ARMAX Identification Via Hereditary Algorithm

André Monin

ARMAX Identification Via Hereditary Algorithm

André Monin

Abstract— In this paper, we limit ourselves to the most commonly models used, say the Output Error (OE) model and Auto Regressive Moving Average with exogenous input (ARMAX). In both cases, we treat the case multi-input, single-output (MISO). Our aim is to give the parameters of the model under lattice form, in reference to its numerical robustness allowing to deal with great state space dimension (> 20). This calculation is achieved considering time varying predictor of the output based on the predictor derived at the previous step. It leads to an hereditary algorithm introduced in previous works [1].

Index Terms— Linear identification, lattice.

I. INTRODUCTION.

The paper concerns the problem of linear system identification. The stochastic linear identification is an issue that has been fully investigated for many years and many algorithms have been designed to this purpose (see [2] for bibliography), the most recent being the subspace identification method [3]. In some previous work [4] [1], we had derived some new realization and identification algorithms based on a hereditary computation in the single output AutoRegressive Moving Average (ARMA) case. We extend here these algorithms when the system has exogenous non-stationary multi-input. The main difference lies in the fact that the lattice parameters, involved in the computation of the system parameters, are no more stationary, as it was the case in [1].

A. Problem setting.

The aim of linear identification is to fit some linear stochastic realization to an experimental set measurement of the input/output of some unknown system. We restrict ourselves here to the set of AutoRegressive Moving Average with exogenous input (ARMAX) models:

$$y_t = \sum_{i=1}^{n} a^i y_{t-i} + \sum_{k=1}^{p} \sum_{i=0}^{n-1} b^{i,k} u_{t-i}^k + \sum_{i=1}^{n} c^i v_{t-i} + v_t \quad (1)$$

where y_t stands for the (measured) output, $\{u_t^k, k = 1...n\}$ stands for the (measured) inputs, v_t stands for a white noise with unknown variance and $\{a^i, i = 1...n\} \cup \{b^{i,k}, i = 0...n-1, k = 1...p\} \cup \{c^i, i = 1...n\}$ stands for the set of unknown parameters to be identified. Note that, for convenience, we fix the same order n for the AR, the MA and the X part. Referring to the Gaussian case, the accuracy of the model derived is commonly measured through the one-step-ahead-prediction error of the output, that is:

$$J_t = \frac{1}{t} \sum_{\tau=1}^{t} \left\| y_\tau - \widehat{y}_{\tau|\tau-1} \right\|^2 \quad (2)$$

A. Monin is with the LAAS-CNRS, Toulouse, France. E-mail: monin@laas.fr.

where $\widehat{y}_{\tau|\tau-1}$ stands for the predictor of the output. If one may assume that the zeros of the MA part are all inside the unit circle, the output predictor $\widehat{y}_{t|t-1}$ is [2]:

$$\widehat{y}_{t|t-1} = \sum_{i=1}^{n} a^i \widehat{y}_{t-i|t-i-1} + \sum_{k=1}^{p} \sum_{i=0}^{n-1} b^{i,k} u_{t-i}^k + \sum_{i=1}^{n} k^i \widetilde{y}_{t-i|t-i-1}$$

$$(3)$$

where $\widetilde{y}_{t|t-1} = y_t - \widehat{y}_{t|t-1}$ is the prediction error and $k^i = a^i - c^i$.

Note that it is assumed that the whole data record over the interval $]-\infty, t]$ is available. Obviously, in practice, only data over the interval $[1, t]$ are known. The common way to get round this difficulty is to use windowed data. Most often, one just assumes that the past data, for $t < 1$, is zero. This is equivalent to consider that the initial condition, in a state space representation, is zero, which is not true in most cases. The main drawback of this assumption is that, for short experiments, it is more difficult to fit data to the model with non-zero initial conditions. The true model for finite data record would be:

$$\forall t = 1...n,$$
$$y_t = x_t^0 + \sum_{i=1}^{n} a^i y_{t-i} + \sum_{k=1}^{p} \sum_{i=0}^{n-1} b^{i,k} u_{t-i}^k + \sum_{i=1}^{n} c^i v_{t-i} + v_t$$

where $\{x_t^0, t = 1...n\}$ stands for the n initial conditions of the system. For $t > n$, 1 is then fully defined. To represent the initial condition effect, it is convenient to introduce an extra virtual impulsive input u_t^0 defined by $u_1^0 = 1$ and $u_t^0 = 0, \forall t \neq 1$. The complete model is then:

$$y_t = \sum_{i=1}^{n} a^i y_{t-i} + \sum_{k=0}^{p} \sum_{i=0}^{n-1} b^{i,k} u_{t-i}^k + \sum_{i=1}^{n} c^i v_{t-i} + v_t$$

with, $b^{i,0} = x_{i+1}^0, \forall i = 0...n-1$. With this modified model, all data must be assumed zero $\forall t < 1$.

