

ARMAX MCMMPPE

M=armax\_e(D,si)

1. Se definesc valorile si=[na nb nc] asociate modelului ARMAX[na,nb,nc] de estimat
2. Pe baza setului de date D se identifica un model ARX[n\_alpha,n\_beta] ce va fi utilizat pentru estimarea zgomotului. Zgomotul estimat va fi inlocuit in forma de regresie liniara ARMAX, deoarece valorile adevarate nu se cunosc.  
Ce valori se aleg pentru n\_alpha si n\_beta? n\_alpha=na si n\_beta=nb? Nu.  
2.1 Se va imparti ecuatia ARMAX prin C(q^-1) pentru ca sa se obtina un model ARX

$$\frac{A(q^{-1})}{C(q^{-1})}y[n] = \frac{B(q^{-1})}{C(q^{-1})}u[n] + e[n]$$

ARX

$$\frac{A(q^{-1})}{C(q^{-1})} \cong A_{\alpha}(q^{-1}) = 1 + \alpha_1 q^{-1} + \dots + \alpha_{nc} q^{-nc}$$

$$\frac{B(q^{-1})}{C(q^{-1})} \cong B_{\beta}(q^{-1}) = 1 + \beta_1 q^{-1} + \dots + \beta_{nb} q^{-nb}$$

ARMAX

$$A(q^{-1})y[n] = B(q^{-1})u[n] + C(q^{-1})e[n]$$

$A_{\alpha}(q^{-1})$

$B_{\beta}(q^{-1})$

au nr infinit de coeficienti in urma impartirii.  
se va face trunchiere la ordinele n\_alpha si n\_beta  
alegerea n\_alpha si n\_beta ARX se va raporta la  
na,nb,nc ARMAX

ARMAX are coef. c1,...cnc in plus fata de  
ARX, deci vom alege pentru n\_alpha, n\_beta  
valori mai mari decat na,nb,nc.  
Astfel, n\_alpha si n\_beta vor fi alese a.i sa fie indeplinita conditia:

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

- 2.2. Avand setul de date D si n\_alpha, n\_beta, se va identifica un model ARX[n\_alpha,n\_beta]  
pentru proces utilizand MCMMP.

Laboratorul 3, se poate aplica functia MATLAB arx. Se obtin parametrii ARX si se estimeaza componentele lui e.  
Pentru estimarea lui e se poate utiliza functia Matlab pe.

3. Forma de regresie liniara ARMAX:

$$y[m] = \Phi_{\alpha\beta}^T[n] \Theta + e[m]$$

$$\Phi_{\alpha\beta}^T[n] \stackrel{def}{=} \begin{bmatrix} -y[n-1] - y[n-2] \dots - y[n-na] & | & u[n-1] u[n-2] \dots u[n-nb] & | & \dots \\ & & & & \dots e[n-1] e[n-2] \dots e[n-nc] \end{bmatrix}$$

3 componente,  
ca si vectorul parametrilor

$$\Theta^T \stackrel{def}{=} \begin{bmatrix} a_1 & a_2 \dots a_{na} & | & b_1 & b_2 \dots b_{nb} & | & c_1 & c_2 \dots c_{nc} \end{bmatrix}$$

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

Se inlocuiesc valorile lui e estimate, deoarece nu erau cunoscute=>  $\Phi_{\alpha\beta}^T[n]$  este acum cunoscut .

Se va utiliza MCMMP pt aflarea Theta\_N.  
Se implementeaza formula de calcul  
Theta\_N inlocuind  $\Phi_{\alpha\beta}^T[n]$  si y[n]:

date măsurate

MCMMP

$$\hat{\Theta}_N \stackrel{def}{=} \left( \frac{1}{N} \sum_{n=1}^N \Phi_{\alpha\beta}[n] \Phi_{\alpha\beta}^T[n] \right)^{-1} \left( \frac{1}{N} \sum_{n=1}^N \Phi_{\alpha\beta}[n] y[n] \right)$$
$$\hat{\lambda}_N^2 \stackrel{def}{=} \frac{1}{N} \sum_{n=1}^N \left( y[n] - \Phi_{\alpha\beta}^T[n] \hat{\Theta}_N \right)^2$$

$$e[m] = y[m] - \Phi^T[m] \hat{\Theta}_N$$

se inlocuiesc valorile

BJ MCMMPPE

BJ[nb,nc,nd,nf]

