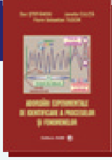


❺ Identificare parametrică prin MMEP

☞ Contextul de lucru

9 p

- Modelele de regresie liniară avînd vectorul regresorilor nemăsurabil necesită metode de identificare mai complexe.
- Astfel de modele se regăsesc atît în clasa ARMAX, cît şi (mai ales) în clasa RSISO.

Exemple

ARMAX[na,nb,nc]

$$\begin{cases} A(q^{-1})y[n] = B(q^{-1})u[n] + C(q^{-1})e[n] \\ E\{e[n]e[m]\} = \lambda^2\delta_0[n-m] \end{cases} \quad \forall\, n,m \in \mathbb{N}$$

$$\varphi^T[n] \overset{def}{=} \Big[-y[n-1]\ -y[n-2] \cdots -y[n-na] \;\Big|\; u[n-1]\ u[n-2] \cdots u[n-nb] \;\Big|\; \ldots$$

3 componente,

$$\ldots\ \boxed{e[n-1]\ e[n-2] \cdots e[n-nc]\Big]}$$

ca şi vectorul parametrilor

$$\theta^T \overset{def}{=} \Big[a_1\ a_2 \cdots a_{na} \;\Big|\; b_1\ b_2 \cdots b_{nb} \;\Big|\; c_1\ c_2 \cdots c_{nc}\Big] \quad \forall\, n \in \mathbb{N}^*$$

☹ Ultima componentă nu este măsurabilă.

BJ[nb,nc,nd,nf]

(Box Jenkins)

$$\begin{cases} y[n] = \dfrac{B(q^{-1})}{F(q^{-1})}u[n] + \dfrac{C(q^{-1})}{D(q^{-1})}e[n] \\ E\{e[n]e[m]\} = \lambda^2\delta_0[n-m] \end{cases} \quad \forall\, n,m \in \mathbb{N}$$

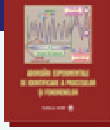
$$\varphi^T[n] \overset{def}{=} \Big[-y[n-1]\ -y[n-2] \cdots -y[n-nd-nf] \;\Big|\; u[n-1]\ u[n-2] \cdots u[n-nb-nd] \;\Big|\; \ldots$$

3 componente

$$\ldots\ \boxed{e[n-1]\ e[n-2] \cdots e[n-nc-nf]\Big]}$$

$$\theta^T \overset{def}{=} \Big[b_1\ b_2 \cdots b_{nb} \;\Big|\; c_1\ c_2 \cdots c_{nc} \;\Big|\; d_1\ d_2 \cdots d_{nd} \;\Big|\; f_1\ f_2 \cdots f_{nf}\Big] \quad \forall\, n \in \mathbb{N}^*$$

4 componente



❺ Identificare parametrică prin MMEP

☞ Contextul de lucru

2 metode de identificare

MCMMP Extinsă (MCMMPE)

mai puțin precisă, dar mai simplă

- Strategia generală de identificare:

① Estimarea zgomotului care intervine în componenta nemăsurabilă cu ajutorul unui model avînd vectorul regresorilor complet măsurabil (dar mai puțin precis).

② Determinarea parametrilor originali ai modelului cu ajutorul vectorului regresorilor avînd componenta nemăsurabilă estimată în etapa precedentă.

Metoda Minimizării Erorii de Predicție (MMEP)

mai precisă, dar mai complexă

- Strategia generală de identificare:

① Inițializarea procesului recursiv din etapa următoare folosind MCMMPE.

② Determinarea recursivă a parametrilor modelului, plecînd de la inițializarea din etapa precedentă şi folosind o metodă de optimizare (Metoda Gauss-Newton – MGN).

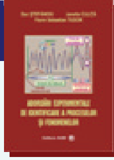
- Tipurile de modele cu care se operează în cadrul lucrării de laborator:

ARMAX[na,nb,nc]

$$\begin{cases} A(q^{-1})y[n] = B(q^{-1})u[n] + C(q^{-1})e[n] \\ E\{e[n]e[m]\} = \lambda^2\delta_0[n-m] \end{cases} \quad \forall\, n,m\in\mathbb{N}$$

BJ[nb,nc,nd,nf]

$$\begin{cases} y[n] = \dfrac{B(q^{-1})}{F(q^{-1})}u[n] + \dfrac{C(q^{-1})}{D(q^{-1})}e[n] \\ E\{e[n]e[m]\} = \lambda^2\delta_0[n-m] \end{cases} \quad \forall\, n,m\in\mathbb{N}$$