B. Computing the parameters via hereditary projection.

Recall that the criterion 2 is not quadratic with respect to the parameters, as it is the case for ARX models ($c^i = 0, i = 1...n$). Indeed, if one develops the dependence of the criterion with respect to the parameters a^i for instance, the AR part appearing in 3 involves an infinite development, which is not explicitly tractable. Many approximating algorithms have been developed to compute the model parameters as the so-called Extended Least Square [5] [6] [2] or Gauss-Newton algorithms [7] [6] [8]. In none of these approaches, transient behavior is optimized. In this paper, we deal with a time varying approach preserving a transient convergence similar to that of [1]. Let us denote $\widehat{y}_\tau^{t-1}, \tau = 1...t-1$ the optimal predictor that minimizes the criterion J_{t-1}. The parameters have been obtained from the

knowledge of the data record $\{y_\tau, u_\tau^k, \tau = 1...t-1, k = 0...p\}$. At time t, the new predictor trajectory is computed as follows:

$$\forall \tau = 1...t, \ \widehat{y}_\tau^t = \sum_{i=1}^n a_t^i \widehat{y}_{\tau-i}^{t-i} + \sum_{k=0}^p \sum_{i=0}^{n-1} b_t^{i,k} u_{t-i}^k$$

That means that the whole trajectory from 1 to t is recomputed at each step. In such a time varying scheme, the criterion is quadratic with respect to $\{a_t^i, b_t^{i,k}\}$ (but not with respect to $\{a_\tau^i, b_\tau^{i,k}, \tau = 1...t - 1\}$ previously computed). Therefore, the optimal parameters can be computed using classical projection formula. On the other hand, it leads to a hereditary algorithm. That means that the memory needed for computation is linearly growing with time due to the updating of the overall least-square criterion as data accumulate. In other words, the non-linearity complexity is handled via hereditary computation.

II. GENERAL LEVINSON FORMULA.

Let us note the experimental mean of a process $H_\tau \in \mathbb{R}^{n \times p}$ defined over the interval $[1, t]$ as $E^t[H_\bullet] = \frac{1}{t} \sum_{\tau=1}^t H_\tau$.

Definition 1: The projection of a process $X_\tau \in \mathbb{R}^n, \tau = 1...t$ over the process $Y_\tau \in \mathbb{R}^p, \tau = 1...t$, is the process $Z_\tau \triangleq P_\tau^t[X_\bullet | Y_\bullet] = K_t Y_\tau$ where $K_t = E^t[X_\bullet Y_\bullet^T](E^t[Y_\bullet Y_\bullet^T])^{-1}$.

It is clear that this projection yields the process that minimizes the mean square error $E^t[\|X_\bullet - Z_\bullet\|^2]$.

Let us define $Z_\bullet^t \in \mathbb{R}^{p+3}$ as:

$$Z_\tau^t = [u_{\tau+1}^0, ..., u_{\tau+1}^p, \widehat{y}_\tau^t, \widetilde{y}_\tau^t]^T, \tau = 1...t \tag{4}$$

Assume that $\forall \tau < 1, Z_\tau^t = 0$. The linear predictor $\widehat{y}_\tau^t$ is then computed using the projection of definition 1:

$$\widehat{y}_\tau^t = P_\tau^t[y_\bullet | Z_{\bullet-1}^{t-1}, ..., Z_{\bullet-n}^{t-n}] \tag{5}$$

Suppose that $\forall \tau < 1, X_\tau = Y_\tau = 0$. One then has:

Proposition 2: $P_\tau^t[X_{\bullet-1} | Y_{\bullet-1}] = P_{\tau-1}^{t-1}[X_\bullet | Y_\bullet]$

