Systems Dynamics

Course ID: 267MI – Fall 2023

Thomas Parisini
Gianfranco Fenu

University of Trieste
Department of Engineering and Architecture

267MI –Fall 2023

Lecture 13
Batch PEM Identification Algorithms

Lecture 13: Table of Contents

Least-Squares Batch Identification Algorithm

Batch PEM Identification Algorithms

PEM identification algorithms can be classified in two main categories:

- **Batch Algorithms**: observed data are elaborated in **single batch** and the determination of the model is carried out once **all data are available**
- **Recursive Algorithms**: observed data are elaborated in a recursive way **as soon as they become available** according to their temporal ordering

Least-Squares Batch Identification Algorithm

The Algorithm

Least-Squares Batch Identification Algorithm

- Recall that the first step is to chose the family of models $\mathcal{M} = \{\mathcal{M}(\vartheta) : \vartheta \in \Theta\}$ which also implies to obtain a corresponding family of predictors $\widehat{\mathcal{M}} = \left\{\widehat{\mathcal{M}}(\vartheta) : \vartheta \in \hat{\Theta}\right\}$
- Consider **ARX models**:

$$\mathcal{M}(\vartheta) : \quad A(z)\, y(t) = B(z)\, u(t-1) + \xi(t)$$

$$A(z) = 1 - a_1 z^{-1} - a_2 z^{-2} - \cdots - a_n z^{-n} \qquad \vartheta = \begin{bmatrix} a_1 \\ \vdots \\ a_n \\ b_1 \\ \vdots \\ b_n \end{bmatrix}$$

$$B(z) = b_1 + b_2 z^{-1} + \cdots + b_n z^{-n}$$

$$\widehat{\mathcal{M}}(\vartheta) : \quad \hat{y}(t) = [1 - A(z)]\, y(t) + B(z)\, u(t-1)$$

where we used the shorthand $\hat{y}(t)$ for $\hat{y}(t|t-1)$

Least-Squares Batch Identification Algorithm (cont.)

- Let us resort to the **Least-Squares** technique. Then:

$$\vartheta = \begin{bmatrix} a_1 \\ \vdots \\ a_n \\ b_1 \\ \vdots \\ b_n \end{bmatrix} \qquad \varphi(t) = \begin{bmatrix} y(t-1) \\ \vdots \\ y(t-n) \\ u(t-1) \\ \vdots \\ u(t-n) \end{bmatrix}$$

and hence

$$\begin{aligned} \mathcal{M}(\vartheta): &\quad y(t) = \varphi(t)^\top \vartheta + \xi(t) \\ \widehat{\mathcal{M}}(\vartheta): &\quad \hat{y}(t) = \varphi(t)^\top \vartheta \end{aligned}$$

where it is important to recall that the predictor has a **linear structure with respect to the vector ϑ of unknown parameters**

Least-Squares Batch Identification Algorithm (cont.)

- The prediction error is given by:

$$\varepsilon(t) = y(t) - \hat{y}(t) = y(t) - \varphi(t)^\top \vartheta$$

where $y(t)$ is the output observed variable of the true system to be identified; this variable is going to be predicted at time $t-1$ by the predictor.

- Consider the quadratic cost function:

$$J(\vartheta) = \sum_{t=1}^{N} [\varepsilon(t)]^2 = \sum_{t=1}^{N} \left[y(t) - \varphi(t)^\top \vartheta\right]^2$$

and the minimizing vector

$$\vartheta^\circ = \arg\min_{\vartheta} J(\vartheta)$$

Least-Squares Batch Identification Algorithm (cont.)

- Recalling the Least-Squares methodology and its solution:

$$\sum_{t=1}^{N} \varphi(t)\, y(t) = \left[\sum_{t=1}^{N} \varphi(t)\, \varphi(t)^\top \right] \vartheta$$

Least-Squares Normal Equations

$(2n \text{ equations},\ 2n \text{ unknowns})$

- and if $\sum_{t=1}^{N} \varphi(t)\, \varphi(t)^\top$ is **non-singular**, one gets:

$$\hat{\vartheta}_N = \left[\sum_{t=1}^{N} \varphi(t)\, \varphi(t)^\top \right]^{-1} \sum_{t=1}^{N} \varphi(t)\, y(t)$$

Least-Squares Formula

Least-Squares Batch Identification Algorithm (cont.)

- Also recall that:
  - If $\det \left[\sum_{t=1}^{N} \varphi(t)\, \varphi(t)^\top \right] \neq 0 \implies$ $\hat{\vartheta}_N$ is the unique global minimum
  - If $\det \left[\sum_{t=1}^{N} \varphi(t)\, \varphi(t)^\top \right] = 0 \implies$ $\hat{\vartheta}_N$ is one among the infinite global minima
- where the condition

$$\det \left[\sum_{t=1}^{N} \varphi(t)\, \varphi(t)^\top \right] \neq 0$$

is called **Identifiability Condition**

Least-Squares Batch Identification Algorithm

Examples

Least-Squares Batch Identification Algorithm (cont.)

It is worth noting that the LS algorithm is associated with identification of ARX models for the sake of simplicity but what matters is the **linearity with respect to the unknown parameters**.

Example 1

$$\mathcal{S}: \quad y(t) = \frac{1}{2}\, u(t-1) + \frac{1}{1+dz^{-1}}\, e(t)\,, \quad e(\cdot) \sim WN(0, \lambda^2)$$

where the only unknown is the parameter d. Hence:

$$\begin{aligned} (1+dz^{-1})\, y(t) &= \frac{1}{2}(1+dz^{-1})\, u(t-1) + e(t) \\ \implies \; y(t) &= -d\, y(t-1) + \frac{1}{2}\, u(t-1) + \frac{1}{2} d\, u(t-2) + e(t) \end{aligned}$$

Least-Squares Batch Identification Algorithm ((cont.))

