *recursive* estimate of the matrix \mathbf{H} in model (1) as the data $\{x(t), y(t)\}$ are getting sequentially acquired in time. In doing so, no additional information other than that of sparsity is available on the vector sequence $\{f(t)\}$. As argued earlier in the paper,

this problem has numerous applications in various engineering fields [1, 5]. While *batch* sparse estimation has been extensively treated in the existing literature [6, 7], *recursive* sparse estimation is a problem that has not received much attention so far. And yet there are many practical situations where a recursive processing of the measurements is desirable. For example, when we are interested in monitoring on-line a physical process with the objective of detecting abrupt changes, the data are sequentially sampled, hence calling for a recursive treatment. Or when the number of data available for parameter identification is too large, it may be computationally difficult to process them all at once using batch algorithms. In such a situation, recursive estimation is an attractive alternative. Despite the relevance of the problem of recursive sparse estimation, only a few recent works brought it up [8, 9]. Note however that the solutions proposed in those papers appeared to be a recursive extension of the well-known LASSO algorithm [10] which applies to a different framework from the one of the present work. To emphasize the differences, note first that the data-generating model in (1) is a multi-variable one so that the unknown variable to be computed is a parameter matrix instead of a vector as is usually the case in related works [1, 5–7]. Second, while the LASSO seeks to estimate a parsimonious vector of parameters, the sparsity constraint is imposed here on a sequence of error vectors. This comes with an additional challenge specially when the data are also corrupted by zero-mean noise as in model (1). To the best of our knowledge, our work is the first one to address the problem of identifying a multi-variable model such as (1) (involving a sparse error sequence) in a recursive framework.

The contribution of this paper is twofold. In the first part, a new method is presented for solving the recursive sparse estimation problem stated above. Our approach consists in the optimization of an ℓ_2 -norm based cost function. For the sake of easy recursive update, we approximate the solution through a Weighted Recursive Least Squares (W-RLS) algorithm. The way the weights are selected is inspired by the idea used in the re-weighting ℓ_1 -minimization algorithm of [6] which, recall, applies to a different setting from the one treated here. The

second contribution of this work is an application of the method developed in the first part to the identification of Jump Markov Linear Systems (JMLS). If s denotes the number of subsystems composing the JMLS under consideration, we show that the submodels can, following a similar procedure as in [4], be recursively identified using s W-RLS identifiers operating in parallel.

The rest of this paper is organized as follows. The novel method is outlined in Section 2. In Section 3, we show that this method can be applied for the identification of JMLS from input-output data. Numerical simulation results are reported in Section 4. All of the simulation results, on linear models as well as on JMLS, reveal an excellent behavior of our algorithm. In the end, the conclusion and future work are discussed in Section 5.

2. THE RECURSIVE SPARSE LEARNING METHOD

In this section we propose a solution to the identification problem stated in the previous section. Recalling the multi-variable model (1), our goal is to identify the parameter matrix \mathbf{H} recursively, from the experimental measurements $\{x(t), y(t)\}$, which are corrupted by unknown noise and sparse error sequences, $\{e(t)\}$ and $\{f(t)\}$. For the sake of clarity in the presentation, we will assume that the noise sequence $\{e(t)\}$ is identically equal to zero¹.

2.1 The main idea

We start by emphasizing that no other prior information is available, except the sparsity condition on the error sequence $\{f(t)\}$. Standing on this point, the solution to our problem is to minimize the number of non-zero vectors in the error sequence $\{f(t)\}$, subject to (1). It is formulated as,

$$\begin{aligned} \min_{\mathbf{H}} |\{t : f(t) \neq 0\}|, \\ \text{subject to } y(t) = \mathbf{H}x(t) + f(t), t = 1, \dots, N \end{aligned} \quad (2)$$

where the notation $|\mathcal{S}|$ with \mathcal{S} representing a set, stands for the cardinality of the set \mathcal{S} . The optimization problem (2) attempts to find the matrix \mathbf{H} fitting the data record $\{(x(t), y(t))\}$ to model (1) and also making the error sequence $\{f(t)\}$ as sparse as possible. Sparsity is reinforced here by minimizing the cardinality of the set $\{t : f(t) \neq 0\}$. Noting that $f(t) \neq 0$ if and only if $\|f(t)\|_\ell \neq 0$, where ℓ stands for any norm, problem (2) can be reformulated as

$$\begin{aligned} \min_{\mathbf{H}} |\{t : \|f(t)\|_\ell \neq 0\}| \\ \text{subject to } y(t) = \mathbf{H}x(t) + f(t), t = 1, \dots, N. \end{aligned} \quad (3)$$

