

A CLASS OF REAL-TIME AR IDENTIFICATION ALGORITHMS IN THE CASE OF MISSING OBSERVATIONS

Sina Mirsaidi and Jacques Oksman
 École Supérieure d'Électricité - Service des Mesures
 Plateau de Moulon, 91192 Gif-sur-Yvette France
 Mirsaidi@soleil.supelec.fr, Oksman@supelec.fr

ABSTRACT

This paper deals with the problem of adaptive AR estimation from incomplete observations. The method is based on the optimization of a weighted squared error criterion. Various approximates of this criterion lead to different algorithms. The formal description of these algorithms are given and their performances in stationary and non-stationary environments are compared.

1 INTRODUCTION

A common problem in time series analysis is that of parametric estimation from a set of uniformly spaced measurements. Efficient estimates can be achieved by parametric estimation techniques such as autoregressive (AR) or autoregressive moving-average (ARMA) modeling. In practice it often happens that some of the measurements are missing or the sampling is irregular. Most of the work on the missing data problem has been dealt with various non parametric techniques. The book by Parzen [1] brought together valuable references in this field and some other parametric techniques may be found in [2][3][4]. Most of the cited works can be used in block-mode operations and are not well suited for adaptive identification contexts.

Recently, we proposed an adaptive method of AR estimation for irregularly observed time series (or time series with missing data) [5] [6]. It essentially extends the gradient based adaptive modeling technique to the non uniform sampling case. The AR parameters are obtained by minimizing a cost function that is defined as a weighted mean squared error. This cost function has been shown to be unimodal, at least in some special cases [7][8]. A new adaptive data compression technique based on the mentioned method was proposed in [9].

In this paper, we propose a class of real-time algorithms based on different approximations of the described cost function. The choice of each algorithm depends on the application and is a compromise between complexity, robustness and tracking capacity. In what follows, we describe briefly the developpement of each algorithm. The performance of each of them is then illustrated by some numerical examples.

2 DESCRIPTION OF METHOD

The original signal $\{y_n\}$ is modeled by an autoregressive process as :

$$y_n = \boldsymbol{\theta}^T \mathbf{y}_n + v_n \quad (1)$$

where $\boldsymbol{\theta}^T = [\theta_1, \dots, \theta_M]$ is the AR vector parameters and $\mathbf{y}_n^T = [y_{n-1}, \dots, y_{n-M}]$ represents the vector of M last signal samples, M being the model order. $\{v_n\}$ is a zero mean white noise process with variance σ_v^2 . We suppose that the sampling is non uniform. Both regular and random pattern of misses can be handled. For convenience and without loss of generality, we assume unity spacing of samples prior to modification of the sampling train. Let $\tau = \{t_1, \dots, t_n\}$ represent the sampling scheme where $t_1 < t_2 \dots < t_n$. It is obvious that $t_i - t_{i-1} \geq 1$ for $i = 2, \dots, n$. We define the cost function as the mean squared value of the prediction error at sampling instants :

$$J_{t_i} = E(\omega_{t_i}(y_{t_i} - \hat{y}_{t_i})^2) \quad (2)$$

The problem is to find the vector of parameters $\boldsymbol{\theta}$ that minimizes J_{t_i} .

In order to evaluate the AR parameters, we suppose that the model is estimated at some instant t_i . At each following instant, if the sample is lost, we predict it by means of the last estimated AR parameters. Otherwise, we use the new sample and its estimate to update the model parameters. In order to obtain the estimate of the signal sample at instant t_i , we use the following equation :

$$\hat{y}_i = \boldsymbol{\theta}^T \mathbf{h}_i \quad (3)$$

where the elements of the $M \times 1$ vector $\mathbf{h}_i^T = [h_{i-1}, \dots, h_{i-M}]$ are defined as below :

$$h_j = \begin{cases} y_j & \text{if } y_j \text{ is not lost} \\ \hat{y}_j & \text{otherwise} \end{cases} \quad (4)$$

It is well known that if samples y_{i-1}, \dots, y_{i-M} are not lost, Eq. (3) represents the optimal linear predictor in

the mean square sense. A brief description of three algorithms based on this method is shown below. Details of the mathematical rules for obtaining the equations may be found in [6][8] and are not given here for simplicity.