(Box Jenkins)

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

Mid=bj\_e(D,si)

1. Se definesc valorile si=[nb,nc,nd,nf] asociate modelului BJ[nb,nc,nd,nf] de estimat.
2. Se va aduce ecuatia BJ la acelasi numitor => o ecuatie ARMAX.

$$y[n] = \frac{B(q^{-1})}{F(q^{-1})}u[n] + \frac{C(q^{-1})}{D(q^{-1})}e[n]$$

$$F(q^{-1}) D(q^{-1}) y[n] = B(q^{-1}) D(q^{-1}) u[n] + C(q^{-1}) F(q^{-1}) e[n]$$

$$\frac{A(q^{-1})}{F,D} y[n] = \frac{B(q^{-1})}{B,D} u[n] + \frac{C(q^{-1})}{C,F} e[n]$$

Am rescris BJ ca ARMAX

Gradul A\_(F,D) este egal cu suma gradelor polinoamelor F,D = nf+nd

Gradul B\_(B,D) este egal cu suma gradelor polinoamelor B,D =nb+nd

Gradul C\_(C,F) este egal cu suma gradelor polinoamelor C,F =nc+nf

3. Pe baza setului de date D se identifica un model ARX[n\_alpha,n\_beta] ce va fi utilizat pentru estimarea zgomotului. Zgomotul estimat va fi inlocuit in forma de regresie liniara BJ-ARMAX, deoarece valorile adevarate nu se cunosc.

Astfel, n\_alpha si n\_beta vor fi alese a.i sa fie indeplinita conditia:

$$\min\{n_{\alpha}, n_{\beta}\} \gg \max\{n_f + n_d, n_b + n_d, n_c + n_f\}$$

4. Idem pasul 3 ARMAX MCMMPPE.

Pasii 3 si 4 se pot inlocui prin M=armax\_e(D,si) cu si=[nf+nd,nb+nd,nc+nf]. Obiectul IDMODEL M va contine parametrii estimati  $\hat{\Theta}_N^T = \begin{bmatrix} a_1^{F,D} & a_2^{F,D} \dots a_{na}^{F,D} & | & b_1^{B,D} & b_2^{B,D} \dots b_{nb}^{B,D} & | & c_1^{C,F} & c_2^{C,F} \dots c_{nc}^{C,F} \end{bmatrix}$

5. Am obtinut  $\hat{\Theta}_N^T = \begin{bmatrix} a_1^{F,D} & a_2^{F,D} \dots a_{na}^{F,D} & | & b_1^{B,D} & b_2^{B,D} \dots b_{nb}^{B,D} & | & c_1^{C,F} & c_2^{C,F} \dots c_{nc}^{C,F} \end{bmatrix}$  corespunzator modelului  $\frac{A(q^{-1})}{F,D} y[n] = \frac{B(q^{-1})}{B,D} u[n] + \frac{C(q^{-1})}{C,F} e[n]$  , dar

se doresc parametrii lui BJ: b1,..b\_nb c1...c\_nc d1...d\_nd f1..f\_nf .

polinomul B coef. b1,..b\_nb

BJ[nb,nc,nd,nf]

polinomul C coef. c1...c\_nc

$$y[n] = \frac{B(q^{-1})}{F(q^{-1})}u[n] + \frac{C(q^{-1})}{D(q^{-1})}e[n]$$

polinomul F coef. f1..f\_nf .

polinomul D coef. d1...d\_nd

Daca se identifica radacinile comune ale A\_(F,D) si B\_(B,D) => radacinile polinomului D => polinomul D al modelului BJ

Daca se identifica radacinile comune ale A\_(F,D) si C\_(C,F) => radacinile polinomului F => polinomul F al modelului BJ

Dupa extragerea radacinilor polinomului F din polinomul C\_(C,F) => radacinile polinomului C => polinomul C al modelului BJ

Dupa extragerea radacinilor polinomului D din polinomul B\_(B,D) => radacinile polinomului B => polinomul B al modelului BJ

Deci, am obtinut parametrii modelului BJ.