This XAR model has the **structure** of a ARX(1,2) model:

$$\vartheta = \begin{bmatrix} d \\ b_1 \\ b_2 \end{bmatrix} \qquad \varphi(t) = \begin{bmatrix} y(t-1) \\ u(t-1) \\ u(t-2) \end{bmatrix}$$

However:

- to identify the original model using this ARX structure is not efficient because we do not take advantage of the information for which $b_1 = 0.5$
- Moreover, the parameters a_1, b_2 actually depend on a single parameter and also this information is not exploited.
- Finally, trying to obtain the estimate of a **single** parameter by estimating **three** parameters is not efficient as well.

Least-Squares Batch Identification Algorithm (cont.)

But the original model can be rewritten as:

$$y(t) = \frac{1}{2}\,u(t-1) + d\left[-y(t-1) + \frac{1}{2}\,u(t-2)\right] + e(t)$$

and hence

$$\tilde{y}(t) = \tilde{\varphi}(t)\,\tilde{\vartheta} + e(t) \quad \text{with} \quad \tilde{\vartheta} = d$$

where

$$\tilde{\varphi}(t) = -y(t-1) + \frac{1}{2}\,u(t-2)\,, \quad \tilde{y}(t) = y(t) - \frac{1}{2}\,u(t-1)$$

Least-Squares Batch Identification Algorithm (cont.)

Example 2

Assume that the true system to be identified takes on the form of a **nonlinear** model:

$$\mathcal{S}: \quad y(t) = a\, y(t-1)^2 + b_1 u(t-3) + b_2 u(t-5)^3 + e(t)\,,\; e(\cdot) \sim WN(0, \lambda^2)$$

However, letting

$$\vartheta = \begin{bmatrix} a \\ b_1 \\ b_2 \end{bmatrix}, \qquad \varphi(t) = \begin{bmatrix} y(t-1)^2 \\ u(t-3) \\ u(t-5)^3 \end{bmatrix}$$

we obtain a linear structure

$$y(t) = \varphi(t)^\top \vartheta + e(t)$$

and, again, we are able to proceed in the usual way

Least-Squares Batch Identification Algorithm

Asymptotic Analysis of the Algorithm

Asymptotic Analysis of the LS Batch Identification Algorithm

- In general, we have seen that in PEM methods, under suitable assumptions, the estimate asymptotically converges to the set Δ of minima of the function $\bar{J}(\vartheta) = E\,\left\{[\varepsilon(t)]^2\right\}$
- The function $\bar{J}(\vartheta)$ can be evaluated only by using the knowledge of the true system $\mathcal{S}$
- Suppose that $\exists\,\vartheta^\circ : \mathcal{S} = \mathcal{M}(\vartheta^\circ)$ which, in our case, means to assume that there exists ϑ° (true parametrization) such that:

$$\mathcal{S} : \quad y(t) = \varphi(t)^\top \vartheta^\circ + \xi(t)\,, \quad \xi(\cdot) \sim WN(0, \lambda^2)$$

- If $\mathcal{S}$ is as. stable (zeroes of $A(z)$ with $|\cdot| < 1$) then, the stationarity of $u(\cdot)$ and of $\xi(\cdot)$ implies the stationarity of $y(\cdot)$

Asymptotic Analysis of the LS Batch Identification Algorithm (cont.)

- The prediction error is given by:

$$\varepsilon(t) = \varphi(t)^\top \, (\vartheta^\circ - \vartheta) + \xi(t)$$

But $\varphi(t)^\top \, (\vartheta^\circ - \vartheta)$ is a scalar and hence it is equal to its transpose:

$$\begin{aligned}
\varepsilon(t)^2 &= (\vartheta^\circ - \vartheta)^\top \, \varphi(t)\,\varphi(t)^\top \, (\vartheta^\circ - \vartheta) + \xi(t)^2 + 2\,(\vartheta^\circ - \vartheta)^\top \, \varphi(t)\,\xi(t) \\
\implies \; E\left[\varepsilon(t)^2\right] &= (\vartheta^\circ - \vartheta)^\top \, E\left[\varphi(t)\,\varphi(t)^\top\right] \, (\vartheta^\circ - \vartheta) + E\left[\xi(t)^2\right] \\
&\quad +2\,(\vartheta^\circ - \vartheta)^\top \, E\left[\varphi(t)\,\xi(t)\right] \\
\implies \; E\left[\varepsilon(t)^2\right] &= (\vartheta^\circ - \vartheta)^\top \, E\left[\varphi(t)\,\varphi(t)^\top\right] \, (\vartheta^\circ - \vartheta) + \lambda^2
\end{aligned}$$

  - If $E\left[\varphi(t)\,\varphi(t)^\top\right] > 0$: The LS algorithm **converges a.s. to the true parametrization**
  - If $E\left[\varphi(t)\,\varphi(t)^\top\right] \geq 0$: **Identifiability does not hold**

Asymptotic Analysis of the LS Batch Identification Algorithm (cont.)

- Let us now evaluate the asymptotic variance of the estimate:

$$\psi(t,\vartheta)^\top = -\frac{\partial}{\partial\vartheta}\varepsilon_\vartheta(t) = -\frac{\partial}{\partial\vartheta}\left[\varphi(t)^\top\,(\vartheta^\circ - \vartheta) + \xi(t)\right] = \varphi(t)^\top$$

and observe that, due to linearity in the parameters, $\psi(t,\vartheta)^\top$ does not depend on ϑ. Hence:

$$\bar{R} = E\left[\varphi(t)\,\varphi(t)^\top\right]$$

which implies that for large values of N the variance of the estimate is
$\frac{\lambda^2}{N}\,E\left[\varphi(t)\,\varphi(t)^\top\right]^{-1}$
Computing the **empirical estimates**, one gets:

$$\text{var}\left[\hat{\vartheta}_N\right] = \frac{\lambda^2}{N}\left[\frac{1}{N}\sum_{t=1}^{N}\varphi(t)\,\varphi(t)^\top\right]^{-1} = \lambda^2\,S(N)^{-1} \quad (\star)$$

