Identificarea Sistemelor

LABORATOR 5

Mărgăritescu Vlad - 342B3

PROBLEMA 1 (MMEP pentru modelul ARMAX)
Am rulat rutina ISLAB_6A si am atasat rezultatele:

* Proposed optimal indices:
<F-test on prediction error>: [na nb nc] = [3 3 2]
<F-test on fitness (identification data)>: [na nb nc] = [3 2 2]
<F-test on fitness (validation data)>: [na nb nc] = [3 2 2]
<GAIC-Rissanen criterion>: [na nb nc] = [2 2 1]

```
# Insert optimal indices [na nb nc]: [2 2 1]

o Optimum model:


Mid =
Discrete-time ARMAX model: A(z)y(t) = B(z)u(t) + C(z)e(t)
  A(z) = 1 - 1.474 z^-1 + 0.6829 z^-2

  B(z) = 0.9767 z^-1 + 0.5718 z^-2

  C(z) = 1 - 0.8502 z^-1

Sample time: 1 seconds

Parameterization:
   Polynomial orders:   na=2   nb=2   nc=1   nk=1
   Number of free coefficients: 5
   Use "polydata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using ARMAX on time domain data "Did".
Fit to estimation data: 77.15% (prediction focus)
FPE: 0.9635, MSE: 0.9257

Model Properties

                                <Press a key>
```