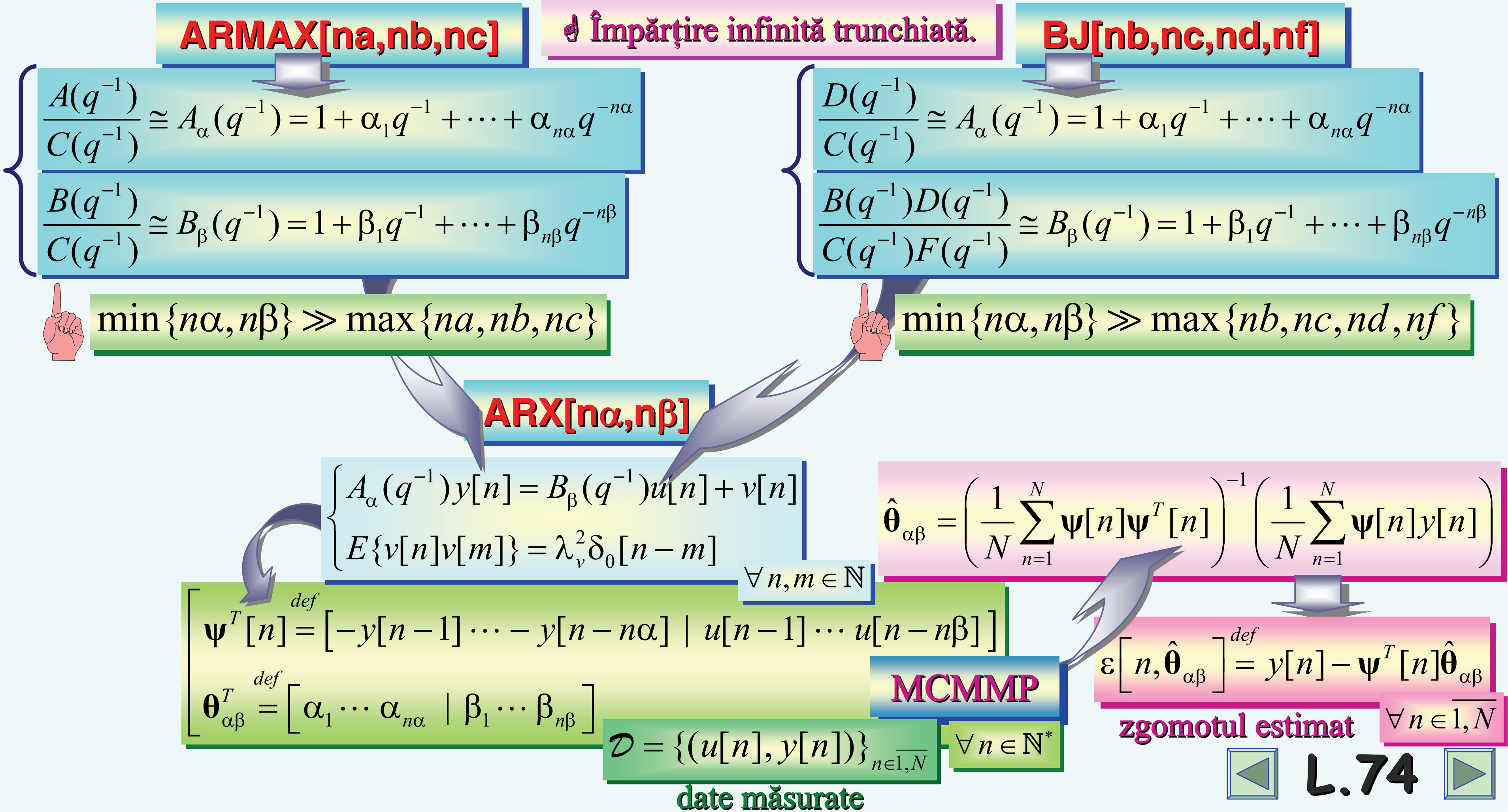


⑤ Identificare parametrică prin MMEP

☞ Contextul de lucru

MCMMPE

① Estimarea zgomotului care intervine în componenta nemăsurabilă cu ajutorul unui model aproximant de tip ARX.



ARMAX[na,nb,nc]

☝ Împărțire infinită trunchiată.

BJ[nb,nc,nd,nf]

$$\begin{cases} \dfrac{A(q^{-1})}{C(q^{-1})} \cong A_\alpha(q^{-1}) = 1 + \alpha_1 q^{-1} + \cdots + \alpha_{n\alpha} q^{-n\alpha} \\ \dfrac{B(q^{-1})}{C(q^{-1})} \cong B_\beta(q^{-1}) = 1 + \beta_1 q^{-1} + \cdots + \beta_{n\beta} q^{-n\beta} \end{cases}$$

☝ $\min\{n\alpha, n\beta\} \gg \max\{na, nb, nc\}$

$$\begin{cases} \dfrac{D(q^{-1})}{C(q^{-1})} \cong A_\alpha(q^{-1}) = 1 + \alpha_1 q^{-1} + \cdots + \alpha_{n\alpha} q^{-n\alpha} \\ \dfrac{B(q^{-1})D(q^{-1})}{C(q^{-1})F(q^{-1})} \cong B_\beta(q^{-1}) = 1 + \beta_1 q^{-1} + \cdots + \beta_{n\beta} q^{-n\beta} \end{cases}$$