As it appears in 5, the linear predictor is computed as a linear projection over the space spanned by $\{Z_{\bullet-1}^{t-1}, ..., Z_{\bullet-n}^{t-n}\}$. The Levinson formulae are devoted to recursively compute an orthogonal basis of this space according to the inner product defined by $\langle X_\bullet | Y_\bullet \rangle_t \triangleq E^t[X_\bullet Y_\bullet^T]$. Let us define the family $\{V_{\bullet-1}^{0,t-1}, ..., V_{\bullet-1}^{n-1,t-1}\}$ (called backward residuals) to be the orthonormalized basis of the space spanned by $\{Z_{\bullet-1}^{t-1}, ..., Z_{\bullet-n}^{t-n}\}$. According to this definition, one has:

$$E^t[V_{\bullet-1}^{i,t-1}(V_{\bullet-1}^{j,t-1})^T] = 0, \forall i \neq j$$
$$E[V_{\bullet-1}^{i,t-1}(V_{\bullet-1}^{i,t-1})^T] = I, \forall i = 0...n-1$$

where I stands for the identity matrix. One has:

Proposition 3: Backward and forward normalized residuals obey the following relations:

$$\forall \tau = 1...t, \begin{cases} V_\tau^{0,t} = (\frac{t}{t+1} L_t^0)^{-1/2} Z_\tau^t \\ V_\tau^{i,t} = \left[\frac{t}{t+1}(I - S_t^{i-1}(S_t^{i-1})^T) \right]^{-1/2} \\ \times (V_{\tau-1}^{i-1,t-1} - S_t^{i-1} W_\tau^{i-1,t}), \forall i = 1...n-1 \end{cases} \tag{6}$$

$$\forall \tau = 1...t, \begin{cases} W_\tau^{0,t} = (L_t^0)^{-1/2} Z_\tau^t \\ W_\tau^{i,t} = \left[I - S_t^{i-1}(S_t^{i-1})^T \right]^{-1/2} \\ \times (W_\tau^{i-1,t} - (S_t^{i-1})^T V_{\tau-1}^{i-1,t-1}), \forall i = 1...n-1 \end{cases}$$

where:

$$L_t^0 = E^t[Z_\bullet^t(Z_\bullet^t)^T] \tag{7}$$
$$S_t^i = E^t[V_{\bullet-1}^{i,t-1}(W_\bullet^{i,t})^T], \forall i = 0...n-2 \tag{8}$$

Moreover, thanks to the orthonormality property of the residuals, the one-step-ahead predictor has the following expression:

$$\widehat{y}_{\tau|\tau-1}^t = \sum_{i=0}^{n-1} e_{p+2}^T M_t^i V_{\tau-1}^{i,t-1}, \forall \tau = 1...t \tag{9}$$

where e_k stands for the vector of $\mathbb{R}^{p+3}$ defined by: $e_k^k = 1$, $\forall i \neq k, e_k^i = 0$ and:

$$M_t^i = E^t[Z_\bullet^t(V_{\bullet-1}^{i,t-1})^T] \tag{10}$$

See appendix VIII for proof.

III. HEREDITARY ALGORITHM.

Suppose that, at previous step $(t-1)$, the whole sequences $\{V_\tau^{i,t-1}, i = 0...n-1, \tau = 1...t-1\}$ (backward residual trajectories) are available. One then computes the matrices $\{M_t^i, i = 0...n-1\}$ and L_t^0 defined by 10 and 7 performing a summation over the interval $[1, t]$. Note that although this calculus seems implicit, it can be performed using the orthogonality property of the innovation process $E^t[(y_\bullet - \widehat{y}_{\bullet|\bullet-1}^t)V_{\bullet-1}^{i,t-1}] = 0$. One may then compute the matrices $\{S_t^i, i = 0...n-1\}$ using recursively equations 18 and 25 of appendix VIII. It remains then to update the residual trajectories. Here is the hereditary part of the algorithm:

For $\tau = 1...t$, do:
- $\widehat{y}_\tau^t = \sum_{i=0}^{n-1} e_{p+2} M_t^i V_{\tau-1}^{i,t-1}$
- $Z_\tau^{0,t} = \begin{bmatrix} u_{\tau+1}^0 & ... & u_{\tau+1}^p & \widehat{y}_\tau^t & (y_\tau - \widehat{y}_\tau^t) \end{bmatrix}^T$
- $W_\tau^{0,t} = (L_t^0)^{-1/2} Z_\tau^{0,t}$
- For $i = 1...n-1$,

$$W_\tau^{i,t} = \left[I - S_t^{i-1}(S_t^{i-1})^T \right]^{-1/2} (W_\tau^{i-1,t} - (S_t^{i-1})^T V_{\tau-1}^{i-1,t-1})$$

- For $i = n-1...1$,

$$V_\tau^{i,t} = \left[\frac{t}{t+1}(I - S_t^{i-1}(S_t^{i-1})^T) \right]^{-1/2} (V_{\tau-1}^{i-1,t-1} - S_t^{i-1} W_\tau^{i-1,t})$$

- $V_\tau^{0,t} = (\frac{t}{t+1} L_t^0)^{-1/2} Z_\tau^t$

Obviously, during the first steps of the algorithm, the covariance matrices, such as S_t^i, are singular since all data prior to $\tau = 1$ are assumed to be zero. Therefore, the coefficients related to these data are undetermined. In fact, only one new non-zero coefficient is derived at each step. That means that one needs at least $n \times (p+3)$ steps to assure the regularity of all matrices. There are two ways to solve this problem.

- One may use pseudo-inverse procedure when one needs to inverse a singular matrix.
- One may also compute previously the rank of matrices by analyzing the algorithm. For example, one has to compute the inverse of the square root of L_t^0. It is clear that, if the input-output sequence is not singular, the rank of $L_t^0 \in \mathbb{R}^{p+3}$ is equal to t, since $t \leqslant (p+3)$. Moreover, one has to compute the inverse of the square root of the matrices

$A_t^i = (I - (S_t^i)^T S_t^i)$. If r denotes the rank of S_t^i, the rank of A_t^i is obviously $p + 3 - r$. Because only one parameter is derived at each step, one needs to wait $(p + 3)(i + 1)$ steps to assure that S_t^i has a rank equal to $p + 3$. More precisely, one has:

- If $t \leqslant (p+3)(i-1) + 1$ or $t \geqslant (p+3)i$, then $rk(A_t^i) = p + 3$
- If $t \in [(p+3)(i-1) + 1, (p+3)i]$, then $rk(A_t^i) = (p+3)i - t + 1$
- If $t \in [(p+3)i + 1, (p+3)(i+1)]$, then $rk(A_t^i) = t - (p+3)i$

where $rk(A)$ stands for the rank of the matrix A.

Note that these procedures do not harm the optimality of the algorithm since it just provides a particular set of parameters among all possible solutions.

IV. COMPUTING THE TRANSFER FUNCTION COEFFICIENTS

For some purposes, it could be convenient to deal with the transfer function related to the linear system involved under lattice form. To achieve this goal, it is necessary to take into account that the lattice parameters are not stationary, since the inputs are not, in general, a perfect white noise process. Indeed, some parameters are resulting from the projection of the inputs over their own past. First note that, at time t, the state of the linear one-step-ahead predictor can be reduced to the sets:

$$\{V_{t-1}^{0,t-1}, ..., V_{t-n}^{0,t-n}\}_{sp} = \{Z_{t-1}^{t-1}, ..., Z_{t-n}^{t-n}\}_{sp}$$

thanks to formula 6. According to 3, if one sets all components of the state to zero except one term to be equal to one, the parameters of the transfer function can be computed as the output of the filter. More precisely, if one sets:

$$
\begin{aligned}
Z_{t-i}^{t-i} &= \quad 0 \quad .. \quad 0 \quad \underset{j}{1} \quad 0 \quad ... \quad 0 \\
Z_{t-j}^{t-j} &= \quad 0, \forall j \neq i
\end{aligned}
$$

one obtains a^i for $j = p+2$, d^{i-1} for $j = p+3$, $b^{i-1,k}$ for $j = k$ ($j \neq 0$) and x_i^0, the i-th component of the initial condition, for $j = 0$. The algorithm is then structured as follows:

- For $i = 1...n$, do $V_{t-i}^{0,t-i} = (\frac{t-i}{t-i+1} L_{t-i}^0)^{-1/2} Z_{t-i}^{t-i}$ where Z_{t-i}^{t-i} is initialized as described above.
- For $i = 1...n$, do $W_{t-i}^{0,t-i} = \sqrt{\frac{t-i+1}{t-i}} V_{t-i}^{0,t-i}$
- For $k = 1...n-1$ do for $\tau = 1...n-1$ do:
  - $V_{t-\tau}^{k,t-\tau} = \left[\frac{t-\tau}{t-\tau+1}(I - S_{t-\tau}^{k-1}(S_{t-\tau}^{k-1})^T)\right]^{-1/2}(V_{t-\tau-1}^{k-1,t-\tau} - S_{t-\tau}^{k-1} W_{t-\tau}^{k-1,t-\tau})$
  - $W_{t-\tau}^{k,t-\tau} = \left[I - (S_{t-\tau}^{k-1})^T S_{t-\tau}^{k-1}\right]^{-1/2}(W_{t-\tau}^{k-1,t-\tau} - (S_{t-\tau}^{k-1})^T V_{t-\tau-1}^{k-1,t-\tau-1})$
- $\theta = \sum_{i=0}^{n-1} e_{p+2}^T M_t^i V_{t-1}^{i,t-1}$ where θ stands for the parameter to be computed, that is a^i, $b^{i,k}$ or x_0^i, depending on the value of j.

V. EXTENSIONS TO MORE GENERAL LINEAR SYSTEMS.

In the introduction, we stated that, for convenience, we fixed the same order n for the numerators and denominator of the transfer function. In fact, there is no trouble to deal with more general transfer functions.

A. Introduction of delays on the inputs.

If one wishes to introduce time delays on the input, one just has to replace Z_τ^t defined in 4 with $Z_\tau^t = [u_{\tau+1}^0, u_{\tau+1-\sigma_1}^1 ..., u_{\tau+1-\sigma_p}^p, \hat{y}_\tau^t, \hat{y}_\tau^t]^T$, $\tau = 1...t$ where σ_k stands for the delay fixed for the k-th input. Note that the first input u^0 being the virtual impulsive input intended to represent the initial condition effect, it remains unchanged.

B. Dealing with different orders for numerators and denominator.

If one wishes to deal with different orders for numerators and denominator, first replace n with the maximum of the orders desired. The definition of the predictor $\hat{y}_\tau^t$ is then modified as $\hat{y}_\tau^t = P_\tau^t[y_\bullet | E_1 Z_{\bullet-1}^{t-1}, ..., E_n Z_{\bullet-n}^{t-n}]$ where E_i is a square matrix that sets to zero the appropriate components of $Z_{\bullet-i}^{t-i}$. All computations of the algorithms being linear operations, there is no difficulty to modify the algorithm with this change.

C. Time-varying systems.

In the case of time-varying systems, if T stands for the tracking-time constant, one just has to modify the definition of the experimental mean as follows:

$$
\begin{aligned}
E^t[X_\bullet] &= \frac{1}{t}\sum_{\tau=1}^{t} X_\tau, \forall t < T \\
E^t[X_\bullet] &= \frac{1}{T}\sum_{\tau=t-T+1}^{t} X_\tau, \forall t \geq T
\end{aligned}
$$

VI. EXPERIMENTAL RESULTS.

As the main goal of this identification algorithm is to optimize the transient (short experiments), the first test we made was proceeded in the noise free case. We have verified, in simulation, that for n-dimensional linear systems with p inputs, the parameters are obtained exactly (in fact with a precision lower that 10^{-10}) after $n(p + 3)$ steps, which is obviously the minimal number of steps required.

We used for experiments some datasets coming from DAISY [9]. We have compared the performances of our hereditary algorithm with the results obtained by the standard MATLAB identification toolbox functions: `n4sid.m` (subspace identification), `armax.m` and `oe.m` (Newton like algorithms). For the ARMAX and OE functions, we added the extra impulsive input described in I, in view to deal with an unknown initial condition An example of results got with the dataset [96-011] with $n = 4$ is shown figure 1. We have represented the relative output error $\sqrt{\sum_{\tau=1}^{T}(y_\tau - \hat{y}_\tau^t)^2 / \sum_{\tau=1}^{T} y_\tau^2}$, computed over the whole dataset ($T = 1680$ samples), when the parameters are computed over only a smaller varying data length (from 10 to 500 per step of 10). In view to clarify the graphical representation, we set this error to be zero when the system derived was unstable. It appears clearly that, for this example, the output error of the hereditary algorithm has almost reached its minimum after only 150 steps. Note that this output error is no more fluctuating for longest datasets,

contrary to those obtained with standard algorithms. Moreover, all systems obtained with our algorithm are stable for a dataset of length $\geqslant 30$ and leads to an error $\leqslant 25\%$.