Remark: $(\star)$ **only** holds if $\exists\,\vartheta^\circ : \mathcal{S} = \mathcal{M}(\vartheta^\circ)$

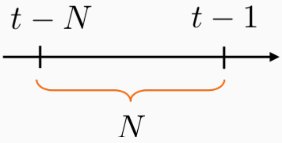
Least-Squares Batch Identification Algorithm

Operational Procedure

Operational Batch LS Identification Procedure

- Set the order of the ARX model to be identified

  and from the observed data $u(\cdot)$ and $y(\cdot)$ build the regression vector $\varphi(\cdot)$



- Perform a singularity test on matrix $S(N)$
- If $S(N) > 0$ compute $\hat{\vartheta}_N = [S(N)]^{-1} \sum_{t=1}^{N} \varphi(t)\, y(t)$
- Evaluate the estimate uncertainty $\mathrm{var}\,[\hat{\vartheta}_N] = \hat{\lambda}^2\, S(N)^{-1}$ where $\hat{\lambda}^2$ is an empirical estimate of λ^2
- Check the witheness of the prediction error $\varepsilon(t) = y(t) - \varphi(t)^\top\, \hat{\vartheta}_N$ which is of fundamental importance to verify the “goodness” of the identified model (order and structure).

Least-Squares Batch Identification Algorithm

Persistency of Excitation

Persistency of Excitation

- Let us analyze the matrix $S(N) = \sum_{t=1}^{N} \varphi(t)\,\varphi(t)^\top$ and, just to get more insight, let us focus on the simple ARX(1,1) case:

$$\vartheta = \begin{bmatrix} a_1 \\ b_1 \end{bmatrix} \qquad \varphi(t) = \begin{bmatrix} y(t-1) \\ u(t-1) \end{bmatrix}$$

$$\implies \varphi(t)\,\varphi(t)^\top = \begin{bmatrix} y(t-1)^2 & y(t-1)u(t-1) \\ u(t-1)y(t-1) & u(t-1)^2 \end{bmatrix}$$

$$\implies S(N) = \begin{bmatrix} \sum_{t=1}^{N} y(t-1)^2 & \sum_{t=1}^{N} y(t-1)u(t-1) \\ \sum_{t=1}^{N} u(t-1)y(t-1) & \sum_{t=1}^{N} u(t-1)^2 \end{bmatrix}$$

Notice that the elements of the matrix $S(N)$ diverge for $N \to \infty$

Persistency of Excitation (cont.)

- Notice that $\mathrm{rank}\left[\varphi(t)\,\varphi(t)^\top\right] = 1\,,\ \forall\,\varphi(t)$ and hence $S(1)$ is non-singular only if $\dim\left[\varphi(t)\right] = 1$ (only one parameter to be estimated).
  Hence:

  given the model's complexity, the data cardinality has to be large enough

- It is convenient to introduce

$$R(N) = \frac{1}{N}\,S(N)$$

$$\implies\ \hat{\vartheta}_N = \left[R(N)\right]^{-1}\frac{1}{N}\sum_{t=1}^{N}\varphi(t)\,y(t)$$

Persistency of Excitation (cont.)

- In the ARX(1,1) case under consideration:

$$R(N) = \begin{bmatrix} \frac{1}{N}\sum_{t=1}^{N} y(t-1)^2 & \frac{1}{N}\sum_{t=1}^{N} y(t-1)u(t-1) \\ \frac{1}{N}\sum_{t=1}^{N} u(t-1)y(t-1) & \frac{1}{N}\sum_{t=1}^{N} u(t-1)^2 \end{bmatrix} \longrightarrow \bar{R}$$

where $\bar{R} = \begin{bmatrix} \gamma_{yy}(0) & \gamma_{uy}(0) \\ \gamma_{yu}(0) & \gamma_{uu}(0) \end{bmatrix}$

Persistency of Excitation (cont.)

- In the general case $ARX(n_a, n_b)$:

$$\bar{R} = \begin{bmatrix} \bar{R}_{yy}^{n_a} & \bar{R}_{yu} \\ \bar{R}_{uy} & \bar{R}_{uu}^{n_b} \end{bmatrix}$$

where

$$\bar{R}_{yy}^{n_a} = E\left\{ \begin{bmatrix} y(t-1) \\ \vdots \\ y(t-n_a) \end{bmatrix} \begin{bmatrix} y(t-1) & \cdots & y(t-n_a) \end{bmatrix} \right\}$$

$$\bar{R}_{uu}^{n_b} = E\left\{ \begin{bmatrix} u(t-1) \\ \vdots \\ u(t-n_b) \end{bmatrix} \begin{bmatrix} u(t-1) & \cdots & u(t-n_b) \end{bmatrix} \right\}$$

and so on.

Persistency of Excitation (cont.)

- Hence, the positive definiteness of $R(N)$ is the condition to be satisfied in order to obtain a **unique estimate** at least for a sufficiently large number N of observed data
- Consider the **Sylvester test**: a symmetric square matrix A is positive definite if and only if all principal minors are positive, that is, if and only if:

$$\begin{aligned}
D_1 &= \det\left(a_{11}\right) > 0 \\
D_2 &= \det\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} > 0 \\
D_3 &= \det\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} > 0 \\
&\vdots \\
D_n &= \det\left(A\right) > 0
\end{aligned}$$

Persistency of Excitation (cont.)