☝ $\min\{n\alpha, n\beta\} \gg \max\{nb, nc, nd, nf\}$

ARX[nα,nβ]

$$\begin{cases} A_\alpha(q^{-1}) y[n] = B_\beta(q^{-1}) u[n] + v[n] \\ E\{v[n]v[m]\} = \lambda_v^2 \delta_0[n-m] \end{cases} \quad \forall\, n, m \in \mathbb{N}$$

$$\begin{bmatrix} \psi^T[n] \overset{def}{=} \left[-y[n-1] \cdots -y[n-n\alpha] \mid u[n-1] \cdots u[n-n\beta] \right] \\ \theta_{\alpha\beta}^T \overset{def}{=} \left[\alpha_1 \cdots \alpha_{n\alpha} \mid \beta_1 \cdots \beta_{n\beta} \right] \end{bmatrix}$$

$$\mathcal{D} = \{(u[n], y[n])\}_{n \in \overline{1,N}} \quad \forall\, n \in \mathbb{N}^*$$

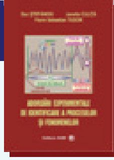
date măsurate

MCMMP

$$\hat{\theta}_{\alpha\beta} = \left(\frac{1}{N} \sum_{n=1}^{N} \psi[n]\psi^T[n] \right)^{-1} \left(\frac{1}{N} \sum_{n=1}^{N} \psi[n] y[n] \right)$$

$$\varepsilon\left[n, \hat{\theta}_{\alpha\beta}\right] \overset{def}{=} y[n] - \psi^T[n]\hat{\theta}_{\alpha\beta} \quad \forall\, n \in \overline{1,N}$$

zgomotul estimat



❺ Identificare parametrică prin MMEP

☞ Contextul de lucru

MCMMPE (continuare)

② Determinarea parametrilor originali ai modelului cu ajutorul vectorului regresorilor avînd componenta nemăsurabilă estimată în etapa precedentă.

$$\varepsilon\left[n,\hat{\boldsymbol{\theta}}_{\alpha\beta}\right] \overset{def}{=} y[n]-\boldsymbol{\psi}^T[n]\hat{\boldsymbol{\theta}}_{\alpha\beta} \qquad \forall n\in\overline{1,N}$$

zgomotul estimat

Vectorul estimat al regresorilor

ARMAX[na,nb,nc]

$$\boldsymbol{\varphi}_{\alpha\beta}[n] \overset{def}{=} \Big[-y[n-1]\ \cdots -y[n-na]\ \Big|\ u[n-1]\ \cdots\ u[n-nb]\ \Big|\ \cdots$$
$$\cdots\ \varepsilon[n-1,\hat{\boldsymbol{\theta}}_{\alpha\beta}]\cdots\varepsilon[n-nc,\hat{\boldsymbol{\theta}}_{\alpha\beta}]\Big]$$

BJ[nb,nc,nd,nf]

$$\boldsymbol{\varphi}_{\alpha\beta}[n] \overset{def}{=} \Big[-y[n-1]\ \cdots -y[n-nd+nf]\ \Big|\ u[n-1]\ \cdots\ u[n-nb+nd]\ \Big|\ \cdots$$
$$\cdots\ \varepsilon[n-1,\hat{\boldsymbol{\theta}}_{\alpha\beta}]\cdots\varepsilon[n-nc+nf,\hat{\boldsymbol{\theta}}_{\alpha\beta}]\Big]$$

$$\mathcal{D}=\{(u[n],y[n])\}_{n\in\overline{1,N}}$$

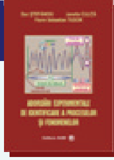
date măsurate

MCMMP

$$\hat{\boldsymbol{\theta}}_N \overset{def}{=} \left(\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{\varphi}_{\alpha\beta}[n]\boldsymbol{\varphi}_{\alpha\beta}^T[n]\right)^{-1}\left(\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{\varphi}_{\alpha\beta}[n]y[n]\right)$$

$$\hat{\lambda}_N^2 \overset{def}{=} \frac{1}{N}\sum_{n=1}^{N}\left(y[n]-\boldsymbol{\varphi}_{\alpha\beta}^T[n]\hat{\boldsymbol{\theta}}_N\right)^2$$

În cazul modelului **BJ**, coeficienţii celor 4 polinoame se determină cu ajutorul coeficienţilor vectorului parametrilor estimaţi, **prin identificarea rădăcinilor comune**.



❺ Identificare parametrică prin MMEP

☞ Contextul de lucru

MMEP (algoritmul general)

Date de intrare

- $\mathcal{D}_N = \{(u[n], y[n])\}_{n\in\overline{1,N}}$ (date intrare-ieşire măsurate)
- $\varepsilon = f[n,\boldsymbol\theta]$ (expresia erorii de predicție, cel puțin derivabile)
- $\eta > 0$ (prag de precizie)

$$\varepsilon[n,\boldsymbol\theta] = y[n] - \boldsymbol\varphi^T[n]\boldsymbol\theta$$

Inițializare

- $\hat{\boldsymbol\theta}_{N,0} \in \mathbb{R}^{n\theta}$ ➔ evaluat cu ajutorul MCMMPE
- $\alpha_0 \in \mathbb{R}^*$ ➔ ales fie arbitrar, fie cu ajutorul unui algoritm specific
- $k = 0$ ➔ indicele iterativ inițial

Buclă iterativă

☝ Se utilizează iterația specifică din cadrul MGN.