Algorithm # 1 We approximate the cost function by the instantaneous estimation error. Therefore:

$$J_{t_i} = (y_{t_i} - \hat{y}_{t_i})^2 \quad (5)$$

$$= (y_{t_i} - \boldsymbol{\theta}^T \mathbf{h}_{t_i})^2$$

The only difference with the case where the measurements are contiguous is that \mathbf{h}_{t_i} is a function of $\boldsymbol{\theta}$. The gradient $\mathbf{g}_{t_i} = \frac{\partial J_{t_i}}{\partial \boldsymbol{\theta}}$ is obtained as below :

$$\mathbf{g}_{t_i} = -2(y_{t_i} - \boldsymbol{\theta}^T \mathbf{h}_{t_i}) \frac{\partial \boldsymbol{\theta}^T \mathbf{h}_{t_i}}{\partial \boldsymbol{\theta}} \quad (6)$$

$$= -2e_{t_i} \frac{\partial \hat{y}_{t_i}}{\partial \boldsymbol{\theta}}$$

where e_{t_i} is the estimation error. It can be shown that the term $\frac{\partial \hat{y}_{t_i}}{\partial \boldsymbol{\theta}}$ can be obtained by the following equation :

$$\frac{\partial \hat{y}_{t_i}}{\partial \boldsymbol{\theta}} = \mathbf{h}_{t_i} + \left[\frac{\partial \hat{y}_{t_i-1}}{\partial \boldsymbol{\theta}} \mid \dots \mid \frac{\partial \hat{y}_{t_i-M}}{\partial \boldsymbol{\theta}} \right] \boldsymbol{\theta} \quad (7)$$

Eq. (7) is true for *all* instants $j = 1, 2, \dots, t_n$. We obtain in this way a recursive algorithm for updating the AR parameters :

$$\boldsymbol{\theta}_{t_i} = \boldsymbol{\theta}_{t_{i-1}} - \mu \mathbf{g}_{t_i} \quad (8)$$

where μ is the step-size parameter and \mathbf{g}_{t_i} is obtained using Eqs. (6)-(7). It is important to note that the vector $\boldsymbol{\theta}_{t_i}$ is updated only when a new sample arrives. That means, Eq. 8 is used at instants $t_i, i = 1, \dots, n$, but recursion (7) must be used at *all* instants and it does not depend on the presence or absence of samples.

Algorithm # 2 The cost function is approximated by the sum of L past squared estimation errors. This is in fact a finite memory version of the cost function with a memory length L :

$$J_{t_i} = \frac{1}{L} \sum_{i=t_{i-L}+1}^{t_i} (y_{t_i} - \hat{y}_{t_i})^2 \quad (9)$$

we adopt the following definitions :

$$\mathbf{y}_{t_{i+1-L}}^T = \left[y_{t_{i+1-L}}, \dots, y_{t_{i+1}} \right] \quad (10)$$

$$H_{t_{i+1-L}}^T = \left[\mathbf{h}_{t_{i+1-L}} \mid \dots \mid \mathbf{h}_{t_{i+1}} \right] \quad (11)$$

Using these matrix and vector notations Eq. (9) can be rewritten as :

$$J_{t_{i+1,L}} = \frac{1}{L} \left(\mathbf{y}_{t_{i+1,L}} - H_{t_{i+1,L}} \boldsymbol{\theta} \right)^T \left(\mathbf{y}_{t_{i+1,L}} - H_{t_{i+1,L}} \boldsymbol{\theta} \right) \quad (12)$$