- Hence $\bar{R} > 0 \implies \bar{R}_{uu}^{n_b} > 0$ that is $\bar{R}_{uu}^{n_b} > 0$ is a **necessary condition** for $\bar{R}$ to be non-singular
- In general, for a generic n, we have:

$$\bar{R}_{uu}^{n} = \begin{bmatrix} \gamma_{uu}(0) & \gamma_{uu}(1) & \cdots & \gamma_{uu}(n-1) \\ \gamma_{uu}(1) & \gamma_{uu}(0) & \cdots & \gamma_{uu}(n-2) \\ & \ddots & \ddots & \\ \gamma_{uu}(n-1) & \cdots & \gamma_{uu}(1) & \gamma_{uu}(0) \end{bmatrix}$$

which is a Toeplitz matrix (all elements on the diagonals coincide) and depends only on $u(\cdot)$ hence on the **experimental conditions**.

Persistency of Excitation (cont.)

Persistency of Excitation

Definition. The input variable $u(\cdot)$ is persistently exciting of order n if $\bar{R}_{uu}^{n}$ is non-singular.

A **necessary condition** to be able to identity a $ARX(n_a, n_b)$ model is that the input $u(\cdot)$ is persistently exciting of order n_b

Remark. From the Sylvester test it turns out that if $u(\cdot)$ is persistently exciting of order n then it is p.e. of order $\tilde{n}, \forall\, \tilde{n} < n$

Least-Squares Identifiability

Least-Squares Identifiability

The Case of ARX Models

LS Identifiability in the Case of ARX Models

Recall that:

- To analyze the identifiability of a given system $\mathcal{S}$ through a given class of models $\mathcal{M}$ means to analyze the **cardinality of the set** Δ
- In general:

$$\left.\begin{array}{l}\textbf{Experimental conditions}\\[2em]\textbf{Structure of the class of models}\end{array}\right\} \Longrightarrow \quad \textbf{Cardinality of } \Delta$$

- In our case, we want to analyze the identifiability of a given system $\mathcal{S}$ by a given family of models $\mathcal{M} = ARX(n_a, n_b)$

LS Identifiability in the Case of ARX Models (cont.)

- If we are allowed to design the identification experimental conditions, we have to make sure that $u(\cdot)$ is **sufficiently rich** so as to guarantee that Δ contains only one element.
- If the experimental conditions cannot be designed, the **complexity of the models** (that is, the number of parameters to be identified) **has to be reduced** by limiting ourselves to identify what is actually identifiable for the given the experimental context
- In our case $\mathcal{M} = ARX(n_a, n_b)$, $u(\cdot)$ **sufficiently rich** means $u(\cdot)$ p.e. of order n_b
- Observe that $u(\cdot) = WN(0, \lambda^2)$ is p.e. of arbitrary order because, in this case, $\bar{R}^n_{uu}$ is a **diagonal** matrix. This is not necessarily the best choice. The important point is to make sure to design input variables $u(\cdot)$ with a **suitable spectrum exciting all the system's modes of behaviour**.

Least-Squares Identifiability

Structure of the Family of Models

Identifiability: Structure of the Family of Models to be Identified

Recall that:

- Assume that $\mathcal{S} \in \mathcal{M}$ but also that the chosen family has a **complexity larger than the one of the true system**

  Example $\mathcal{S} = ARMAX(1,1,1)\,, \quad \mathcal{M} = ARMAX(2,2,2)$
  Clearly, irrespective of the experimental conditions, Δ will be necessarily made of an infinite number of elements because $\mathcal{S}$ can be described by an infinite number of models belonging to the family in which there are **common factors**.

  It is important to guarantee that the family $\mathcal{M}$ **is not over-parametrised**

- In our considered case $\mathcal{M} = ARX(n_a, n_b)$, having a structural non-identifiability means that $\bar{R}$ is singular despite the fact that $\bar{R}_{uu}^{n_b} > 0$

LS Identifiability: Summing Up

- Suppose that:
  - $\mathcal{S}$ is $ARX(n_a, n_b)$ with with no common factors between $A(z)$ and $B(z)$
  - $\mathcal{M} = ARX(n_a, n_b)$
  - $u(\cdot)$ p.e. of order n_b

  Then, the estimates of the parameters of the $ARX(n_a, n_b)$ model converge a.s. to the true parametrization
- If $u(\cdot)$ is not p.e. of order n_b and the estimate does not converge even for large values of N very likely the complexity of the model to be identified should be reduced.
- If the estimate does converge but the prediction error $\varepsilon(\cdot)$ **is not white** this means that the family of models $\mathcal{M} = ARX(n_a, n_b)$ is not adequate; hence **either the order or the family itself** has to be changed.

Least-Squares Identifiability

Example

Important Example

- Consider a system to be identified which can be described by a **ARMAX(1,1,1)** model:

$$\mathcal{S}: \quad y(t) = a^\circ\, y(t-1) + b^\circ\, u(t-1) + \xi(t) + c^\circ\, \xi(t-1)$$
$$|a^\circ| < 1\,, \quad \xi(\cdot) \sim WN(0,\lambda^2)\,, \quad u(\cdot) \sim WN(0,\mu^2)$$

  where the processes $u(\cdot)$ and $\xi(\cdot)$ are supposed to be uncorrelated.

- Let us consider the **ARX(1,1)** family of models:

$$\widehat{\mathcal{M}}: \quad \hat{y}(t) = a\, y(t-1) + b\, u(t-1)$$

  and let us use the LS algorithm to identify the system $\mathcal{S}$ by a ARX model.

Important Example (cont.)