Fig 1. Identificarea indicilor structurali optimi – ISLAB_6A_1

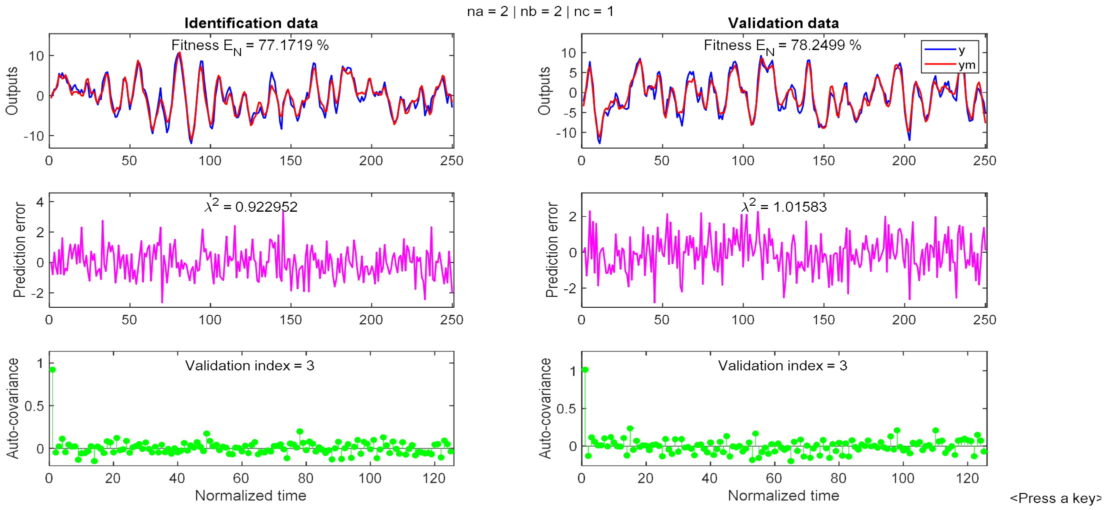

Fig 2. Performante model ARMAX identificat cu MMEP – ISLAB_6A_1

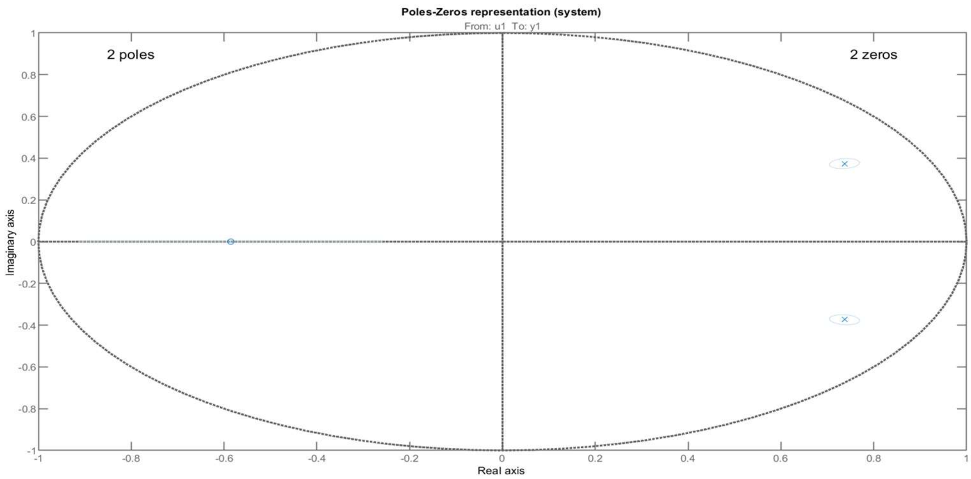

Fig 3. Reprezentarea poli-zerouri (intrare) – ISLAB_6A_1

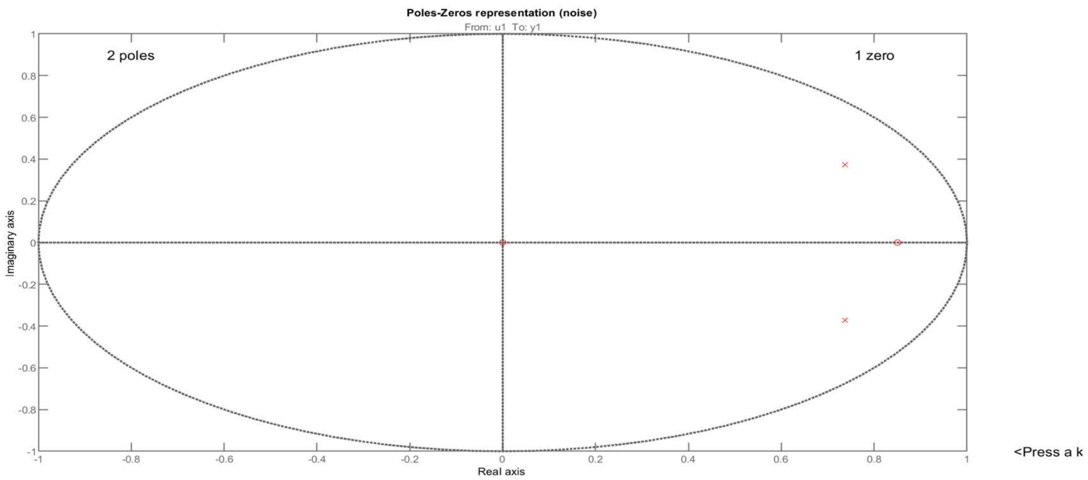

Fig 4. Reprezentarea poli-zerouri (zgomot) – ISLAB_6A_1

Am rulat inca o data rutina ISLAB_6A:

* Proposed optimal indices:

<F-test on prediction error>: [na nb nc] = [3 2 2]
<F-test on fitness (identification data)>: [na nb nc] = [3 2 2]
<F-test on fitness (validation data)>: [na nb nc] = [3 2 2]
<GAIC-Rissanen criterion>: [na nb nc] = [3 1 2]

```
# Insert optimal indices [na nb nc]: [3 1 2]

o Optimum model:


Mid =
Discrete-time ARMAX model: A(z)y(t) = B(z)u(t) + C(z)e(t)
  A(z) = 1 - 1.856 z^-1 + 1.232 z^-2 - 0.241 z^-3

  B(z) = 1.014 z^-1

  C(z) = 1 - 1.339 z^-1 + 0.4851 z^-2

Sample time: 1 seconds

Parameterization:
   Polynomial orders:   na=3   nb=1   nc=2   nk=1
   Number of free coefficients: 6
   Use "polydata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using ARMAX on time domain data "Did".
Fit to estimation data: 78.08% (prediction focus)
FPE: 1.15, MSE: 1.096
```