By differentiating the cost function with respect to $\boldsymbol{\theta}$ and rewriting Eq. (8), we obtain the following updating equations :

$$\begin{cases} \boldsymbol{\theta}_{t_{i+1}} = \boldsymbol{\theta}_{t_i} - \frac{\mu}{L} \mathbf{g}_{t_{i+1,L}} \\ \mathbf{g}_{t_{i+1,L}} = \frac{\partial J_{2,t_{i+1,L}}^T}{\partial \boldsymbol{\theta}} - 2 \frac{\partial J_{1,t_{i+1,L}}^T}{\partial \boldsymbol{\theta}} \end{cases} \quad (13)$$

where :

$$J_{1,t_{i+1,L}} = \mathbf{y}_{t_{i+1,L}}^T H_{t_{i+1,L}} \boldsymbol{\theta}$$

$$J_{2,t_{i+1,L}} = \boldsymbol{\theta}^T H_{t_{i+1,L}}^T H_{t_{i+1,L}} \boldsymbol{\theta}$$

The step-size parameter μ is divided by L for normalization purpose. Following the same procedure as described in [6][8], we can obtain recursive equations for estimating $\mathbf{g}_{t_{i+1}}$.

The description for the variables appearing in this algorithm is the following :

$\mathbf{u}_{t_{i+1,L}}, \mathbf{v}_{t_{i+1,L}}, \mathbf{q}_{t_{i+1,L}}, \mathbf{p}_{t_{i+1,L}}, \mathbf{f}_{j,t_{i+1,L}} : M \times 1$ vectors.
 $B_{t_{i+1,L}}, K_{j,t_{i+1,L}}, G_{j,t_{i+1,L}} : M \times M$ matrices.

The updating equations for the term $\frac{\partial J_{2,t_{i+1,L}}^T}{\partial \boldsymbol{\theta}}$ are now :

$$\begin{cases} \frac{\partial J_{2,t_{i+1,L}}^T}{\partial \boldsymbol{\theta}} = 2\mathbf{u}_{t_{i+1,L}} + \mathbf{v}_{t_{i+1,L}} \\ \mathbf{u}_{t_{i+1,L}} = B_{t_{i+1,L}} \hat{\boldsymbol{\theta}}_{t_i} \\ B_{t_{i+1,L}} = B_{t_i,L} + \mathbf{h}_{t_{i+1}} \mathbf{h}_{t_{i+1}}^T - \mathbf{h}_{t_{i-L}} \mathbf{h}_{t_{i-L}}^T \\ (\mathbf{v}_{t_{i+1,L}})_j = \hat{\boldsymbol{\theta}}_{t_i}^T K_{j,t_{i+1,L}} \hat{\boldsymbol{\theta}}_{t_i} \\ K_{j,t_{i+1,L}} = G_{j,t_{i+1,L}} + G_{j,t_{i+1,L}}^T \\ G_{j,t_{i+1,L}} = G_{j,t_i,L} + \frac{\partial \mathbf{h}_{t_{i+1}}^T}{\partial \theta_j} \mathbf{h}_{t_{i+1}}^T - \frac{\partial \mathbf{h}_{t_{i-L}}^T}{\partial \theta_j} \mathbf{h}_{t_{i-L}}^T \end{cases} \quad (14)$$

In the same way, the resulting equations for obtaining $\frac{\partial J_{1,t_{i+1,L}}^T}{\partial \boldsymbol{\theta}}$ are given below :