In principle, ℓ can be any type of norm. For example, one can take ℓ to be the 2-norm or the ∞ -norm. But in practice, the type of norm is of critical importance in the quality of the results one can hope for. By now defining

$$\bar{f} = (\|f(1)\|_\ell \dots \|f(N)\|_\ell)^\top, \quad (4)$$

problem (3) is equivalent to,

$$\begin{aligned} \min_{\mathbf{H}} \|\bar{f}\|_0 \\ \text{subject to } y(t) = \mathbf{H}x(t) + f(t), t = 1, \dots, N. \end{aligned} \quad (5)$$

Here and in the following, $\|z\|_0$ refers to the ℓ_0 -norm of z defined by $\|z\|_0 = |\{i : z(i) \neq 0\}|$ for any vector $z = (z(1) \dots z(N))^\top$. It is well known that an ℓ_0 -minimization

¹ However the method to be presented still works in the presence of noise. This is illustrated in Section 4.

problem as above is nonconvex and generally NP-hard, because its solution requires an intractable combinatorial search [6]. The idea of recasting this nonconvex optimization problem into a convex one has been developed in the past several decades [6, 7], by changing the ℓ_0 norm into the ℓ_1 norm in the objective function (5). Here, we pursue the same approach to obtain,

$$\begin{aligned} \min_{\mathbf{H}} \|\bar{f}\|_1 \\ \text{subject to } y(t) = \mathbf{H}x(t) + f(t), t = 1, \dots, N \end{aligned} \quad (6)$$

where $\|\bar{f}\|_1 = \sum_{t=1}^N \|f(t)\|_\ell$ is the ℓ_1 -norm of \bar{f} . From now on, we assume that $\ell = 2$. Note that we can then drop the constraints in the above problem and view it as the one of computing the parameter matrix \mathbf{H} by minimizing the cost function

$$J(\mathbf{H}) = \sum_{t=1}^N \|y(t) - \mathbf{H}x(t)\|_2. \quad (7)$$

It is important to notice that the cost function (7) consists in a sum of norms (on the fitting errors) instead of the usual sum of squared norms². This detail is crucial in promoting a solution \mathbf{H} such that the vector sequence $\{f(t) = y(t) - \mathbf{H}x(t)\}$ is sparse [11]. To see why, note that (7) can be rephrased as

$$J(\mathbf{H}) = \sum_{t=1}^N \omega(t) \|y(t) - \mathbf{H}x(t)\|_2^2, \quad (8)$$

where the weight $\omega(t) = 1/\|y(t) - \mathbf{H}x(t)\|_2$ has been introduced. In fact, $\omega(t) = 1/\|f(t)\|_2$ if $f(t) \neq 0$ and $\omega(t) = \infty$ if $f(t) = 0$. Here the cost function (8) shows the exact behavior of the objective function (7). Comparing (8) with the standard least squares, we can see that the objective (7) (or, (8)) behaves as if infinite weights were assigned to the fitting errors for which $f(t) = 0$. Hence this cost function tends to assign an infinitely high contribution to the errors for which $f(t) = 0$ and to annihilate the effects of the errors for which $f(t) \neq 0$. As a consequence, provided the cardinality of the set $\{t : f(t) \neq 0\}$ is reasonably small, minimizing (7) may result in the exact matrix \mathbf{H} , despite the presence of the sparse error vector sequence.

In practice, one can compute a minimizer of (7) by considering the objective (8). This latter cost function (8) would correspond simply to a weighted least squares problem if $\omega(t)$ were known for any t . Unfortunately, this is not the case here since the weights $\omega(t)$ depend on the unknown matrix \mathbf{H} . In the following, an approximate solution is computed in a recursive framework.

2.2 Derivation of the recursive estimator

As already emphasized before, our goal here is to obtain the matrix \mathbf{H} in a recursive fashion. To this end, denote with $\hat{\mathbf{H}}(0)$ the initial value assigned to the estimate of \mathbf{H} and let $\mathbf{P}(0) \in \mathbb{R}^{n \times n}$ be a positive-definite matrix. Define $\hat{\mathbf{H}}(t)$ to be the estimate of \mathbf{H} at time t based on the data pairs $(x(k), y(k))$ sampled up to time t . $\hat{\mathbf{H}}(t)$ can be viewed as the matrix $\mathbf{H}(t)$ that minimizes the cost function

$$\begin{aligned} J(\mathbf{H}(t)) = \sum_{k=1}^t \lambda^{t-k} \omega(k) \|y(k) - \mathbf{H}(t)x(k)\|_2^2 \\ + \lambda^t \text{tr} [(\mathbf{H}(t) - \mathbf{H}(0))^\top \mathbf{P}^{-1}(0)(\mathbf{H}(t) - \mathbf{H}(0))] \end{aligned} \quad (9)$$