The asymptotic theory ensures the almost sure convergence to one of the minima of the function

$$\begin{aligned}\bar{J}(\vartheta) = E\,\left\{[\varepsilon(t)]^2\right\} &= E\,\left\{[y(t) - \hat{y}(t)]^2\right\} \\ &= E\,[y(t)]^2 + E\,[\hat{y}(t)]^2 - 2E\,[y(t)\,\hat{y}(t)]\end{aligned}$$

Hence:

$$E\,[\hat{y}(t)]^2 = a^2\,E\,[y(t-1)]^2 + b^2\,E\,[u(t-1)]^2 + 2\,a\,b\,E\,[y(t-1)\,u(t-1)]$$

But $y(t-1)$ depends on $u(t-2)$, $y(t-2)$, $\xi(t-1)$ and hence, given our hypotheses, we have $E\,[y(t-1)\,u(t-1)] = 0$ and then

$$E\,[\hat{y}(t)]^2 = a^2\,\gamma_{yy}(0) + b^2\,\gamma_{uu}(0)$$

Moreover:

$$\begin{aligned}E\,[y(t)\,\hat{y}(t)] &= a\,E\,[y(t)\,y(t-1)] + b\,E\,[y(t)\,u(t-1)] \\ &= a\,\gamma_{yy}(1) + b\,\gamma_{uy}(1)\end{aligned}$$

Important Example (cont.)

Thus:

$$\bar{J}(\vartheta) = \left(1 + a^2\right)\,\gamma_{yy}(0) + b^2\,\gamma_{uu}(0) - 2\,a\,\gamma_{yy}(1) - 2\,b\,\gamma_{uy}(1)$$

and hence:

$$\frac{\partial \bar{J}}{\partial \vartheta} = \begin{bmatrix} \frac{\partial \bar{J}}{\partial a} & \frac{\partial \bar{J}}{\partial b} \end{bmatrix} = \left[2a\,\gamma_{yy}(0) - 2\,\gamma_{yy}(1) \mid 2b\,\gamma_{uu}(0) - 2\,\gamma_{uy}(1)\right]$$

$$\Longrightarrow \quad \bar{\vartheta} = \begin{bmatrix} \bar{a} \\ \bar{b} \end{bmatrix} = \begin{bmatrix} \dfrac{\gamma_{yy}(1)}{\gamma_{yy}(0)} \\[2ex] \dfrac{\gamma_{uy}(1)}{\gamma_{uu}(0)} \end{bmatrix}$$

Important Example (cont.)

Now, plugging in the information on the “true” system we obtain:

$$\begin{aligned}\gamma_{uy}(1) = E\left[y(t)\,u(t-1)\right] = E\,\big[&a^\circ\,y(t-1)\,u(t-1) + b^\circ\,u(t-1)^2\\ &+\xi(t)\,u(t-1) + c^\circ\,\xi(t-1)\,u(t-1)\big]\\ =\;& b^\circ\,\gamma_{uu}(0)\end{aligned}$$

$$\begin{aligned}\gamma_{yy}(1) = E\left[y(t)\,y(t-1)\right] = E\,\big[&a^\circ\,y(t-1)^2 + b^\circ\,u(t-1)\,y(t-1)\\ &+\xi(t)\,y(t-1) + c^\circ\,\xi(t-1)\,y(t-1)\big]\\ =\;& a^\circ\,\gamma_{yy}(0) + c^\circ\,\lambda^2\end{aligned}$$

Hence:

$$\bar{a} = \frac{\gamma_{yy}(1)}{\gamma_{yy}(0)} = \frac{a^\circ\,\gamma_{yy}(0) + c^\circ\,\lambda^2}{\gamma_{yy}(0)} = a^\circ + c^\circ\,\frac{\operatorname{var}(\xi)}{\operatorname{var}(y)}$$

$$\bar{b} = b^\circ$$

Important Example (cont.)

- Summing up, we got:

$$\hat{\vartheta}_N = \begin{bmatrix} \hat{a}_N \\ \hat{b}_N \end{bmatrix} \longrightarrow \begin{bmatrix} a^\circ + c^\circ \dfrac{\operatorname{var}(\xi)}{\operatorname{var}(y)} \\ b^\circ \end{bmatrix} \quad \text{a.s.}$$

and then the estimation error of the true parameter a°, for a given c°, is **inversely proportional to the signal/noise ratio**.
Moreover, the true value can only be obtained for $c^\circ = 0$ or $\operatorname{var}(\xi) = 0$ and then only in the case in which the ARMAX model is actually ARX or deterministic.

Important Example (cont.)

- Prediction error:

$$
\begin{aligned}
\varepsilon(t) &= y(t) - \hat{y}(t) = y(t) - \bar{a}\, y(t-1) - \bar{b}\, u(t-1) \\
&= a^\circ\, y(t-1) + b^\circ\, u(t-1) + \xi(t) + c^\circ\, \xi(t-1) \\
&\qquad - \left(a^\circ + c^\circ \frac{\operatorname{var}(\xi)}{\operatorname{var}(y)}\right) y(t-1) - b^\circ\, u(t-1) \\
&= \xi(t) + c^\circ\, \xi(t-1) - c^\circ \frac{\operatorname{var}(\xi)}{\operatorname{var}(y)}\, y(t-1)
\end{aligned}
$$

which **is not white**, except in the case $c^\circ = 0$

Choice of Models Complexity

Choice of Models Complexity

Whiteness Test

Premise: Anderson Whiteness Test

- The results of the identification procedure have to be checked a posteriori verifying that the prediction error is as much similar as possible to a **white process**.
- Given a zero-mean stationary process $\varepsilon(\cdot)$ consider the empirical estimate of the covariance function:

  $$\hat{\gamma}(\tau) = \frac{1}{N} \sum_{t=1}^{N-\tau} \varepsilon(t)\, \varepsilon(t+\tau)$$

  where N is the length of the considered time-horizon.
- The *Anderson test* makes use of the **normalized** empirical covariance function:

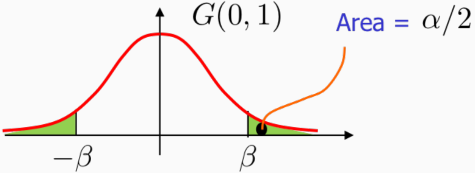
  $$\hat{\rho}(\tau) = \frac{\hat{\gamma}(\tau)}{\hat{\gamma}(0)}$$

- It can be shown that if $\varepsilon(\cdot)$ is white, then $\sqrt{N}\,\hat{\rho}(\tau) \sim \mathrm{As}\, G(0,1)$ and that $\hat{\rho}(i)$ is **asymptotically uncorrelated** with $\hat{\rho}(j),\ i \neq j$

Premise: Anderson Whiteness Test (cont.)