$$\begin{cases} \frac{\partial J_{1,t_{i+1,L}}^T}{\partial \boldsymbol{\theta}} = \mathbf{q}_{t_{i+1,L}} + \mathbf{p}_{t_{i+1,L}} \\ \mathbf{q}_{t_{i+1,L}} = \mathbf{q}_{t_i,L} + \mathbf{h}_{t_{i+1}} y_{t_{i+1}} - \mathbf{h}_{t_{i-L}} y_{t_{i-L}} \\ (\mathbf{p}_{t_{i+1,L}})_j = \mathbf{f}_{j,t_{i+1,L}} \hat{\boldsymbol{\theta}}_{t_i} \\ \mathbf{f}_{j,t_{i+1,L}} = \mathbf{f}_{j,t_i,L} + y_{t_{i+1}} \frac{\partial \mathbf{h}_{t_{i+1}}^T}{\partial \theta_j} - y_{t_{i-L}} \frac{\partial \mathbf{h}_{t_{i-L}}^T}{\partial \theta_j} \end{cases} \quad (15)$$

The notation $(\cdot)_j$ stands for the j th element a vector. All vectors and matrices having index j correspond to this element. So, each updating equation containing these variables must be used for $j = 1, \dots, M$, that means for all the corresponding vector elements. The algorithm can be started with zero initial conditions for all the variables. This is especially recommended for vector $\boldsymbol{\theta}$ in the case of large model orders. The choice of initial conditions with large absolute values increases the risk of instability during the first steps.

Algorithm # 3 The cost function in this algorithm is defined as the mean-value of the squared prediction error over *all* the available samples :

$$J_{t_i} = \frac{1}{i} \sum_{j=1}^i \omega_{t_j} (y_{t_j} - \hat{y}_{t_j})^2 \quad (16)$$

ω_{t_j} is the weighting factor. The choice of an exponential weighting factor ($\omega_{t_j} = \lambda^{t_i - t_j}$) affords the possibility of operating in non stationary environments. Details of gradient estimation together with the formal description of the equations may be found in [5][6][8].

Simulation results have shown that the choice of the same value of μ as the one used in the classical LMS algorithm for contiguous measurements, is largely sufficient for the stability of the algorithms :

$$0 < \mu < \frac{1}{M \times \text{signal power}} \quad (17)$$

3 SIMULATION

In this section we study the performance of the described algorithms in the case of random sampling. Bernoulli pattern of misses is considered (the probability of missing each sample $q = 1 - p$ is fixed and the misses are independent). In all examples, simulations have been done with $q = 0.4$.

Example 1 The performances of the described algorithms in stationary cases are shown in this example. The test signals are two AR(2) processes with parameters $\theta_1^T = [1 \quad -1.8 \quad 0.9]$ and $\theta_2^T = [1 \quad -0.3 \quad 0.5]$. The samples of the first signal are much more correlated than those of the second one. Figs 1-2 show the convergence behavior of the mean squared error for each test signal. Algorithm # 3 has the best performance. This is more pronounced when the signal samples are more correlated.

Example 2 The algorithms were also tested in non stationary cases. The test signal is an AR(1) process with an abrupt change in its parameter. Algorithm # 3 was used with a forgetting factor $\lambda = 0.99$. The memory length was chosen to be $L = 100$ in algorithm # 2. Fig 3 shows the evolution of the estimated AR parameter in each case. The convergence behaviour of MSE for each algorithm is shown in Fig 4. We note that algorithm # 1 has the best tracking capacity. Algorithm # 2 follows better the signal variations than algorithm # 3 with a forgetting factor. However, the estimated AR parameter obtained by using algorithm # 3 have less variance than the ones in algorithms # 1 and # 2.

Example 3 We also tested these algorithms in the case of slowly time varying AR parameters. The test signal is an AR(2) process with $\theta^T = [1 \quad \theta_1 \quad \theta_2]$, where

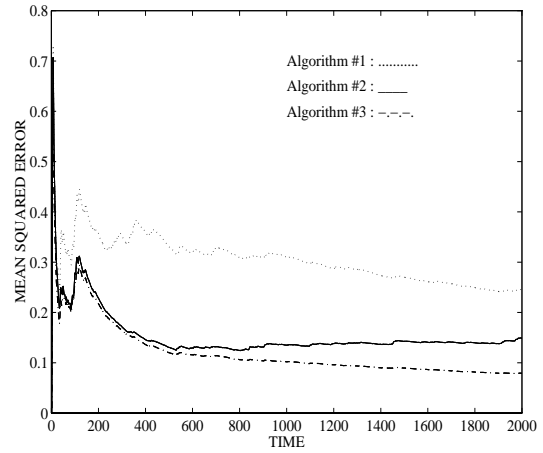


Figure 1: Convergence behavior of mean squared error. Example 1 : first test signal.