In view to illustrate the behavior of the algorithm, we have simulated a 6-order linear system with zeros located at $\{z = 1, z = 0.9\exp(\pm i\pi/4), z = 0.9\exp(\pm 3i\pi/8)\}$ and poles located at $\{p = 0.9\exp(\pm i\pi/8), p = 0.9\exp(\pm i\pi/3), p = 0.5\exp(\pm 13i\pi/32)\}$. The input is a Gaussian white noise and signal to noise ratio is $12dB$. The hereditary estimates of the first two parameters of the denominator are plotted figure 2 and compared to the results obtained with the standard `oe.m` MATLAB identification toolbox function. It appears clearly that the algorithm that uses a classical Newton type method falls on several occasions in local minima.
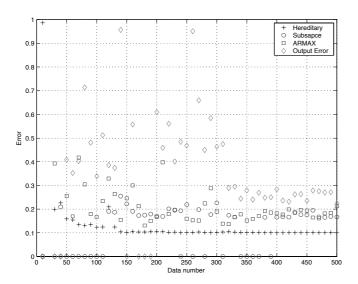

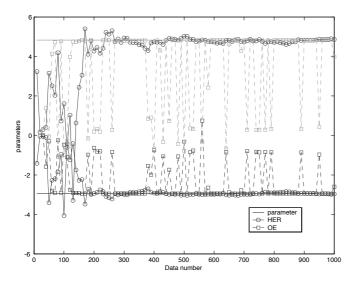Fig. 1. Heat flow density through a two layer wall dataset

Fig. 2. Time-varying parameter estimates

VII. CONCLUSION.

We have derived a new identification algorithm for ARMAX systems based on a hereditary computation. The main benefit of this algorithm is that it is optimal in the transient. We have shown that the introduction of an extra virtual impulsive input could be used to extract the estimation of initial conditions. Moreover, it appears that no tuning parameter is needed when the input-output is not singular. When it is not the case, for binary signals for example, the algorithm uses pseudo-inverses since the dataset remain singular, which does not harm its efficiency.

VIII. PROOF OF PROPOSITION 3.

Let us define the following sets:

- Backward residuals: $\forall \tau = 1...t$,

$$\begin{cases} X_\tau^{0,t} = Z_\tau^t \\ X_\tau^{i,t} = Z_{\tau-i}^{t-i} - P_\tau^t[Z_{\bullet-i}^{t-i}|Z_{\bullet-i+1}^{t-i+1},...,Z_\bullet^t], \forall i = 1...n-1 \end{cases} \tag{11}$$

- Forward residuals: $\forall \tau = 1...t$,

$$\begin{cases} Z_\tau^{0,t} = Z_\tau^t \\ Z_\tau^{i,t} = Z_\tau^t - P_\tau^t[Z_\bullet^t|Z_{\bullet-1}^{t-1},...,Z_{\bullet-i}^{t-i}], \forall i = 1...n-1 \end{cases}$$

With these definitions, one clearly has:

$$\{Z_{\bullet-1}^{t-1},...,Z_{\bullet-n}^{t-n}\}_{sp} = \{X_{\bullet-1}^{0,t-1}\}_{sp} \oplus ... \oplus \{X_{\bullet-1}^{n-1,t-1}\}_{sp} \tag{12}$$
$$\{Z_\bullet^t,...,Z_{\bullet-i}^{t-i}\}_{sp} = \{Z_{\bullet-i}^{0,t-i}\}_{sp} \oplus ... \oplus \{Z_\bullet^{i,t}\}_{sp} \tag{13}$$

Using 13, the backward residuals can be rewritten as follows:

$$X_\tau^{i,t} = Z_{\tau-1}^{t-i} - P_\tau^t[Z_{\bullet-i}^{t-i}|Z_{\bullet-i+1}^{t-i+1},...,Z_{\bullet-1}^{t-1}] - P_\tau^t[Z_{\bullet-i}^{t-i}|Z_\bullet^{i-1,t}] \tag{14}$$