² The sum of squared norms correspond to the ordinary least squares.

where $\lambda \in]0, 1]$ is a forgetting factor. Here, the weights are approximated as

$$\omega(k) = \frac{1}{\|y(k) - \hat{\mathbf{H}}(k-1)x(k)\|_2 + \eta}, \quad (10)$$

with η being a small positive number. With this approximation, the weight sequence $\{\omega(k)\}_{k=1}^t$ is now completely known. As a result, the criterion defined in (9)-(10) can be regarded as a Weighted-Recursive Least Squares (W-RLS) criterion. As is well known, the matrix $\hat{\mathbf{H}}(t)$ that minimizes (9) can be computed from $\hat{\mathbf{H}}(t-1)$ by applying the following update rules [12]:

$$K(t) = \frac{\omega(t)\mathbf{P}(t-1)x(t)}{\lambda + \omega(t)x^\top(t)\mathbf{P}(t-1)x(t)} \quad (11)$$

$$\hat{\mathbf{H}}(t) = \hat{\mathbf{H}}(t-1) + (y(t) - \hat{\mathbf{H}}(t-1)x(t))K^\top(t) \quad (12)$$

$$\mathbf{P}(t) = \frac{1}{\lambda}(\mathbf{P}(t-1) - K(t)x^\top(t)\mathbf{P}(t-1)), \quad (13)$$

where $\omega(t)$ is defined as in (10), the initial values $\hat{\mathbf{H}}(0)$ can be chosen randomly, and $\mathbf{P}(0) = \alpha\mathbf{I}_n$ with $\alpha \gg 1$, \mathbf{I}_n is an identity matrix with the dimension of n .

3. APPLICATION TO JUMP MARKOV LINEAR SYSTEMS

In the previous section, we presented a method for identifying the parameter matrix recursively from a collection of measurements which are corrupted by unknown sparse error and white noise sequences. In this section, we will show that this method can be interestingly applied to the identification of Jump Markov Linear Systems (JMLS).

A JMLS corresponds to the behavior that results from switchings among a set of linear subsystems, the transitions between subsystems being determined by a finite state Markov chain. JMLS can be represented by

$$\begin{cases} x(t+1) = \mathbf{A}_{\sigma(t)}x(t) + \mathbf{B}_{\sigma(t)}u(t) \\ y(t) = \mathbf{C}_{\sigma(t)}x(t) + \mathbf{D}_{\sigma(t)}u(t) \end{cases}, \quad (14)$$

where $x(t) \in \mathbb{R}^{n_x}$ stands for the continuous state, with n_x representing the order of the system (i.e, the dimension of the state space). This order is assumed to be the same for all of the submodels. $u(t) \in \mathbb{R}^{n_u}$, $y(t) \in \mathbb{R}^{n_y}$, $\sigma(t) \in S = \{1, \dots, s\}$ are respectively the input, the output and the discrete state (which will be also called the mode). $\mathbf{A}_{\sigma(t)} \in \mathbb{R}^{n_x \times n_x}$, $\mathbf{B}_{\sigma(t)} \in \mathbb{R}^{n_x \times n_u}$, $\mathbf{C}_{\sigma(t)} \in \mathbb{R}^{n_y \times n_x}$, $\mathbf{D}_{\sigma(t)} \in \mathbb{R}^{n_y \times n_u}$ are the parameter matrices associated with the discrete state $\sigma(t)$. It is assumed that $\sigma(t)$, $t \geq 1$, is a discrete-time, first-order stationary Markov chain with constant transition probabilities defined as,

$$\pi_{ij} = \mathbb{P}\{\sigma(t) = i | \sigma(t-1) = j\} \quad (15)$$

for any $i, j \in S = \{1, 2, \dots, s\}$. Hence, the transition probability matrix $\boldsymbol{\pi}$ is an $s \times s$ constant matrix, with the components satisfying $\pi_{ij} \geq 0$ and $\sum_{i=1}^s \pi_{ij} = 1$, for each $j \in S$. The initial probability distributions are denoted as,

$$\pi_j^0 = \mathbb{P}\{\sigma(0) = j\} \quad (16)$$

for all $j \in S$ and are such that $\pi_j^0 \geq 0$, and $\sum_{j=1}^s \pi_j^0 = 1$.

Given the measurements sequence³ $\{x(t), u(t), y(t)\}_{t=1}^N$, our objective is to identify the parameter matrices $\mathbf{A}_j, \mathbf{B}_j, \mathbf{C}_j,$

³ The continuous state $x(t)$ is assumed completely measurable while the discrete state $\sigma(t)$ is unknown.

$\mathbf{D}_j, j \in S$, as well as the transition probabilities defined in (15). Moreover, all of the identification process will be *on-line* in our work.