① Se reactualizează aproximația optimului

$$\hat{\boldsymbol\theta}_{N,k+1} = \hat{\boldsymbol\theta}_{N,k} - \alpha_k \underbrace{\left[\sum_{n=1}^{N}\nabla\varepsilon[n,\hat{\boldsymbol\theta}_{N,k}]\left[\nabla\varepsilon[n,\hat{\boldsymbol\theta}_{N,k}]\right]^T\right]^{-1}}_{\mathbf{R}_{N,k}^{-1}} \underbrace{\left[\sum_{n=1}^{N}\varepsilon[n,\hat{\boldsymbol\theta}_{N,k}]\nabla\varepsilon[n,\hat{\boldsymbol\theta}_{N,k}]\right]}_{\mathbf{r}_{N,k}}$$

② Se reactualizează pasul variabil

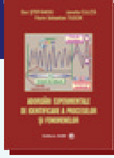
$$\alpha_{k+1} = \alpha_k + \frac{\mathbf{r}_{N,k+1}^T\,\mathbf{R}_{N,k}^{-1}\,\mathbf{r}_{N,k}}{\mathbf{r}_{N,k}^T\,\mathbf{R}_{N,k}^{-1}\,\mathbf{R}_{N,k+1}\,\mathbf{R}_{N,k}^{-1}\,\mathbf{r}_{N,k}}$$

$$|\alpha_k|\cdot\left\|\mathbf{R}_{N,k}^{-1}\,\mathbf{r}_{N,k}\right\| < \eta$$

NU: $k \leftarrow k+1$ → ①

DA: **Date de ieşire**

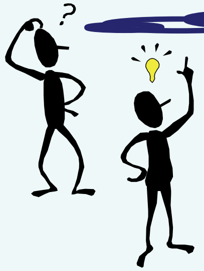
- $\hat{\boldsymbol\theta}_{N,k+1}$ ➔ aproximația de precizie η
- $k+1$ ➔ numărul de iterații efectuate



❺ Identificare parametrică prin MMEP

☞ Contextul de lucru

MMEP (continuare)



Cum se pot determina eroarea de predicție și gradientul acesteia?

Ar putea fi utilizată ecuația modelului de identificare, cu parametrii curent estimați.

Va fi analizat cazul modelului **ARMAX**. Cazul modelului **BJ** se analizează similar.

$$\hat{A}_{N,k}(q^{-1})y[n]=\hat{B}_{N,k}(q^{-1})u[n]+\hat{C}_{N,k}(q^{-1})\varepsilon\left[n,\hat{\boldsymbol{\theta}}_{N,k}\right]$$

$\forall n\in\mathbb{N}^*$

modelul de identificare estimat la pasul curent

☝ Ecuație recurentă pentru determinarea erorii de predicție.

$$\begin{aligned}\varepsilon[n,\hat{\boldsymbol{\theta}}_{N,k}]={}&y[n]+\hat{a}^{N}_{1,k}\,y[n-1]+\cdots+\hat{a}^{N}_{na,k}\,y[n-na]-\\&-\hat{b}^{N}_{1,k}\,u[n-1]-\cdots-\hat{b}^{N}_{nb,k}\,u[n-nb]-\\&-\hat{c}^{N}_{1,k}\,\varepsilon\left[n-1,\hat{\boldsymbol{\theta}}_{N,k}\right]-\cdots-\hat{c}^{N}_{nc,k}\,\varepsilon\left[n-nc,\hat{\boldsymbol{\theta}}_{N,k}\right]\end{aligned}$$

$\forall n\in\mathbb{N}^*$

$$\boldsymbol{\varphi}[n,\boldsymbol{\theta}]\overset{def}{=}\begin{bmatrix}\varphi_y[n]\\ \hline \varphi_u[n]\\ \hline \varphi_\varepsilon[n,\boldsymbol{\theta}]\end{bmatrix}=\begin{bmatrix}[-y[n-i]]_{i=\overline{1,na}}\\ \hline [u[n-j]]_{j=\overline{1,nb}}\\ \hline [\varepsilon[n-1,\boldsymbol{\theta}]]_{l=\overline{1,nc}}\end{bmatrix}$$

$$\boldsymbol{\theta}^T\overset{def}{=}\begin{bmatrix}\boldsymbol{\theta}_a^T & \boldsymbol{\theta}_b^T & \boldsymbol{\theta}_c^T\end{bmatrix}$$