But, at previous step $t-1$, according to 11, one had:

$$\begin{aligned} X_{\tau-1}^{i-1,t-1} &= Z_{\tau-i}^{t-i} - P_{\tau-1}^{t-1}[Z_{\bullet-i+1}^{t-i}|Z_{\bullet-i+2}^{t-i+1},...,Z_\bullet^{t-1}] \\ &= Z_{\tau-i}^{t-i} - P_\tau^t[Z_{\bullet-i}^{t-i}|Z_{\bullet-i+1}^{t-i+1},...,Z_{\bullet-1}^{t-1}] \end{aligned}$$

according to property 2:

$$X_{\tau-1}^{i-1,t-1} = Z_{\tau-1}^{t-i} - P_\tau^t[Z_{\bullet-i}^{t-i}|Z_{\bullet-i+1}^{t-i+1},...,Z_{\bullet-1}^{t-1}]$$

Therefore, formula 14 takes the following form:

$$X_\tau^{i,t} = X_{\tau-1}^{i-1,t-1} - P_\tau^t[X_{\bullet-1}^{i-1,t-1}|Z_\bullet^{i-1,t}] \tag{15}$$

In the same way, using 12 in the expression of forward residuals, one has:

$$\begin{aligned} Z_\tau^{i,t} &= Z_\tau^t - P_\tau^t[Z_\bullet^t|X_{\bullet-1}^{0,t-1}] - ... - P_\tau^t[Z_\bullet^t|X_{\bullet-1}^{i-1,t-1}] \\ &= Z_\tau^{i-1,t} - P_\tau^t[Z_\bullet^t|X_{\bullet-1}^{i-1,t-1}] \end{aligned}$$

The aim is now to construct the set of residuals $\{V_\tau^{i,t}, i = 0...n-1\}$, where $V_\tau^{i,t} = N_t^i X_\tau^{i,t}$, such that $E^t[V_{\bullet-1}^{i,t-1}(V_{\bullet-1}^{i,t-1})^T] = I$. Note that, using the assumption that $V_\tau^{i,t} = 0, \forall \tau < 1$, this is equivalent to ensure that:

$$E^t[V_\bullet^{i,t}(V_\bullet^{i,t})^T] = \frac{t+1}{t}I \tag{16}$$

Let us define $L_t^i \triangleq E^t[Z_\bullet^{i,t}(Z_\bullet^{i,t})^T], \forall i = 0...n-1$ and $M_t^i \triangleq E^t[Z_\bullet^{i,t}(V_{\bullet-1}^{i-1,t-1})^T], \forall i = 0...n-1$. According to 15, one has:

$$V_\tau^{i,t} = N_t^i(N_{t-1}^{i-1})^{-1}\left[V_{\tau-1}^{i-1,t-1} - \left(M_t^i\right)^T\right](L_t^i)^{-1}Z_\tau^{i-1,t} \tag{17}$$

The autocorrelation matrix of $V_\tau^{i,t}$ can then be written as follows:

$$E^t[V_\bullet^{i,t}(V_\bullet^{i,t})^T] = N_t^i(N_{t-1}^{i-1})^{-1}$$

$$\times \left[I - (M_t^{i-1})^T(L_t^{i-1})^{-1}M_t^{i-1}\right](N_{t-1}^{i-1})^{-T}(N_t^i)^T$$

where A^{-T} stands for the inverse of transposed matrix A. Let us define:

$$S_t^i \triangleq (M_t^i)^T(L_t^i)^{-1/2} \tag{18}$$

Assumption 16 leads to:

$$N_t^i(N_{t-1}^{i-1})^{-1} = \left[\frac{t}{t+1}(I - S_t^{i-1}(S_t^{i-1})^T)\right]^{-1/2} \tag{19}$$

