ID6
ARMAX Identification

6.13 PEM ESTIMATION OF ARMAX MODELS

In the case of ARMAX models it is not possible to implement a PEM estimate, by finding the value of θ which minimizes the cost function $J_N(\theta)$, in an analytic way as has been done with ARX models. The only possibility consists in relying on iterative algorithms like Newton–Raphson consisting in the expression

$$\theta^{k+1} = \theta^k - \left(J_N^{''}(\theta^k)\right)^{-1} J_N^{'}(\theta^k), \tag{6.13.1}$$

where θ^k denotes the parameter vector

$$\theta^k = \left[\, \alpha_1^k \ \ldots \ \alpha_n^k \ \beta_1^k \ \ldots \ \beta_n^k \ \gamma_1^k \ \ldots \ \gamma_n^k \,\right]^T \tag{6.13.2}$$

obtained at the kth iteration. Denoting with $\psi(t,\theta)$ $(d \times 1)$ the gradient of $\varepsilon(t,\theta)$ (with sign changed)

$$\psi(t,\theta) = -\frac{\partial \varepsilon(t,\theta)}{\partial \theta} \tag{6.13.3}$$

it follows that

$$J_N^{'}(\theta) = -\frac{2}{N}\sum_{t=1}^{N} \varepsilon(t,\theta)\,\psi(t,\theta) \tag{6.13.4}$$

$$J_N^{''}(\theta) = \frac{\partial}{\partial \theta}\left(J_N^{'}(\theta)\right)^T = \frac{2}{N}\sum_{t=1}^{N} \psi(t,\theta)\,\psi^T(t,\theta) + \frac{2}{N}\sum_{t=1}^{N} \varepsilon(t,\theta)\frac{\partial^2 \varepsilon(t,\theta)}{\partial \theta^2}. \tag{6.13.5}$$

At the minimum of $J_N(\theta)$, $\varepsilon(t,\theta)$ tends asymptotically, for $N \to \infty$, to the white noise $w(t)$, independent from $\partial^2 \varepsilon(t,\theta)/\partial \theta^2$; in the surroundings of this minimum it is thus possible to introduce the approximation

$$J_N^{''}(\theta) \simeq \frac{2}{N}\sum_{t=1}^{N} \psi(t,\theta)\,\psi^T(t,\theta) \tag{6.13.6}$$

leading to the iterative Gauss–Newton algorithm

$$\theta^{k+1} = \theta^k + \left[\sum_{t=1}^{N} \psi(t,\theta^k)\, \psi^T(t,\theta^k) \right]^{-1} \sum_{t=1}^{N} \varepsilon(t,\theta^k)\, \psi(t,\theta^k)\,. \tag{6.13.7}$$

The implementation of this algorithm requires the computation of the gradient of $\varepsilon(t,\theta)$. Assuming $\varepsilon(t,\theta) = w(t)$, from expression (6.1.6) we obtain

$$r(z^{-1})\, \varepsilon(t,\theta) = q(z^{-1})\, y(t) - p(z^{-1})\, u(t); \tag{6.13.8}$$

deriving with respect to α_i we obtain

$$r(z^{-1})\, \frac{\partial \varepsilon(t,\theta)}{\partial \alpha_i} = -z^{i-n-1} y(t) = -y(t+i-n-1) \tag{6.13.9}$$

$$\frac{\partial \varepsilon(t,\theta)}{\partial \alpha_i} = -\frac{1}{r(z^{-1})}\, y(t+i-n-1). \tag{6.13.10}$$

To compute $\partial \varepsilon(t,\theta)/\partial \alpha_i$ for $i = 1,\ldots,n$ it is thus sufficient to filter the sequence $y(t)$ with $1/r(z^{-1})$. Similarly, by deriving (6.13.8) with respect to β_i we find that

$$\frac{\partial \varepsilon(t,\theta)}{\partial \beta_i} = -\frac{1}{r(z^{-1})}\, u(t+i-n-1) \tag{6.13.11}$$

while deriving with respect to γ_i we find

$$\varepsilon(t+i-n-1,\theta) + r(z^{-1})\, \frac{\partial \varepsilon(t,\theta)}{\partial \gamma_i} = 0 \tag{6.13.12}$$

$$\frac{\partial \varepsilon(t,\theta)}{\partial \gamma_i} = -\frac{1}{r(z^{-1})}\, \varepsilon(t+i-n-1,\theta). \tag{6.13.13}$$