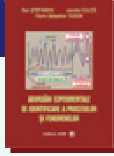
$$\begin{cases}\nabla_{\boldsymbol{\theta}_a}\varepsilon\left[n,\hat{\boldsymbol{\theta}}_N[k]\right]=-\varphi_y[n]-\hat{c}_{1,k}\nabla_{\boldsymbol{\theta}_a}\varepsilon\left[n-1,\hat{\boldsymbol{\theta}}_N[k]\right]-\cdots-\hat{c}_{nc,k}\nabla_{\boldsymbol{\theta}_a}\varepsilon\left[n-nc,\hat{\boldsymbol{\theta}}_N[k]\right]\\ \nabla_{\boldsymbol{\theta}_b}\varepsilon\left[n,\hat{\boldsymbol{\theta}}_N[k]\right]=-\varphi_u[n]-\hat{c}_{1,k}\nabla_{\boldsymbol{\theta}_b}\varepsilon\left[n-1,\hat{\boldsymbol{\theta}}_N[k]\right]-\cdots-\hat{c}_{nc,k}\nabla_{\boldsymbol{\theta}_b}\varepsilon\left[n-nc,\hat{\boldsymbol{\theta}}_N[k]\right]\\ \nabla_{\boldsymbol{\theta}_c}\varepsilon\left[n,\hat{\boldsymbol{\theta}}_N[k]\right]=-\varphi_\varepsilon\left[n,\hat{\boldsymbol{\theta}}_N[k]\right]\end{cases}$$

$\forall n\in\mathbb{N}^*$

Inițializare cauzală

$$\nabla\varepsilon[n,\hat{\boldsymbol{\theta}}_{N,k}]=\varepsilon[n,\hat{\boldsymbol{\theta}}_{N,k}]=0$$

$$u[n]=y[n]=0 \qquad \forall n\leq 0$$



❺ Identificare parametrică prin MMEP

☞ Contextul de lucru

Procesele generatoare de date

ARMAX[2,2,2]

$$\left(1-1.5q^{-1}+0.7q^{-2}\right)y[n]=\left(q^{-1}+0.5q^{-2}\right)u[n]+\left(1-q^{-1}+0.2q^{-2}\right)e[n]$$

$$\forall n\in\mathbb{N}^{*}$$

BJ[2,2,2,2]

$$y[n]=\frac{q^{-1}+0.5q^{-2}}{1-1.5q^{-1}+0.7q^{-2}}u[n]+\frac{1-q^{-1}+0.2q^{-2}}{1+1.5q^{-1}+0.7q^{-2}}e[n]$$

$$\forall n\in\mathbb{N}^{*}$$

e ➔ SPAB Gaussian sau bipolar de medie nulă şi dispersie unitară

Date generate

$$\mathcal{D}_{id}=\{u[n]\}_{n=\overline{1,N}}\cup\{y[n]\}_{n=\overline{1,N}}$$ ➔ date măsurate pentru identificare

$$N=250$$

$$\mathcal{D}_{va}=\{u[n]\}_{n=\overline{1,N}}\cup\{y[n]\}_{n=\overline{1,N}}$$ ➔ date măsurate pentru validare

$$N=250$$

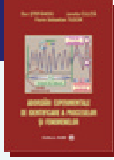
Indici structurali maximali $Na=Nb=Nc=5$

☝ Modelul BJ se poate identifica folosind un model ARMAX.

Teste structurale principale ➔ Aceleaşi din cadrul Lucrării de laborator #4, dar adaptate la indicii structurali ai modelelor ARMAX şi BJ.

Obiectiv

- Compararea performanţelor MCMMPE & MMEP în cazul modelelor ARMAX şi BJ.



❺ Identificare parametrică prin MMEP

☞ Probleme de simulare

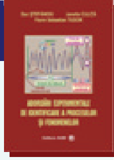
Contextul de lucru

Rutine preliminare

```
[D,V,P] = gen_data(DP,N,sigma,lambda,bin) ;
```

(generează date)

DP	este obiectul de tip **IDMODEL** corespunzător modelului de proces furnizor de date; obiectul poate fi construit de exemplu cu ajutorul funcției **idpoly**; implicit, acest model este identic cu cel de tip ARMAX;
N	este dimensiunea orizontului de măsură (implicit: **N=250**);
sigma	este deviația standard a intrării SPAB (implicit: **sigma=1**);
lambda	este deviația standard a zgomotului alb Gaussian (implicit: **lambda=1**);
bin	este un parametru care arată tipul de intrări dorit: **bin=0** (intrare SPAB Gaussiană); **bin~=0** (implicit, intrare SPAB Gaussiană bipolară);
D	este obiectul de tip **IDDATA** corespunzător datelor generate (intrarea se regăseşte în **D.u**, iar ieşirea în **D.y**);
V	este obiectul de tip **IDDATA** corespunzător zgomotelor generate (zgomotul alb se regăseşte în **V.u**, iar zgomotul colorat (MA-filtrat) – în **V.y**).
P	este obiectul de tip **IDMODEL** corespunzător modelului de proces furnizor de date.