Let $W_\tau^{i,t} = (L_t^i)^{-1/2}Z_\tau^{t,i}$ be the normalized form of the forward residuals. Expression 6 is a directly involves from 17 and 19. Note that, according this definition, one has $S_t^i = E^t[V_{\bullet-1}^{i,t-1}(W_\bullet^{i,t})^T]$. In the same way, the recursive equation of the forward residuals can be written:

$$\begin{aligned}W_\tau^{i,t} &= (L_t^i)^{-1/2}(L_t^{i-1})^{1/2}[W_\tau^{i-1,t} - E^t[W_\bullet^{i-1,t}(X_{\bullet-1}^{i-1,t-1})^T] \\ &\times(E^t[X_{\bullet-1}^{i-1,t-1}(X_{\bullet-1}^{i-1,t-1})^T])^{-1}]X_{\tau-1}^{i-1,t-1} \end{aligned} \tag{20}$$

Introducing the normalized backward residuals, one has:

$$E^t[W_\bullet^{i-1,t}(X_{\bullet-1}^{i-1,t-1})^T] = E^t[W_\bullet^{i-1,t}(V_{\bullet-1}^{i-1,t-1})^T](N_{t-1}^{i-1})^{-T} \tag{21}$$

and:

$$E^t[X_{\bullet-1}^{i-1,t-1}(X_{\bullet-1}^{i-1,t-1})^T] = (N_{t-1}^{i-1})^{-1}(N_{t-1}^{i-1})^T \tag{22}$$

Reporting both 21 and 22 in 20 leads to:

$$\begin{aligned}&E^t[W_\bullet^{i-1,t}(X_{\bullet-1}^{i-1,t-1})^T](E^t[X_{\bullet-1}^{i-1,t-1}(X_{\bullet-1}^{i-1,t-1})^T])^{-1} \\ &= (S_t^{i-1})^T N_{t-1}^{i-1}\end{aligned}$$

Consequently, formula 20 becomes:

$$W_\tau^{i,t} = (L_t^i)^{-1/2}(L_t^{i-1})^{1/2}\left[W_\tau^{i-1,t} - (S_t^{i-1})^T V_{\tau-1}^{i-1,t-1}\right] \tag{23}$$

Moreover, according to the normalization procedure, one has $E^t[W_\bullet^{i,t}(W_\bullet^{i,t})^T] = I$. The computation of the experimental covariance of $W_\tau^{i,t}$ leads to:

$$I = (L_t^i)^{-1/2}(L_t^{i-1})^{1/2}\left[I - (S_t^{i-1})^T S_t^{i-1}\right](L_t^{i-1})^{1/2}(L_t^i)^{-1/2} \tag{24}$$

which leads to:

$$(L_t^i)^{-1/2}(L_t^{i-1})^{1/2} = \left[I - (S_t^{i-1})^T S_t^{i-1}\right]^{-1/2} \tag{25}$$

Reporting 24 in 23 leads to the result.

REFERENCES

[1] A. Monin and G. Salut, "ARMA lattice identification: A new hereditary algorithm," *IEEE Trans. on Signal Processing*, vol. 44, no. 2, pp. 360–370, 1996.
[2] L. Ljung, *System Identification: Theory For The User*. Prentice-Hall Information and System Sciences Series, Upper Saddle River, New Jersey 07458, 1999.
[3] P. Van Overchee and B. De Moor, *Subspace Identification of LInear Systems: Theory, Implementation and Applications*. Kluwer Academic Publishers, 1996.
[4] A. Monin and G. Salut, "Exact ARMA lattice predictors from autocorrelations functions," *IEEE Transactions on Signal Processing*, vol. 42, no. 4, pp. 877–886, 1994.
[5] A. Benveniste and C. Chaure, "AR and ARMA algorithms of levinson type: An innovation approach," *IEEE Transactions on Automatic Control*, vol. 26, no. 6, pp. 1243–1261, 1981.
[6] A. Benveniste, M. Metivier, and P. Priouret, *Algorithmes Adaptatifs et Approximations Stochastiques*. Paris: Edition Masson, 1987.
[7] J. J. Shynk, "Adaptive IRR filtering," *IEEE Acoust. Speech, Signal Processing Magazine*, vol. 6, pp. 4–21, 1989.
[8] F. Michaut, *Méthodes Adaptatives Pour Le Signal*. Paris: Hermès édition, 1992.
[9] B. De Moor, *DaISy: Database for the Identification of Systems*, Department of Electrical Engineering, ESAT/SISTA, K.U.Leuven, Belgium, 2002, uRL: http://www.esat.kuleuven.ac.be/sista/daisy/.