Exista eroare de estimare, posibil sa nu se gaseasca radacini comune. Se poate considera o conditie:  
diferenta in modul dintre cele doua radacini sa fie mai mica decat un epsilon de valoare mica.

Matlab  
roots: Find polynomial roots.  
poly: Convert roots to polynomial.



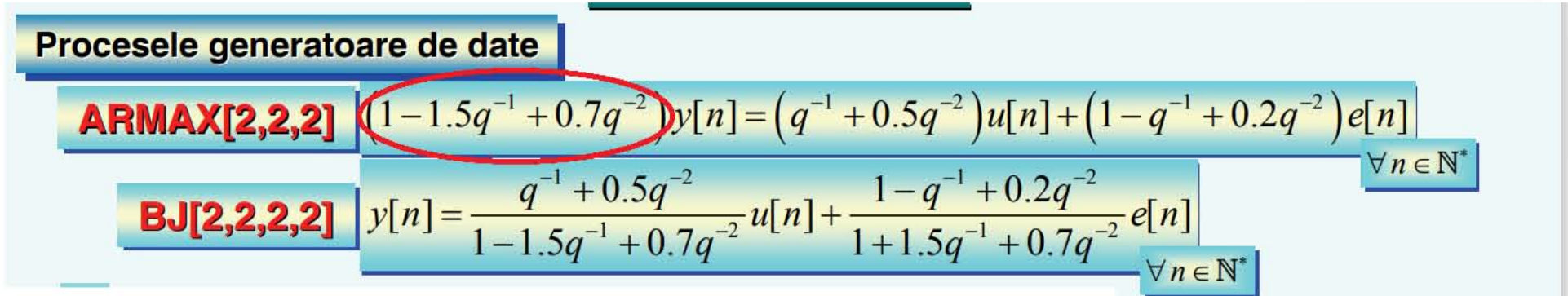
Islab\_6b.m

Islab\_6b.m=Islab\_6a.m, cu modificari:

- Islab\_6a.m si Islab\_6b.m apeleaza gen\_data.m care genereaza datele de identificare.

In rutina gen\_data.m se modifica procesul de tip ARMAX cu un proces de tip BJ. Procesele sunt pe slide-ul 78 in 05\_IS\_Lab\_pp#L.72-L.86.pdf.

```
if (nargin < 1)
    DP = idpoly([1 -1.5 0.7],[0 1 0.5],[1 -1 0.2]);
end
if (isempty(DP))
    DP = idpoly([1 -1.5 0.7],[0 1 0.5],[1 -1 0.2]);
end
if (~isa(DP,'IDMODEL'))
    DP = idpoly([1 -1.5 0.7],[0 1 0.5],[1 -1 0.2]);
end
```



- na,nb,nc pt. ARMAX[na,nb,nc] se inlocuiesc cu nb,nc,nd pt. BJ[nb,nc,nd,nf]

(Find&Replace) si se adauga nf suplimentar.

```
for na=1:Na
    for nb=1:Nb
        for nc=1:Nc
            %Pentru calculul indicilor structurali optimi, ca in Lab4, se variaza gradele polinoamelor ([na,nb,nc] cazul
            %ARMAX, [nb,nc,nd,nf] cazul BJ) si se calculeaza valoarea criteriului GAIC pentru fiecare combinatie de indici
            %structurali. Indicii structurali optimi corespund valorii minime pt criteriul GAIC. Se va cauta minimul criteriului
            %GAIC (valorile criteriului sunt salvate intr-o matrice pentru fiecare pereche de indici structurali) => indicii
            %structurali optimi corespunzatori minimului ([na_opt,nb_opt,nc_opt] cazul ARMAX,
            %[nb_opt,nc_opt,nd_opt,nf_opt] cazul BJ) .

            se transforma in:

            for nb=1:Nb
                for nc=1:Nc
                    for nd=1:Nd
                        for nf=1:Nf
```

- Apelul armax linia 196 trebuie inlocuit cu apelul bj.

armax.m implementeaza MMEP (Metoda minimizarii erorii de predictie) pentru modele ARMAX.