Let us set

$$\bar{x}(t) = (x(t)^\top u(t)^\top)^\top \quad (17)$$

$$\mathbf{H}_{\sigma(t)} = \begin{pmatrix} \mathbf{A}_{\sigma(t)} & \mathbf{B}_{\sigma(t)} \\ \mathbf{C}_{\sigma(t)} & \mathbf{D}_{\sigma(t)} \end{pmatrix} \quad (18)$$

$$\bar{y}(t) = (x(t+1)^\top y(t)^\top)^\top. \quad (19)$$

Then we can rewrite the model (14) as,

$$\bar{y}(t) = \mathbf{H}_{\sigma(t)}\bar{x}(t). \quad (20)$$

The application of the previous method to the estimation of the matrices $\{\mathbf{H}_i\}_{i=1}^s$ from model (20) is based on the following observation. For any $i \in S$, model (20) can be rewritten as

$$\begin{aligned} \bar{y}(t) &= \mathbf{H}_i\bar{x}(t) + (\mathbf{H}_{\sigma(t)} - \mathbf{H}_i)\bar{x}(t) \\ &= \mathbf{H}_i\bar{x}(t) + f'_i(t), \end{aligned} \quad (21)$$

with $f'_i(t) = (\mathbf{H}_{\sigma(t)} - \mathbf{H}_i)\bar{x}(t)$. Now by assuming that the discrete state i is visited by the JMLS frequently enough, we can see that the vector sequence $\{f'_i(t)\}$ in (21) is sparse since $f'_i(t) = 0$ whenever $\sigma(t) = i$. Therefore, if we focus on the identification of a specific submodel with index i , we get a similar model as (1), under noise-free assumption. Standing on this point, we can adapt the method proposed in the previous section to identify the parameter matrices of the JMLS model (20). In fact, if there exists one submodel i of the JMLS that is activated very often over time, then running directly the W-RLS identifier of Section 2 on the data generated by model (20) may yield the corresponding matrix \mathbf{H}_i . However, this may not be the case in practice. For example, when the number of submodels is relatively large, it may happen that none of the submodels is frequently activated.

To deal with this type of situation, the recursive sparse estimation method can be applied efficiently as follows. We estimate all the submodels simultaneously by running s different recursive identifiers in parallel, one identifier for each submodel. At each time t , the discrete state $\sigma(t)$ is estimated as the index of the submodel which, based on the prior estimates of the different parameter matrices, appears to be the most consistent with the data pair $(\bar{x}(t), \bar{y}(t))$. Then the submodel associated with the estimated discrete state has to be updated. To see why this procedure is likely to yield the sought matrices \mathbf{H}_i , $i \in S$, note that each of the s recursive sparse identifiers has to correct only wrong mode assignments (which are expected to be rare). Therefore, thanks to the error correction ability of the identifiers, this algorithm is, as will be confirmed by simulations results, able to recover the JMLS parameters.

In more details, we follow a similar idea as in [4, 13] and select $\hat{\sigma}(t)$ to coincide with the index of the matrix $\hat{\mathbf{H}}_i(t-1)$, $i \in S$ that minimizes the relative fitting error i.e,

$$\hat{\sigma}(t) = \arg \min_{i \in S} \frac{\|\bar{y}(t) - \hat{\mathbf{H}}_i(t-1)\bar{x}(t)\|_2}{\|(\mathbf{I}_{n_x+n_y} \hat{\mathbf{H}}_i(t-1))\|_2}, \quad (22)$$

where $\hat{\mathbf{H}}_i(t-1)$ is the estimated matrix associated with the discrete state $i \in S$ at $t-1$; $\mathbf{I}_{n_x+n_y}$ is an identity matrix of dimension $n_x + n_y$.

In the following part of this section, we are going to focus on how to estimate the transition probabilities *on-line*. Define

two sets $\widehat{Z}_{ij}(t)$ and $\widehat{Z}_i(t)$ as follows: $\widehat{Z}_{ij}(t)$ records the time indices when the system switched from submodel j to submodel i while $\widehat{Z}_i(t)$ records the time indices when the system is switched into submodel i from any submodel. More specifically, let

$$\widehat{Z}_{ij}(t) = \{1 < k \leq t | \widehat{\sigma}(k) = i, \widehat{\sigma}(k-1) = j\}, \quad (23)$$

$$\widehat{Z}_i(t) = \{1 < k \leq t | \widehat{\sigma}(k) = i\}. \quad (24)$$

Then, the transition probability from state j to state i can be evaluated as

$$\widehat{\pi}_{ij}(t) = \frac{|\widehat{Z}_{ij}(t)|}{|\widehat{Z}_i(t)|}. \quad (25)$$