- Set a confidence level $0 < \alpha < 1$ (for example $\alpha = 0.01$)

  Moreover, determine $\beta > 0$ such that the tails of the Gaussian $G(0,1)$ in the intervals $(-\infty, -\beta)$ and $(\beta, +\infty)$ have area $\alpha/2$.



- Consider a certain number M of evaluations of $\hat{\rho}(\tau)$: $\hat{\rho}(0), \hat{\rho}(1), \hat{\rho}(2), \ldots, \hat{\rho}(M)$
- Consider the interval $(-\beta/\sqrt{N}, \beta/\sqrt{N})$ and evaluate the number n of samples of $\hat{\rho}(\tau)$ such that $\hat{\rho}(\tau) \notin (-\beta/\sqrt{N}, \beta/\sqrt{N})$
- If $\dfrac{n}{M} < \alpha$ then $\varepsilon(\cdot)$ is considered white with confidence α

LS Batch Identification Algorithm - Example

Matlab live script

An interactive **Matlab live script** illustrates how to solve the identification problem, given the proper model of a stochastic process.

Steps to retrieve the live script:

- Download as a ZIP archive the whole contents of the folder named ”**Lecture13**,” available in the ”**Class Materials**” file area of the MS Teams course team, and uncompress it in a preferred folder.
- Add the chosen folder and subfolders to the Matlab path.
- Open the live script using the Matlab command:

  ```
  open('L13_LS_batch_ID_alg_example.mlx');
  ```

- Explore the live script and run it.

Choice of Models Complexity

Model Validation

Model Complexity

- Let us characterize the complexity of the model (for a given specific family of models) with the total number n of its parameters
- Consider the quadratic criterion:

$$J(\vartheta) = \frac{1}{N}\sum_{i=1}^{N}[\varepsilon(t)]^2$$

where ϑ is the vector of unknown parameters, $n = \dim(\vartheta)$ and $\varepsilon(t)$ is the prediction error at time instant t: $\varepsilon(t) = y(t) - \hat{y}(t\,|\,t-1)$

- Consider:

$$\hat{\vartheta}_N = \arg\min_{\vartheta} J(\vartheta)$$

- Moreover $J(\hat{\vartheta}_N)$ can be interpreted as an index quantifying the “data interpretation” capabilities of the model
- **For a given realization of the observed data**, $J(\hat{\vartheta}_N)$ decreases as the model complexity n increases and hence $J(\hat{\vartheta}_N)$ is not per se useful to determine the optimal model complexity

Important Example

Consider the process (“true” system):

$$\begin{aligned}\mathcal{S}: \quad & y(t) = 1.2\, y(t-1) - 0.32\, y(t-2) + u(t-1) + 0.5\, u(t-2) + e(t) \\ & e(\cdot) \sim WN(0,1)\,,\; u(\cdot) \sim WN(0,4)\,,\; e(\cdot), u(\cdot) \text{ uncorrelated}\end{aligned}$$

Consider the family of models $ARX(n,n)$:

$$\begin{aligned}\mathcal{M}(\vartheta): \quad y(t) = {} & a_1\, y(t-1) + \cdots + a_n\, y(t-n) \\ & + b_1\, u(t-1) + \cdots + b_n\, u(t-n) + \xi(t)\end{aligned}$$

and let us identify the models in the cases $n = 1, 2, 3$ over a window of 2000 data, that is $\{u(t), y(t)\}_{t=1,\ldots,2000}$

Important Example (cont.)

ARX(1,1)	$\hat{a} = 0.932 \quad (0.6\%)$ $\hat{b} = 0.975 \quad (2.3\%)$ $J = 3.864$ T.And. 5% : 7		
ARX(2,2)	$\hat{a}_1 = 1.204 \quad (1\%)$ $\hat{b}_1 = 0.984 \quad (1\%)$ $J = 0.998$ T.And. 5% : 0 (OK)	$\hat{a}_2 = -0.32 \quad (3\%)$ $\hat{b}_2 = 0.485 \quad (3\%)$	
ARX(3,3)	$\hat{a}_1 = 1.194 \quad (2\%)$ $\hat{b}_1 = 0.984 \quad (1\%)$ $J = 0.997$ T.And. 5% : 0 (OK)	$\hat{a}_2 = -0.299 \quad (10\%)$ $\hat{b}_2 = 0.494 \quad (5\%)$	$\hat{a}_3 = -0.019 \quad (68\%)$ $\hat{b}_3 = -0.016 \quad (120\%)$

Important Example (cont.)

Observations

- Observe that $J(\hat{\vartheta}_{2000})$ decreases when n increases
- The Anderson test provides results that improve when n increases
- For $n \geq 3$ the estimates of the parameters $\hat{a}_n$ and $\hat{b}_n$ are very small and the uncertainties associated with the parameters estimates are very large which is a clear sign of over-parametrization (the model is too complex with respect to the available data)
- In a situation like the one in this example it is possible to conclude that $ARX(2, 2)$ is the correct model. However, in general this is hardly possible

General Remarks

- In general, the Anderson test may not be satisfied even for very large values of n and, in such a case, it is not possible to come up with a clear choice as far as the model order is concerned (as in the example)
- The fact that, for a given observed data realization, $J(\hat{\vartheta}_N)$ decreases when the model complexity n increases – thus avoiding the possibility to use $J(\hat{\vartheta}_N)$ to determine the model complexity – is a direct consequence of a conceptual mistake:

  Use of the same batch of data
  to identify and to validate the model

  Hence, $J(\hat{\vartheta}_N)$ is generally not an indicator of the "goodness" of the identified model

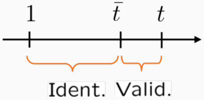
Model Validation

It is necessary to validate the model on data that are different from the ones used to identify the model

Choice of Models Complexity

Cross-Validation

Cross-Validation



- Assume the availability of a sufficiently large batch of N observed data;
- Reserve a part of the batch of data to validate the model that has been identified with the remaining data

- Consider a **cross-validation cost function**:

$$J_{CV}(\vartheta) = \frac{1}{N-\bar{t}} \sum_{k=\bar{t}}^{N} [\varepsilon(k)]^2$$

  and evaluate n such that $J_{CV}(\vartheta)$ is minimized
- For a given batch of observed data, $J_{CV}(\hat{\vartheta}_{\bar{t}})$ **is NOT monotonically decreasing** with respect to the increase of the complexity n and hence $J_{CV}(\hat{\vartheta}_{\bar{t}})$ can be used to decide the optimal complexity of the model
- The CV procedure is rather cumbersome and needs a **large batch of data** to be applicable in an effective way.

Choice of Models Complexity

Final Prediction Error

Final Prediction Error (FPE)

- Let us devise a criterion by which to evaluate the goodness of the model with respect to different realizations of the batch of observed data:

$$\bar{J}(\vartheta) = E\left\{[y(t,s) - \hat{y}(t,s,\vartheta)]^2\right\}$$

  where s is the outcome of the random experiment concerning the data observation
- Hence $\bar{J}(\vartheta)$ characterizes the **average adherence** of the model on all possible data batches.
- As usual we have $\hat{\vartheta}_N = \arg\min_{\vartheta} J(\vartheta)$ where the minimization is carried out on a given specific data batch. Clearly, when considering all possible data realizations, we have $\hat{\vartheta}_N = \hat{\vartheta}_N(s)$
- Averaging again, we define

$$\text{FPE} = E\left\{\bar{J}\left[\hat{\vartheta}_N(s)\right]\right\}$$

  and the optimal model complexity is the one for which FPE is minimized

Final Prediction Error (FPE) (cont.)

Let us evaluate the FPE in a simple/specific case:

$$\mathcal{S}: \quad AR(n) \quad \text{and} \quad \mathcal{M}: \quad AR(n)$$

Then:

$$\begin{aligned} \mathcal{S}: &\quad y(t,s) = \varphi(t,s)^\top \vartheta^\circ + \xi(t) \quad \xi(\cdot) \sim WN(0,\lambda^2) \\ \widehat{\mathcal{M}}(\vartheta): &\quad \hat{y}(t,s) = \varphi(t,s)^\top \vartheta \end{aligned}$$

But $\varphi(t,s)$ and $\xi(t)$ are uncorrelated and hence

$$\begin{aligned} \bar{J}(\vartheta) = E\left\{[y(t,s) - \hat{y}(t,s,\vartheta)]^2\right\} &= E\left\{\left[\varphi(t,s)^\top \, (\vartheta^\circ - \vartheta) + \xi(t)\right]^2\right\} \\ &= (\vartheta^\circ - \vartheta)^\top \, E\left[\varphi(t,s)\,\varphi(t,s)^\top\right] \, (\vartheta^\circ - \vartheta) + \lambda^2 \end{aligned}$$

Setting $\bar{R} = E\left[\varphi(t,s)\,\varphi(t,s)^\top\right]$ we get

$$\text{FPE} = E\left\{\bar{J}\left[\hat{\vartheta}_N(s)\right]\right\} = E\left\{\left[\vartheta^\circ - \hat{\vartheta}_N(s)\right]^\top \bar{R}\left[\vartheta^\circ - \hat{\vartheta}_N(s)\right] + \lambda^2\right\}$$

Final Prediction Error (FPE) (cont.)

On the other hand, for a sufficiently large N:

$$\operatorname{var}\left[\vartheta^{\circ} - \hat{\vartheta}_N(s)\right] \sim \frac{\lambda^2}{N}\,\bar{R}^{-1}$$

Now, setting $\nu = \vartheta^{\circ} - \hat{\vartheta}_N(s)$ we have:

$$\operatorname{var}(\nu) = \frac{\lambda^2}{N}\,\bar{R}^{-1} \implies \bar{R} = \operatorname{var}(\nu)^{-1}\,\frac{\lambda^2}{N}$$

and then

$$\text{FPE} = E\left(\nu^{\top}\,\bar{R}\,\nu\right) + \lambda^2 = E\left[\nu^{\top}\operatorname{var}(\nu)^{-1}\,\nu\right]\frac{\lambda^2}{N} + \lambda^2$$

But $\nu^{\top}\operatorname{var}(\nu)^{-1}\,\nu$ is a scalar and hence:

$$\nu^{\top}\operatorname{var}(\nu)^{-1}\,\nu = \operatorname{tr}\left[\nu^{\top}\operatorname{var}(\nu)^{-1}\,\nu\right]$$

Final Prediction Error (FPE) (cont.)

Therefore (using $\mathrm{tr}\;(AB) = \mathrm{tr}\;(BA)$):

$$\begin{aligned} E\;\left[\nu^\top \operatorname{var}(\nu)^{-1}\,\nu\right] &= E\;\left\{\mathrm{tr}\;\left[\nu^\top \operatorname{var}(\nu)^{-1}\,\nu\right]\right\} \\ &= E\;\left\{\mathrm{tr}\;\left[\operatorname{var}(\nu)^{-1}\,\nu\,\nu^\top\right]\right\} \\ &= \mathrm{tr}\;\left\{E\;\left[\operatorname{var}(\nu)^{-1}\,\nu\,\nu^\top\right]\right\} \\ &= \mathrm{tr}\;\left[\operatorname{var}(\nu)^{-1}\,E\;\left(\nu\,\nu^\top\right)\right] \\ &= \mathrm{tr}\;\left[\operatorname{var}(\nu)^{-1}\,\operatorname{var}(\nu)\right] \\ &= \mathrm{tr}\;(I) = n \end{aligned}$$

Thus:

$$\mathrm{FPE} = \frac{n}{N}\,\lambda^2 + \lambda^2$$

Final Prediction Error (FPE) (cont.)