(arx.m implementeaza MCMMP pt modele ARX)

bj.m implementeaza MMEP (Metoda minimizarii erorii de predictie) pentru modele BJ.

- Lambda = 1000\*lambda\*ones(Na,Nb,Nc); si Lambda(na,nb,nc) = Mid.NoiseVariance;

se transforma in

Lambda = 1000\*lambda\*ones(Nb,Nc,Nd,Nf); si Lambda(nb,nc,nd,nf) = Mid.NoiseVariance;

% Mid.NoiseVariance contine valorile dispersiei zgomotului estimat. Dispersia se utilizeaza pentru calculul criteriului GAIC pentru fiecare pereche de indici structurali .

- GAIC\_R3.m se inlocuieste cu GAIC\_R4.m. GAIC\_R4.m trebuie implementat, indicatiile in fisierul GAICs.pdf (Find&Replace)

ARX:

② Cazul 2D:  $GAIC-R_N^2[na,nb] = \ln(\hat{\lambda}_N^2[na,nb]) + \frac{na+nb}{N} \ln(N)$

$(nb = na+nb)$

\* Implementare:

$GAIC-R_N^2[na,nb] = \ln(\hat{\lambda}_N^2[na,nb]) + \frac{na+nb}{N} \ln(N)$

$+ \frac{\ln(N)}{N} \left( \begin{matrix} 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ na-1 & na-1 & na-1 \end{matrix} + \begin{matrix} 0 & 1 & \dots & nb-1 \\ 0 & 1 & \dots & nb-1 \\ 0 & 1 & \dots & nb-1 \end{matrix} \right) na$

```
GAIC_R2.m - Notepad
File Edit Format View Help
end;
N = log(N)/N;
% Evaluating the GAIC_R criterion
% ~~~~~
[na,nb] = size(Lambda);
GAICR = log(Lambda) + ...
    N*(1:(na))*ones(1,nb) + ones(na,1)*(1:nb)-2);
if ((na<2) & (nb<2))
    war_err(W1);
elseif (na<2)
    [Lambda,nb] = min(GAICR);
elseif (nb<2)
    [Lambda,na] = min(GAICR);
else
    [Lambda,na] = min(GAICR); % For matrices, min(X) is a row vector containing the minimum element from each column. [Y,I] = min(X) returns the indices of the minimum values in vector I.
    [Lambda,nb] = min(Lambda);
    na = na(nb);
end;
% 0-scalar
% 1-vector
% 2-matrix 2D
% 3-matrix 3D
% 4-matrix 4D
```

```
function [na,nb,GAICR] = GAIC_R2(Lambda,N)
% Inputs: 2 Lambda # estimated noise variance matrix
%         N # size of measured data set
% Outputs: na 0 # optimum row order
%          nb 0 # optimum column order
%          GAICR 2 # values of GAIC_R criterion
```

Example:

>> X = [2 8 4; 7 3 9];

>> [Y,I]=min(X)

Y = 2 3 4 % minimul pe fiecare coloana din X

I = 1 2 1 % linia pe care se afla minimul calculat pentru fiecare coloana

ARMAX:  
Islab\_6a.m

③ Cazul 3D:  $GAIC-R_N^3[na,nb,nc] = \ln(\hat{\lambda}_N^3[na,nb,nc]) + \frac{na+nb+nc}{N} \ln(N)$