Here $\widehat{\pi}_{ij}(t)$ is the estimated transition probability at time t ; $|\widehat{Z}_{ij}(t)|$ and $|\widehat{Z}_i(t)|$ represent the cardinality of the sets $\widehat{Z}_{ij}(t)$ and $\widehat{Z}_i(t)$ respectively. Furthermore, we can derive the following update rule for the transition probability,

$$\widehat{\pi}_{ij}(t) = \frac{|\widehat{Z}_i(t-1)|}{|\widehat{Z}_i(t-1)| + \widehat{\delta}_i(t)} \widehat{\pi}_{ij}(t-1) + \frac{\widehat{\delta}_i(t)\widehat{\delta}_j(t-1)}{|\widehat{Z}_i(t-1)| + \widehat{\delta}_i(t)} \quad (26)$$

where

$$\widehat{\delta}_q(t) = \begin{cases} 1 & \text{if } \widehat{\sigma}(t) = q \\ 0 & \text{if } \widehat{\sigma}(t) \neq q. \end{cases} \quad (27)$$

Finally, we summarize in Table 1 the complete algorithm for recursively identifying the parameter matrices \mathbf{H}_i , $i \in S$, along with the transition probabilities $\pi_{ij}(t)$ of the JMLS model.

Table 1. Algorithm for Identifying JMLS.

Algorithm for Identifying JMLS
Initialization:
Provide the matrices $\widehat{\mathbf{H}}_i(0)$, the covariance matrices $\mathbf{P}_i(0)$, $i \in S$, as well as the transition probabilities $\widehat{\pi}_{ij}(0)$, $(i, j) \in S^2$.
For any incoming data pair $(\widehat{x}(t), \widehat{y}(t))$:
(1) Select the activated discrete state $\widehat{\sigma}(t)$ as in (22).
(2) Update the parameter matrix associated with the activated discrete state $\widehat{\sigma}(t)$ by (11)-(13). For any i such that $i \neq \widehat{\sigma}(t)$, $\widehat{\mathbf{H}}_i(t) = \widehat{\mathbf{H}}_i(t-1)$.
(3) Update the transition probability matrix using (26).

4. NUMERICAL RESULTS

In this section we illustrate the whole identification procedure on synthetic data generated by a model with the form (1) and a JMLS model respectively.

4.1 Identification results for a model of the form (1)

Firstly, we test our identification method on the data generated by a linear multi-variable model of the type (1). With the dimensions $m = 3$ and $n = 3$, the ‘‘true’’ model that generates the data is defined as follows:

$$y(t) = \begin{pmatrix} 1.5847 & 0.3944 & -0.8044 \\ -0.6355 & 0.6342 & -0.2397 \\ -0.1957 & -0.9011 & -0.7516 \end{pmatrix} x(t) + f(t) + e(t). \quad (28)$$

Here, the input data sequence $\{x(t)\}$ is generated randomly following a normal distribution, $\mathcal{N}(0, I_n)$. The sparse error

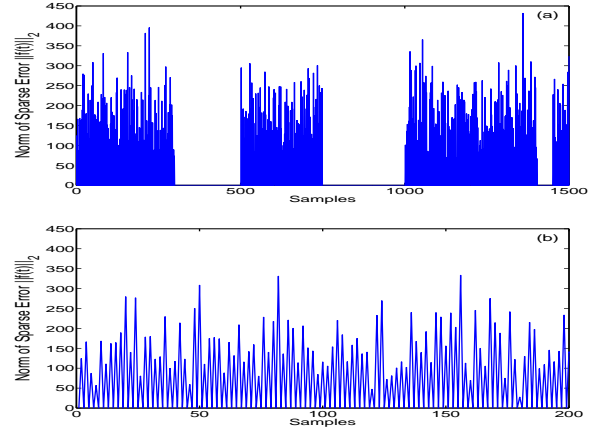


Figure 1. ℓ_2 norm $\|f(t)\|_2$ of the sparse error sequence in model (28): (a) over 1500 samples; (b) over the first 200 samples.

sequence $\{f(t)\}_{t=1}^N$, with $N = 1500$, is chosen such that 1000 vectors out of 1500 are strictly equal to zero. The remaining error vectors are generated from a non-centered normal distribution, $\mathcal{N}(\bar{\mu}_1, \sigma_1^2 I_m)$, $\bar{\mu}_1 = 5(1, \dots, 1)^\top$, $\sigma_1^2 = 100$. For an overview on the temporal distribution of this sparse error sequence, we plot its ℓ_2 -norm over 1500 samples in Fig.1.(a) and over the first 200 samples in Fig.1.(b). Notice that the ℓ_2 norm of non-zero sparse error, as reported in Fig. 1, can be arbitrarily large as stated before. The noise $\{e(t)\}$ in (28) is generated as a Gaussian white noise so as to achieve a Signal-to-Noise-Ratio (SNR) of 30 dB.