For a sufficiently large value of N, an estimate of λ^2 is

$$\hat{\lambda}^2 = \frac{1}{N-n}\sum_{t=1}^{N}[\varepsilon(t)]^2 = \frac{N}{N-n}\,\frac{1}{N}\sum_{t=1}^{N}[\varepsilon(t)]^2 = \frac{N}{N-n}\,J(\hat{\vartheta}_N)^{(n)}$$

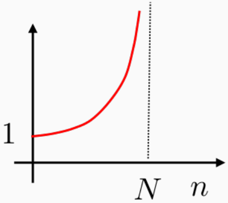
where $J(\hat{\vartheta}_N)^{(n)}$ denotes the specific value of the cost on the given observed data on the model of complexity n.

The final form of the FPE is thus given by:

$$\mathrm{FPE} = \frac{N+n}{N-n}\,J(\hat{\vartheta}_N)^{(n)}$$

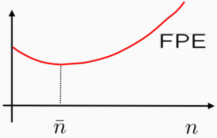
Final Prediction Error (FPE): Remarks

- The function $\dfrac{N+n}{N-n}$ behaves like in the figure, whereas the function $J(\hat{\vartheta}_N)^{(n)}$ is monotonically decreasing with n



- Hence, for a given N, the typical FPE behavior is shown in the figure on the right.

  Thus, the optimal complexity with respect to the FPE criterion is $\bar{n}$



- The FPE formula holds for other families of models just suitably re-defining n. For example, in the ARX case, we set $n = n_a + n_b$ while in the ARMAX case we set $n = n_a + n_b + n_c$

Choice of Models Complexity

Akaike Information Criterion

Akaike Information Criterion (AIC)

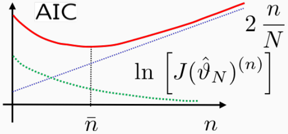
- This is a **statistical criterion**. It is obtained by minimizing the **Kullback distance** between the probability density function of the observed data and the one that would be generated by the model under concern. The Kullback distance is defined as

$$E\left(\ln \frac{p_{\text{true}}}{p_{\text{model}}}\right)$$

- It can be shown that

$$\text{AIC} = 2\,\frac{n}{N} + \ln\left[J(\hat{\vartheta}_N)^{(n)}\right]$$

- Again, the optimal complexity with respect to the AIC criterion is $\bar{n}$



- Notice that the rate of growth of the linear term $2\,\frac{n}{N}$ decreases with N. Hence, AIC “suggests” models of smaller order in presence of fewer observed data.

Choice of Models Complexity

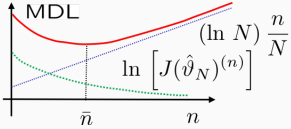
Minimum Description Length

Minimum Description Length (MDL)

- This is an information-theory based criterion: **for a given set of data, the optimal complexity is the one for which the model can be "described" by the minimum number of bits.**
- Taking into account that the growth of the dimension of the vector of parameters is compensated by the (average) decrease of the number of bits that are needed to describe the prediction error. It can be shown that

$$\mathrm{MDL} = (\ln\ N)\,\frac{n}{N} + \ln\left[J(\hat{\vartheta}_N)^{(n)}\right]$$

- Again, the optimal complexity with respect to the MDL criterion is $\bar{n}$



Choice of Models Complexity

Comparison Between Indexes

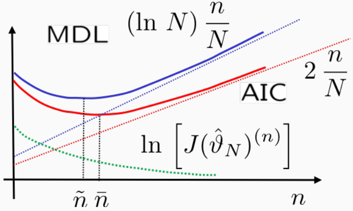
Comparison Between FPE, AIC and MDL

- For large N, FPE and AIC typically yield very similar outcomes:

$$\begin{aligned}\ln \mathrm{FPE} &= \ln\left[\frac{N+n}{N-n}\, J(\hat{\vartheta}_N)^{(n)}\right] = \ln\left[\frac{1+n/N}{1-n/N}\, J(\hat{\vartheta}_N)^{(n)}\right]\\ &= \ln\,(1+n/N) - \ln\,(1-n/N) + \ln\left[J(\hat{\vartheta}_N)^{(n)}\right]\\ &\simeq 2\,\frac{n}{N} + \ln\left[J(\hat{\vartheta}_N)^{(n)}\right] = \mathrm{AIC}\end{aligned}$$

- AIC and MDL have a similar structure and differ for the term multiplying n: for AIC it is $2/N$ while for MDL it is $\ln N/N$

  - For large N, MDL **typically** yields models with lower complexity



- **In general there is no guarantee that the criteria have a single minimum**

Hands On - Comparison Between the Indices

Matlab live script

A couple of interactive **Matlab live scripts** illustrate how to evaluate the performance of the FPE, AIC and MDL indices and of the Cross Validation in selecting the best model from a range of possible models.

Steps to retrieve the live script:

- Download as a ZIP archive the whole contents of the folder named ”**Lecture13**,” available in the ”**Class Materials**” file area of the MS Teams course team, and uncompress it in a preferred folder.
- Add the chosen folder and subfolders to the Matlab path.
- Open the live script using the Matlab command:

```
open('L13_Order_Determination_FPE_AIC_MDL.mlx');
open('L13_Order_Determination_CV.mlx');
```

- Explore the live scripts and run them.

267MI –Fall 2023

Lecture 13

Batch PEM Identification Algorithms

END