Model Properties

<Press a key>

Fig 5. Identificarea indicilor structurali optimi – ISLAB_6A_2

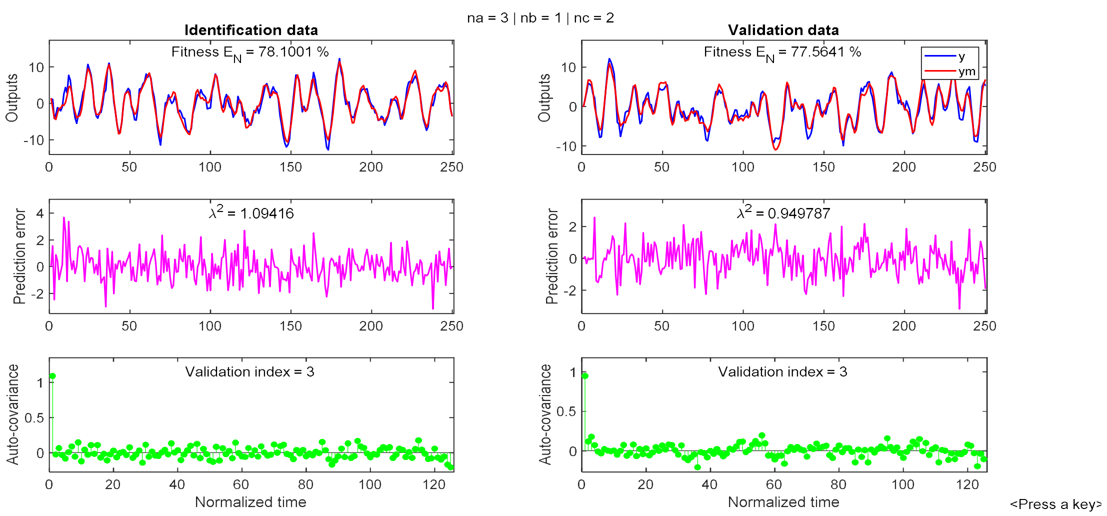

Fig 6. Performante model ARMAX identificat cu MMEP – ISLAB_6A_2

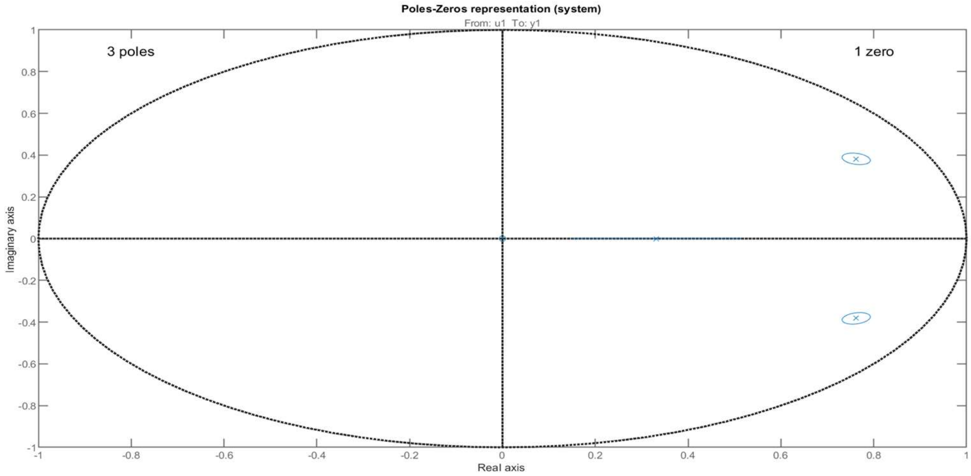

Fig 7. Reprezentarea poli-zerouri (intrare) – ISLAB_6A_2

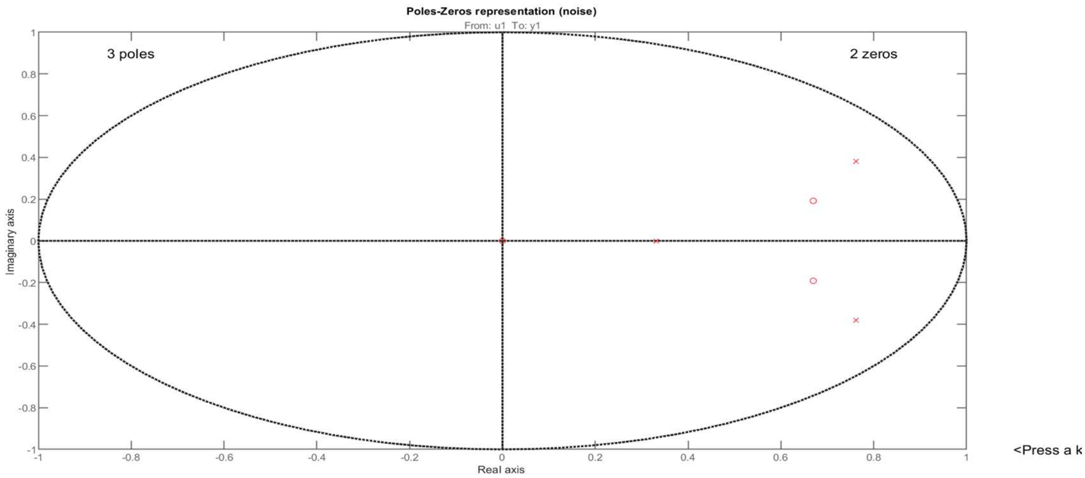

Fig 8. Reprezentarea poli-zerouri (zgomot) – ISLAB_6A_2

Concluzii:
Functia de potrivire E_N si eroarea de predictie lambda^2 sunt diferite pentru fiecare rulare.
Indicii structurali nu difera foarte mult intre rulari.
In mod ideal, lamba^2 tinde spre 1.
Indicele de validare este egal cu 3 in ambele cazuri, astfel modelele sunt valide.

PROBLEMA 2 (MMEP pentru modelul BJ)

ISLAB_6B

Se modifica rutina cu cele 4 componente specifice BJ: B,C,D,F.

```
Inputs:          B = ([1 0.5], by default)
%                C = ([1 -1 0.2], by default)
%                D = ([1 1.5 0.7], by default)
%                F = ([1 -1.5 0.7], by default)
```

Se inlocuiesc indicii de la punctul a (na,nb,nc) cu (nb,nc,nd,nf) peste tot.

```
pf = 0 ;
Nb = 5 ;
Nc = 5 ;
Nd = 5 ;
Nf = 5 ;
Ts = 1 ;
```

Se testeaza stabilitatea pentru cele 2 polinoame F si D.

```
F = roots(F) ;
F(abs(F)>=1) = 1./F(abs(F)>=1) ;      % Correct the stability.
F = poly(F) ;

D = roots(D) ;
D(abs(D)>=1) = 1./D(abs(D)>=1) ;      % Correct the stability.
D = poly(D) ;
```

Apelul armax este inlocuit de apelul bj.
GAIC_R3 este inlocuit de GAIC_R4.
La diagramele poli-zerouri au loc schimbari:
   -La primul subpunct aveam 2 imagini pentru (nb zerouri si na poli) & (nc zerouri si na poli) deoarece:
ARMAX: y[n] = B/A * u[n] + C/A * e[n].
   -La acest subpunt avem 2 imagini pentru (nb zerouri si nf poli) & (nc zerouri si nd poli) deoarece:
BJ: y[n] = B/F * u[n] + C/D * e[n].
In plus s-au adaugat alte moficicari necesare functionarii rutinei.

GAIC_R4

Se bazeaza pe GAIC_R3, dar avem 4 indici (nb,nc,nd,nf) in loc de 3 (na,nb,nc).
In mare parte, nc se inlocuieste cu nf si restul se ajusteaza.
Se foloseste formula pusa in fisierul pdf GAICs.

Se ruleaza rutina ISLAB_6B proiectata:

```
* Proposed optimal indices:

<F-test on prediction error>: [nb nc nd nf] = [ 3 1 3 3]

<F-test on fitness (identification data)>: [nb nc nd nf] = [ 2 1 1 3]

<F-test on fitness (validation data)>: [nb nc nd nf] = [ 2 1 1 3]

<GAIC-Rissanen criterion>: [nb nc nd nf] = [ 2 1 2 2]
```

```
# Insert optimal indices [nb nc nd nf]: [2 1 1 3]

o Optimum model:


Mid =
Discrete-time BJ model: y(t) = [B(z)/F(z)]u(t) + [C(z)/D(z)]e(t)
  B(z) = 1.042 z^-1 + 0.6849 z^-2

  C(z) = 1 + 0.1681 z^-1

  D(z) = 1 - 0.4712 z^-1

  F(z) = 1 - 1.388 z^-1 + 0.5448 z^-2 + 0.06985 z^-3

Sample time: 1 seconds

Parameterization:
   Polynomial orders:   nb=2   nc=1   nd=1   nf=3   nk=1
   Number of free coefficients: 7
   Use "polydata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using BJ on time domain data "Did".
Fit to estimation data: 76.45% (prediction focus)
FPE: 1.04, MSE: 0.983

Model Properties

                                      <Press a key>
```