Our learning method is implemented in MATLAB. Related parameters are given as follows. The forgetting factor is set to be 0.95. The positive number η appearing in (10) should, in principle, be extremely small as its role consists only in preventing division by zero. We set η to be $1e-6$. The initial covariance matrix is chosen as, $\mathbf{P}(0) = \alpha \mathbf{I}_3$, with $\alpha = 10$. And $\widehat{\mathbf{H}}(0)$ is a zero matrix.

The mean and the standard deviation of estimated values for 100 independent simulations with SNR=30 dB, are shown in Table 2. Compared to the true value, these numerical results show that our method achieves a good estimate for the parameter matrix of model (28).

Table 2. Empirical mean and standard deviation of the estimated $\widehat{\mathbf{H}}$ for model (28), with an SNR=30 dB.

Mean	Standard Deviation
$\begin{pmatrix} 1.5846 & 0.3942 & -0.8046 \\ -0.6355 & 0.6337 & -0.2392 \\ -0.1956 & -0.9009 & -0.7516 \end{pmatrix}$	$\pm \begin{pmatrix} 0.0051 & 0.0056 & 0.0051 \\ 0.0061 & 0.0044 & 0.0043 \\ 0.0047 & 0.006 & 0.0047 \end{pmatrix}$

To further analyze the properties of the proposed method, we define a set of relative errors $\{\varepsilon(t)\}_{t=1}^N$ to assess the quality of the estimates over time,

$$\varepsilon(t) = \frac{\|\widehat{\mathbf{H}}(t) - \mathbf{H}\|_F}{\|\mathbf{H}\|_F}, \quad (29)$$

where $\|\cdot\|_F$ represents the *Frobenius* norm. We apply both the standard RLS algorithm and the new method proposed in Section 2 to solve the identification problem. During all the

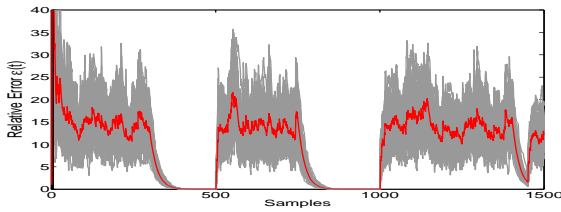


Figure 2. Trajectories of the relative errors $\varepsilon(t)$ over time when standard RLS is applied to model (28): SNR=30 dB.

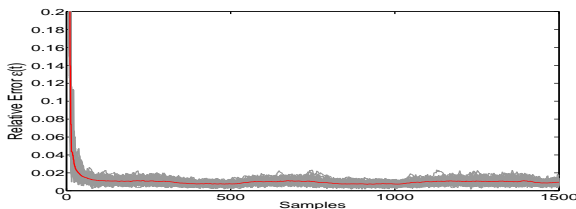


Figure 3. Trajectories of the relative errors $\varepsilon(t)$ over time when our method is applied to model (28): SNR=30 dB.

100 simulations, the trajectories of the relative errors $\{\varepsilon(t)\}_{t=1}^N$ varying over time are reported in Fig.2 for the RLS and in Fig.3 for our method. In these figures, the red bold lines are the averages of the relative errors over the 100 simulations. Fig.2 shows clearly that the standard RLS fails to provide correct estimates. Note that convergence occurs for that algorithm only over the time intervals on which the error sequence $\{f(t)\}$ is equal to zero. In contrast, we can see from Fig.3 that, despite the presence of the error $\{f(t)\}$, the method of this paper achieves accurate estimates. In all the simulations carried out, the relative errors drop off rapidly during the first dozens of steps, and the algorithm almost converges after 80 steps. Hence, the proposed method successfully recovers the parameter matrix, although the sparse error sequence is completely unknown and has a very large magnitude. Note that the small bias which is visible in Fig.3 is induced only by the noise $\{e(t)\}$. This is confirmed by the results of Table 3, where the final estimation error $\varepsilon(N)$ is shown to be strictly equal to zero when $\{e(t)\}$ is identically null.

In order to analyze the robustness of our method with respect to noise, we test it on model (28) with different noise levels. In this experiment, the sparse error sequence $\{f(t)\}$ has the profile given in Fig.1. For each noise level, 100 simulations with $N = 1500$ data pairs each, are carried out. The average of the final relative errors (that is, the error after convergence) over all of the simulations are given in Table 3. As already mentioned, when there is no noise (that is, when $\text{SNR} = \infty$), the relative error is equal to zero. And the relative errors logically increase when the SNR decreases.