The whole procedure to compute $\psi(t,\theta)$ can thus be described as follows. Compute, as a first step, the filtered sequences

$$y^F(t) = \frac{1}{r(z^{-1})}\, y(t), \qquad u^F(t) = \frac{1}{r(z^{-1})}\, u(t), \qquad \varepsilon^F(t) = \frac{1}{r(z^{-1})}\, \varepsilon(t); \tag{6.13.14}$$

$\psi(t,\theta)$ is given by

$$\psi(t,\theta) = \left[\, y^F(t-n)\ \ldots\ y^F(t-1)\ u^F(t-n)\ \ldots\ u^F(t-1)\ \varepsilon^F(t-n)\ \ldots\ \varepsilon^F(t-1)\,\right]^T. \tag{6.13.15}$$

Denoting with L the length of the filtered sequences $y^F(t)$, $u^F(t)$ and $\varepsilon^F(t)$, expression (6.13.15) allows implementing the Gauss-Newton algorithm in the form

$$\theta^{k+1} = \theta^k + \left(H_\psi^T H_\psi\right)^{-1} H_\psi^T\, \varepsilon^\circ \tag{6.13.16}$$

constructing, at every step,

$$H_\psi = \begin{bmatrix} y^F(1) & \dots & y^F(n) & u^F(1) & \dots & u^F(n) & \varepsilon^F(1) & \dots & \varepsilon^F(n) \\ \vdots & & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ y^F(L-n) & \dots & y^F(L-1) & u^F(L-n) & \dots & u^F(L-1) & \varepsilon^F(L-n) & \dots & \varepsilon^F(L-1) \end{bmatrix} \tag{6.13.17}$$

and

$$\varepsilon^\circ = \begin{bmatrix} \varepsilon(n+1) & \dots & \varepsilon(L) \end{bmatrix}^T. \tag{6.13.18}$$

It is often useful to implement, instead of (6.13.18), the variation

$$\theta^{k+1} = \theta^k + \Delta_k \left(H_\psi^T H_\psi \right)^{-1} H_\psi^T \varepsilon^\circ \tag{6.13.19}$$

where Δ_k that can be linked to the variation of the parameters at step k, can be used to improve convergence.

Remark 6.13.1 – It has been shown that PEM estimates of ARMAX models are ML estimates when $w(\cdot)$ is Gaussian; this estimate can however be computed independently from the distribution of $w(\cdot)$ and can be considered as the standard estimate for ARMAX processes since it minimizes the cost function associated with the use of these models for prevision and control.

Remark 6.13.2 – It is important to remember that the polynomials $q(z^{-1})$, $p(z^{-1})$, $r(z^{-1})$, the parameters α_i, β_i, γ_i and the residuals $\varepsilon(t,\theta)$ appearing in expressions (6.13.8)–(6.13.16) refer to step k and have not been denoted as $q(z^{-1})^k$, $p(z^{-1})^k$, $r(z^{-1})^k$, α_i^k, β_i^k, γ_i^k and $\varepsilon(t,\theta^k)$ only to simplify the notation.

Remark 6.13.3 – At every step of the Gauss–Newton algorithm it is necessary to filter the sequences $u(\cdot)$, $y(\cdot)$ and $\varepsilon(\cdot\,,\theta^k)$ with a filter with transfer function $1/r(z^{-1})^k$. It is thus important to check the stability of $r(z^{-1})^k$; if an unstable polynomial is found, it is possible to substitute it with a stable one preserving the spectrum of the residuals (Bauer algorithm) or restart from the values obtained at the preceding step reducing the value of Δ_{k-1} in order to obtain, for continuity reasons, a stable update of $r(z^{-1})^{(k-1)}$.