❺ Identificare parametrică prin MMEP

☞ Probleme de simulare

Contextul de lucru

Rutine preliminare

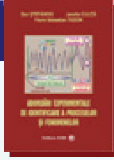
Rutină de bibliotecă MATLAB-IS

`ARMAX`

- Apel: `Mid = armax(D,si) ;`
- Estimează parametrii unui model ARMAX folosind MMEP. Modelul identificat rezultat, `Mid`, este returnat ca obiect `IDMODEL`. Estimarea se efectuează pe baza datelor `D` (obiect `IDDATA`) şi a informaţiei de structură `si = [na nb nc nk]`, unde `na`, `nb` şi `nc` sunt indicii structurali ai modelului, iar `nk` este întîrzierea instrinsecă. Cu ajutorul acestei rutine se pot identifica atît modele AR cît şi modele ARMA unidimensionale (însă nu şi multi-dimensionale). Apelul rutinei este uşor diferit în acest caz:
  - pentru modele AR: `Mid = armax(D.y,na) ;`
  - pentru modele ARMA: `Mid = armax(D.y,[na nc]) ;`

  Observaţi că datele de identificare sunt specificate acum doar sub forma unei *serii de timp* (`D.y`). Pentru identificarea modelelor AR, rutina apelează intern o funcţie specializată numită `ar`, care este diponibilă şi utilizatorului (cu apel similar lui `armax`).

  O altă modalitate de a identifica modele AR şi ARMA este de a folosi obiectul `D` împreună cu o informaţie de structură de forma: `si = [na 0 0 0]` (AR) sau `si = [na 0 nc 0]` (ARMA). Rutina nu funcţionează însă decît dacă `na>1` (nu şi pentru `na=1`). De aceea, se recomandă utilizarea rutinei cu argument serie de timp, pentru aceste modele.



❺ Identificare parametrică prin MMEP

☞ Probleme de simulare

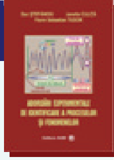
Contextul de lucru

Rutine preliminare

Rutină de bibliotecă MATLAB-IS

BJ

- Apel: `Mid = bj(D,si) ;`
- Estimează parametrii unui model BJ folosind MMEP. Modelul identificat rezultat, **`Mid`**, este returnat ca obiect **`IDMODEL`**. Estimarea se efectuează pe baza datelor **`D`** (obiect **`IDDATA`**) şi a informației de structură **`si = [na nb nc nd nf nk]`**, unde **`na`**, **`nb`**, **`nc`**, **`nd`** şi **`nf`** sunt indicii structurali ai modelului, iar **`nk`** este întîrzierea instrinsecă. În principiu, algoritmul implementat în cadrul acestei rutine este similar cu cel al rutinei **`armax`**, cu adaptările de rigoare impuse de utilizarea modelului BJ.



❺ Identificare parametrică prin MMEP

☞ Probleme de simulare

Problema 6.1 (MMEP pentru modelul ARMAX)

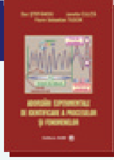
A fost proiectat mini-simulatorul **`ISLAB_6A`** care evaluează estimația (parsimonioasă a) modelului ARMAX asociat procesului, folosind MMEP. Pentru aceasta, s-au parcurs următorii paşi:

a. Se generează 2 seturi de date: unul pentru identificare şi altul pentru validare, folosind rutina **`gen_data`**.

b. Pentru fiecare model identificat cu ajutorul MMEP (funcția **`armax`**), model obținut variind indicii na, nb şi nc, se afişează 3 ferestre grafice: una pentru analiza modelului folosind datele de identificare şi de validare şi alte două pentru reprezentările poli-zeroruri (filtru sistem şi filtru zgomot) cu discuri de încredere corespunzătoare unei raze de 3 ori mai mari decît deviațiile standard aferente. După fiecare fereastră s-a inserat o pauză de aşteptare pentru a permite utilizatorului să analizeze informațiile afişate. Fiecare sub-fereastră a primei ferestre include 3 grafice aranjate pe verticală:

- ➢ ieşirile măsurate şi simulate cu ajutorul modelului, grafic pe care se indică şi valoarea funcției de potrivire, $\mathcal{E}_N$;
- ➢ eroarea de predicție (reziduurile modelului), grafic pe care se indică şi dispersia estimată a zgomotului, λ_N^2 ;
- ➢ secvența de auto-covarianță a erorii de predicție, grafic pe care se indică şi indexul de validare.

Modelele obținute sunt memorate în vederea selectării unuia dintre ele, în urma aplicării testelor de determinare a indicilor structurali optimi şi de validare.



❺ Identificare parametrică prin MMEP

☞ Probleme de simulare

Problema 6.1 (final)

c. Se afişează indicii structurali optimi selectaţi folosind:
   - Testul F aplicat dispersiei estimate a zgomotului (adică erorii de predicţie);
   - Testul F aplicat funcţiei de potrivire pentru datele de identificare;
   - Testul F aplicat funcţiei de potrivire pentru datele de validare;
   - criteriului GAIC în versiunea Rissanen.

d. Se solicită utilizatorului să aleagă indicii structurali pe care îi consideră optimi.

e. Pentru modelul ales, se afişează cele 3 ferestre grafice de la b. Modelul este returnat de către mini-simulator, în vederea unei utilizări ulterioare. Se recomandă returnarea şi a seturilor de date de identificare şi validare.

Pentru testarea mini-simulatorului **`ISLAB_6A`**, se vor iniţia cîteva rulări.