Now, we propose to verify up to what extent of sparsity of the sequence $\{f(t)\}$ our method is able to identify model (28). To this end, we test it for different sparsity levels of $\{f(t)\}$. Here, the sparsity ρ is defined as the number of the non-zero vectors divided by the number of data, i.e.,

$$\rho = \frac{\|\bar{f}\|_0}{N} \times 100\%, \quad (30)$$

where \bar{f} is defined as in (4). For this sparsity measure to make sense in the present recursive context, the temporal distribution of nonzero vectors in the sequence $\{f(t)\}$ needs to be uniform.

The conditions of our experiment are set so as to meet this requirement. The non-zero vectors in the sequence $\{f(t)\}$ are generated randomly from the non-centered normal distribution $\mathcal{N}(\bar{\mu}_2, \sigma_2^2 I_m)$, $\bar{\mu}_2 = 5(1, \dots, 1)^\top$, $\sigma_2^2 = 100$. For every sparsity level, 100 simulations are carried out with $N = 1500$ pairs of noisy data (the SNR being equal to 30 dB). The average of the final relative errors over all of the simulations for each sparsity level are reported in Table 4. It shows that our method can perform dramatically well even for a level of sparsity as high as 90%.

Judging from all of the results displayed in this subsection, we are prompted to conclude that the proposed method is reliable for identifying a model of the form (1), which is subject to occasional gross errors.

4.2 Identification results for a JMLS model

Finally, we apply our algorithm to identify JMLS parameter matrices, as well as the transition probabilities. We take an example of JMLS composed of three submodels, i.e., the discrete state $\sigma(t)$ lives in $S = \{1, 2, 3\}$. With the dimensions $n_x = 2$, $n_u = 1$ and $n_y = 1$, all of the “true” parameter matrices are presented in Table 5, together with the transition probability matrix. The matrices \mathbf{H}_i are defined as in (18) for all $i \in S$.

Table 5. “True” JMLS parameter matrices.

Matrix	“True” Value
\mathbf{H}_1	$\begin{pmatrix} -0.7901 & -0.0458 & -1.0834 \\ -0.0458 & -0.7544 & 0.4504 \\ -2.8025 & 0 & 0 \end{pmatrix}$
\mathbf{H}_2	$\begin{pmatrix} -0.5742 & -0.3092 & 0.6379 \\ -0.3092 & 0.3868 & -2.3072 \\ -1.5833 & -1.0135 & -1.0293 \end{pmatrix}$
\mathbf{H}_3	$\begin{pmatrix} -0.6389 & -0.1516 & 0.9488 \\ -0.1516 & -0.5814 & 0.7786 \\ 0.0005 & 0 & 0 \end{pmatrix}$
π	$\begin{pmatrix} 0.2818 & 0.2594 & 0.4589 \\ 0.2500 & 0.2606 & 0.4867 \\ 0.2687 & 0.2410 & 0.4917 \end{pmatrix}$

The input sequence $\{u(t)\}_{t=1}^N$, with $N = 1500$ for each simulation here, is generated randomly following the normal distribution, $\mathcal{N}(0, I_{n_u})$. $x(0)$ is also given as a random value. A certain amount of Gaussian white noise is added to the output data, so that we get an SNR of 30 dB. The user-defined parameters of the learning method are chosen to be the same as in the previous subsection. And for $i \in S$, the initial parameter matrices $\mathbf{H}_i(0)$, are drawn at random; $\mathbf{P}_i(0) = \alpha_i \mathbf{I}_3$, with $\alpha_i = 10$. The numerical average and standard deviation of the estimated parameter matrices over 100 simulations are reported in Table 6. It can be noticed that there is a little bias appearing in the result. This is due to the fact that the data are affected by the noise, as argued in Subsection 4.1.

Moreover, all of the simulation examples have already shown the effectiveness of the proposed Weighted-RLS. The results reveal that the method proposed in this paper is feasible and efficient in solving identification problem of JMLS.

5. CONCLUSION

In this paper we treated the problem of identifying on-line a linear multi-variable model, subject to a sparse vector error

Table 3. Relative Errors when our method is applied to model (28) with different noise levels.

SNR (dB)	∞	50	40	30	20	10	5
$\varepsilon(N)$	0	0.0012	0.0045	0.0144	0.0449	0.1293	0.2592

Table 4. Relative Errors when our method is applied to model (28) with different sparsity levels—SNR=30 dB.

sparsity ρ (%)	20	30	40	50	70	80	90	95	96
$\varepsilon(N)$	0.0088	0.0092	0.0102	0.0108	0.0133	0.0165	0.0236	0.1806	13.1917

Table 6. Empirical mean and standard deviation of estimated values for JMLS: SNR=30 dB.