$(nb = na+nb+nc)$

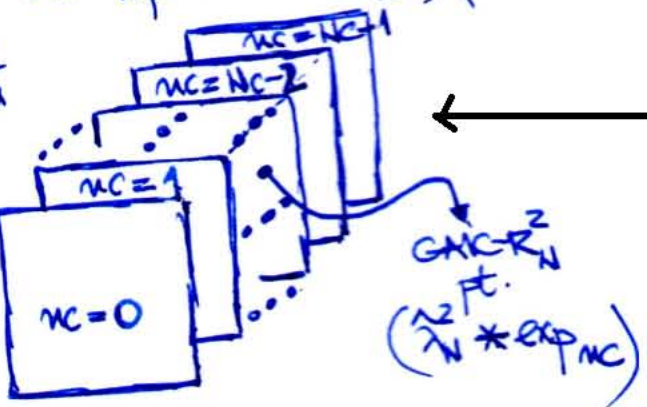
$= \ln(\hat{\lambda}_N^3[na,nb,nc]) + nc \frac{\ln(N)}{N} + \frac{na+nb}{N} \ln(N) =$

$= \ln(\hat{\lambda}_N^3[na,nb,nc] \cdot e^{nc \cdot \ln(N)/N}) + \frac{na+nb}{N} \ln(N)$

$GAIC-R_N^3$  cu  $\hat{\lambda}_N^3$  modificat pt. fiecare nc

\* Implementare: - se construiesc un bloc 3D format din straturi 2D de tip GAIC-R<sup>2</sup>, folosind:

- un ciclu for
- functia "CAT"



```
GAIC_R3.m - Notepad
File Edit Format View Help
return;
end;
% Evaluating the GAIC_R criterion
% ~~~~~
[Na,Nb,Nc] = size(Lambda);
GAICR = [];
Gmin = [];
for nc=1:Nc
    [Na,Nb,G] = GAIC_R2(Lambda(:, :, nc)*exp((nc-1)*log(N)/N), N);
    na = [na Na];
    nb = [nb Nb];
    GAICR = cat(3, GAICR, G);
    Gmin = [Gmin G(Na,Nb)]; % G(Na,Nb) minimul din matricea 2D; Na,Nb sunt optimali
end;
[Gmin,nc] = min(Gmin); % nc indica in care matrice 2D se afla minimul din toata matricea 3D.
na = na(nc);
nb = nb(nc);
% Gmin vector (membrul drept "=") contine minimul din fiecare matrice 2D, aflat pe baza apelului GAIC_R2.m
% Gmin scalar (membrul stang "=") contine minimul din toata matricea 3D.
% END
```

```
function [na,nb,nc,GAICR] = GAIC_R3(Lambda,N)
% Inputs: 3 Lambda # estimated noise variance matrix
%         N # size of measured data set
% Outputs: na 0 # optimum row order
%          nb 0 # optimum column order
%          nc 0 # optimum layer order
%          GAICR 3 # values of GAIC_R criterion
```

Examples

Given

A =  $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$  B =  $\begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$

concatenating along different dimensions produces

$C = \text{cat}(1,A,B)$

$C = \text{cat}(2,A,B)$

$C = \text{cat}(3,A,B)$

BJ:

Islab\_6b.m

④ Cazul 4D:  $GAIC-R_N^4[nb,nc,nd,nf] = \ln(\hat{\lambda}_N^4[nb,nc,nd,nf]) + \frac{nb+nc+nd+nf}{N} \ln(N)$

$(nb = nb+nc+nd+nf)$

$= \ln(\hat{\lambda}_N^4[nb,nc,nd,nf] \cdot e^{(nb+nc+nd+nf) \cdot \ln(N)/N}) + \frac{nb+nc+nd}{N} \ln(N)$

$GAIC-R_N^4$  cu  $\hat{\lambda}_N^4$  modificat pt. fiecare nf

\* Implementare: - se construiesc un bloc 4D format din paralelipipede 3D de tip GAIC-R<sup>3</sup>, folosind:

- un ciclu for
- functia CAT

Apelul GAIC\_R3.m in "for" indica minimul din fiecare matrice 3D. Minimele se salveaza intr-un vector. Minimul din tot blocul 4D se afla apeland "min" pentru acest vector, dupa "for". Minimul e cunoscut, deci si pozitia in matricea 4D e cunoscuta. Aceasta e descrisa de indicii structurali => indicii structurali optimi corespunzatori minimului.