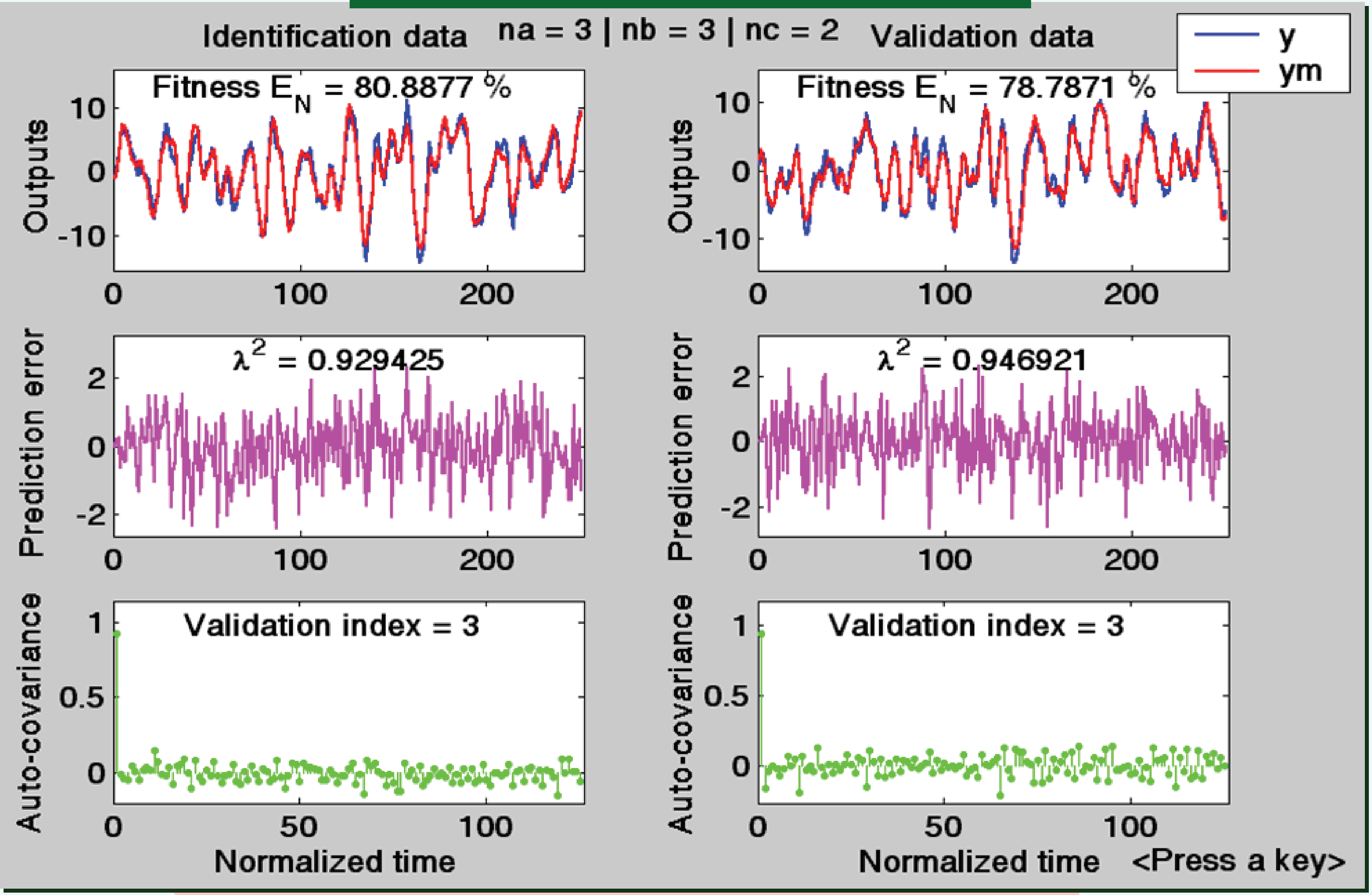
Sunt indicii structurali adevăraţi indicaţi de către majoritatea criteriilor utilizate sau ei diferă de la o rulare la alta? Justificaţi răspunsul.