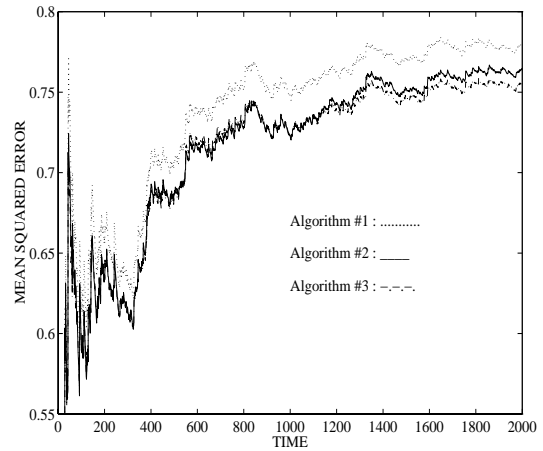


Figure 2: Convergence behavior of mean squared error. Example 1 : second test signal.

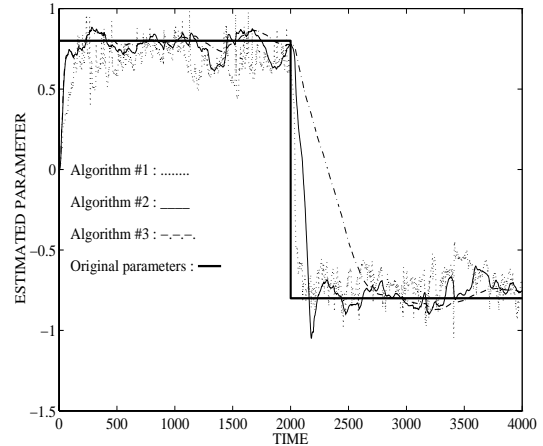


Figure 3: Estimated AR parameter. Test signal of example 2. Algorithm # 2 : $L = 100$. Algorithm # 3 : $\lambda = 0,99$.

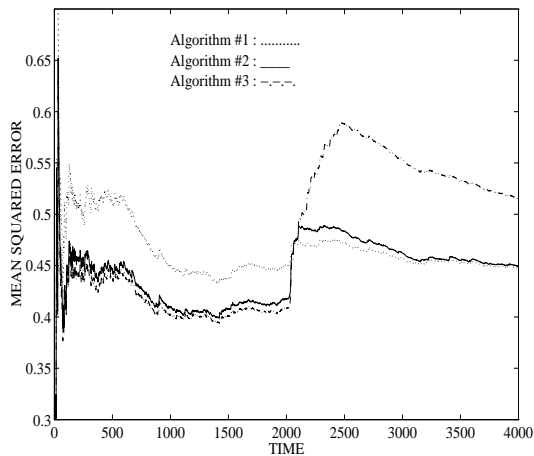


Figure 4: Convergence behavior of mean squared error. Test signal of example 2.

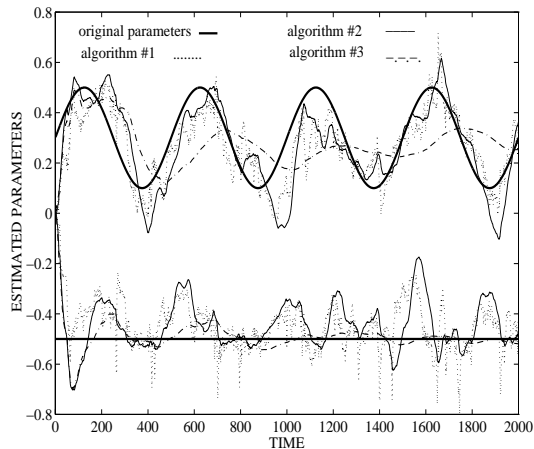


Figure 5: Estimated AR parameters. Test signal of example 3.