Estimated Matrix	Mean	Standard Deviation
$\hat{\mathbf{H}}_1 :$	$\begin{pmatrix} -0.7901 & -0.0458 & -1.0834 \\ -0.0458 & -0.7544 & 0.4504 \\ -2.8051 & 0.0034 & -0.0052 \end{pmatrix}$	$\pm (1.0e-0) \times \begin{pmatrix} 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 \\ 0.0543 & 0.0305 & 0.0318 \end{pmatrix}$
$\hat{\mathbf{H}}_2 :$	$\begin{pmatrix} -0.5744 & -0.3013 & 0.6243 \\ -0.3041 & 0.3636 & -2.2492 \\ -1.5617 & -0.9812 & -1.0114 \end{pmatrix}$	$\pm (1.0e-0) \times \begin{pmatrix} 0.0428 & 0.0554 & 0.1512 \\ 0.0506 & 0.1627 & 0.4068 \\ 0.1564 & 0.2324 & 0.1352 \end{pmatrix}$
$\hat{\mathbf{H}}_3 :$	$\begin{pmatrix} -0.6389 & -0.1516 & 0.9488 \\ -0.1516 & -0.5814 & 0.7786 \\ 0.0005 & 0.0000 & -0.0000 \end{pmatrix}$	$\pm (1.0e-5) \times \begin{pmatrix} 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 \\ 0.6261 & 0.2900 & 0.1770 \end{pmatrix}$
$\hat{\boldsymbol{\pi}} :$	$\begin{pmatrix} 0.301 & 0.2129 & 0.4854 \\ 0.2688 & 0.2635 & 0.4654 \\ 0.2939 & 0.2347 & 0.471 \end{pmatrix}$	$\pm (1.0e-0) \times \begin{pmatrix} 0.0593 & 0.0844 & 0.0864 \\ 0.0764 & 0.0552 & 0.0899 \\ 0.0802 & 0.0713 & 0.0651 \end{pmatrix}$

sequence. This is an important and generic problem having applications in many engineering fields. For example, as illustrated in the paper, the identification problem for switched linear systems can be viewed as a particular instance of that problem. We showed that by appropriately setting some weights, a simple recursive weighted least squares algorithm can be used to efficiently address this problem. Although the effectiveness of the proposed method has been extensively verified on some numerical examples, convergence analysis is still lacking in this first work. This will be considered in future research.

REFERENCES

- [1] E. J. Candès and P. Randall, "Highly robust error correction by convex programming," *IEEE Transactions on Information Theory*, vol. 54, pp. 2829–2840, 2008.
- [2] N. Ozay, M. Sznaiar, C. M. Lagoa, and O. I. Camps, "A sparsification approach to set membership identification of a class of affine hybrid systems," in *IEEE Conference on Decision and Control, Cancun, Mexico*, pp. 123–130, 2008.
- [3] L. Bako, "Identification of switched linear systems via sparse optimization," *Automatica*, vol. 47, pp. 668–677, 2010.
- [4] L. Bako, K. Boukharouba, E. Duviella, and S. Lecoeuche, "A recursive identification algorithm for switched linear/affine models," *Nonlinear Analysis: Hybrid Systems*, doi:10.1016/j.nahs.2010.05.003, 2010.
- [5] E. J. Candès, "Compressive sampling," in *Proceedings of the International Congress of Mathematicians*, (Madrid, Spain), 2006.
- [6] E. J. Candès, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted l_1 minimization," *The Journal of Fourier Analysis and Applications*, vol. 14, pp. 877–905, 2008.
- [7] D. Donoho, "For most large underdetermined systems of linear equations, the minimal ℓ^1 -norm solution is also the sparsest solution," *Communications on pure and applied mathematics*, vol. 59, pp. 797–829, 2006.
- [8] L. Blackhall and M. Rotkowitz, "Recursive sparse estimation using a gaussian sum filter," in *Proceedings of the 17th IFAC World Congress, Seoul, Korea*, 2008.
- [9] D. Angelosante and G. B. Giannakis, "RLS-weighted lasso for adaptive estimation of sparse signals," *IEEE International Conference on Acoustics, Speech and Signal Processing, Taipei, Taiwan*, 2009.
- [10] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of Royal Statistical Society*, vol. 58, pp. 267–288, 1996.
- [11] H. Ohlsson, L. Ljung, and S. Boyd, "Segmentation of arx-models using sum-of-norms regularization," *Automatica*, vol. 46, pp. 1107–1111, 2010.
- [12] S.M.Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Inc. Upper Saddle River, NJ, USA, 1993.
- [13] L. Bako, G. Mercère, R. Vidal, and S. Lecoeuche, "Identification of switched linear state space models without minimum dwell time," in *IFAC Symposium on System Identification*, (Saint Malo, France), 2009.