Program existent

`ISLAB_6A`

❺ Identificare parametrică prin MMEP

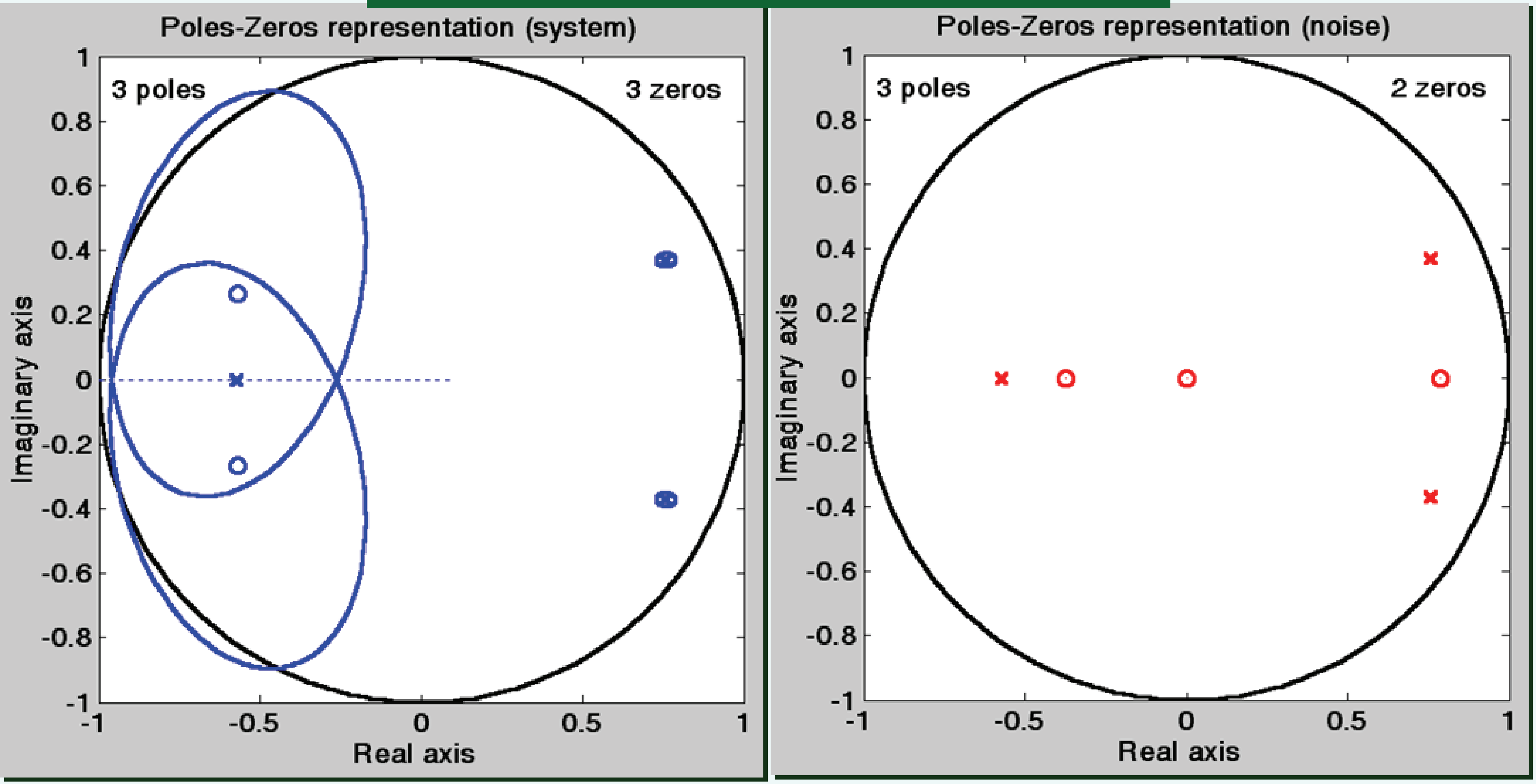
Ce afişează mini-simulatorul **ISLAB_6A**



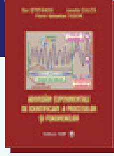
Performanţele modelului ARMAX identificat cu MMEP

❺ Identificare parametrică prin MMEP

Ce afişează mini-simulatorul **ISLAB_6A** (final)



Reprezentarea poli-zerouri a modelului ARMAX identificat cu MMEP



❺ Identificare parametrică prin MMEP

☞ Probleme de simulare

Problema 6.2 (MMEP pentru modelul BJ)

Problema anterioară, **6.1**, se va relua pentru modelul BJ. Mini-simulatorul rezultat va fi denumit **`ISLAB_6B`**. Testați mini-simulatorul și comentați rezultatele de estimare obținute.

Program ce trebuie proiectat **ISLAB_6B**

Problema 6.3 (MCMMPE pentru modelele ARMAX şi BJ)

Biblioteca MATLAB dedicată domeniului IS nu dispune de funcții explicite care implementează MCMMPE. Să se proiecteze două astfel de funcții: **`armax_e`** pentru identificarea modelelor ARMAX și **`bj_e`** pentru identificarea modelelor BJ. Apelul tipic al lor ar trebui să fie similar altor funcții cu obiectiv asemănător (estimarea parametrilor unui model cu structură dată; vezi de exemplu funcțiile **`armax`** și **`bj`**):

```
Mid = armax_e(D,si) ;
Mid = bj_e(D,si) ;
```

Informația de structură are forma: **`si = [na nb nc nk]`** pentru modelul ARMAX și **`si = [na nb nc nd nf nk]`** pentru modelul BJ. Încercați să folosiți funcția **`armax_e`** în cadrul funcției **`bj_e`**. Proiectați apoi mini-simulatorul **`ISLAB_6C`** similar celor din problemele precedente. Comparați performanțele acestui mini-simulator cu ale mini-simulatoarelor precedente. Este MCMMPE de preferat MMEP? Justificați răspunsul într-o manieră riguroasă.

Program ce trebuie proiectat **ISLAB_6C**

Rutine ce trebuie proiectate **ARMAX_E** **BJ_E**