$\theta_1 = 0.3 + 0.2\cos(\frac{2\pi n}{500})$ and $\theta_2 = -0.5$. The evolution of the estimated AR parameters is shown in Fig 5. The values of λ and L are the same as the ones used in Example 2.

4 DISCUSSION

As mentioned before, the choice of each algorithm depends on the application and is a compromise between complexity, robustness, tracking capacity and storage requirements. From the previous and numerous other simulation results we can note the following points : Algorithm # 1 has the least complexity and the best tracking capacity. However, this algorithm is less robust faced to the problem of “missing observations” and leads to estimated parameters with large variances. This is why the steady-state MSE obtained by using this algorithm has the largest value.

Algorithm # 2 minimises the sum of squared prediction

errors over a fixed number (L) of observations. Letting L increase with the number of available samples, algorithm # 3 is obtained. By setting $L = 1$ the first algorithm is reached. The complexities of algorithms # 2 and # 3 are comparable. The choice of a “small” L increases the tracking capacity of algorithm # 2. However, as algorithm # 3 minimises the squared error over *all* available samples, it is more robust particularly when the number of lost samples is large. In stationary cases, algorithm # 3 has the best performance (Figs 1-2). In algorithm # 2, increasing L leads to the same performance as that of algorithm # 3 but this results in a worse tracking capacity. In addition, Algorithm # 2 has more storage requirements. It can be seen from Eqs. (14)-(15) that after each updating operation, vectors \mathbf{h}_{t-L} and $\frac{\partial \mathbf{h}_{t-L}^T}{\partial \theta_j}$, $j = 1, \dots, M$ must be stored. This is not necessary in algorithm # 3.

References

- [1] E. A. Parzen, “Times series analysis of irregularly observed data,” *Lect. Notes Stat*, vol. 25, 1984.
- [2] R. H. Jones, “Fitting a continuous time auto regression to discrete data,” *Applied times series analysis III*, pp. 651–682, 1981.
- [3] Y. Rozen and B. Porat, “Optimal ARMA Parameter Estimation Based on the Sample Covariances for Data with Missing Observations,” *IEEE Trans. Inform. Theory*, vol. 35, no. 2, pp. 342–349, 1989.
- [4] R. Veldhuis, *Restoration of lost samples in digital signals*. Prentice Hall, 1990.
- [5] S. Mirsaidi and J. Oksman, “Extension d’algorithmes de type gradient au cas d’échantillonnage non périodique,” *Proc. Quinzième Colloque Gretsi, Juans les Pins, FRANCE*, vol. 1, pp. 415–418, Sept. 1995.
- [6] S. Mirsaidi, G. Fleury, and J. Oksman, “LMS-like AR modeling in the case of missing observations. Submitted to,” *IEEE trans. Acoust., Speech, Signal Processing*.
- [7] S. Mirsaidi and J. Oksman, “AR Spectral Estimation with Randomly Missed Observations,” *Proc. 8th IEEE Signal Processing Workshop on Statistical Signal and Array Processing, Corfu, Greece*, June 1996.
- [8] S. Mirsaidi, “AR identification in the case of missing observations,” tech. rep., École supérieure d’Électricité, Gif-sur-Yvette, FRANCE, 1995.
- [9] S. Mirsaidi and J. Oksman, “Adaptive signal reconstruction from incomplete data,” *Proc. 30th annual Conference in Information Sciences and Systems, Princeton, New Jersey*, Mar. 1996.