Fig 9. Identificarea indicilor structurali optimi – ISLAB_6B

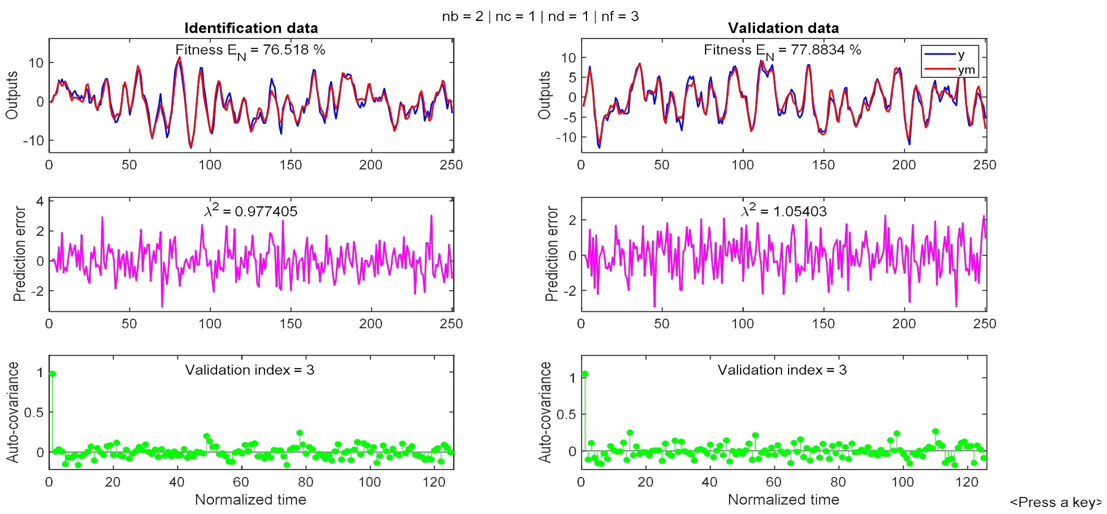

Fig 10. Performante model BJ identificat cu MMEP – ISLAB_6B

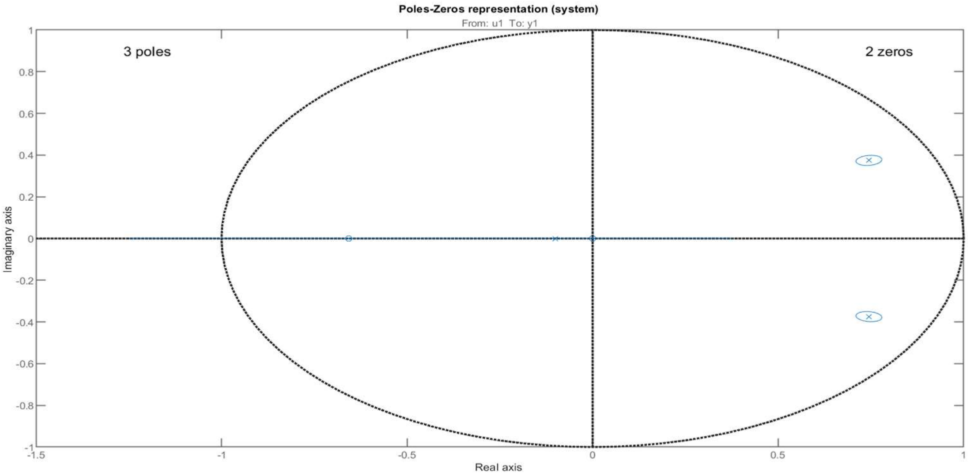

Fig 11. Reprezentarea poli-zerouri (intrare) – ISLAB_6B

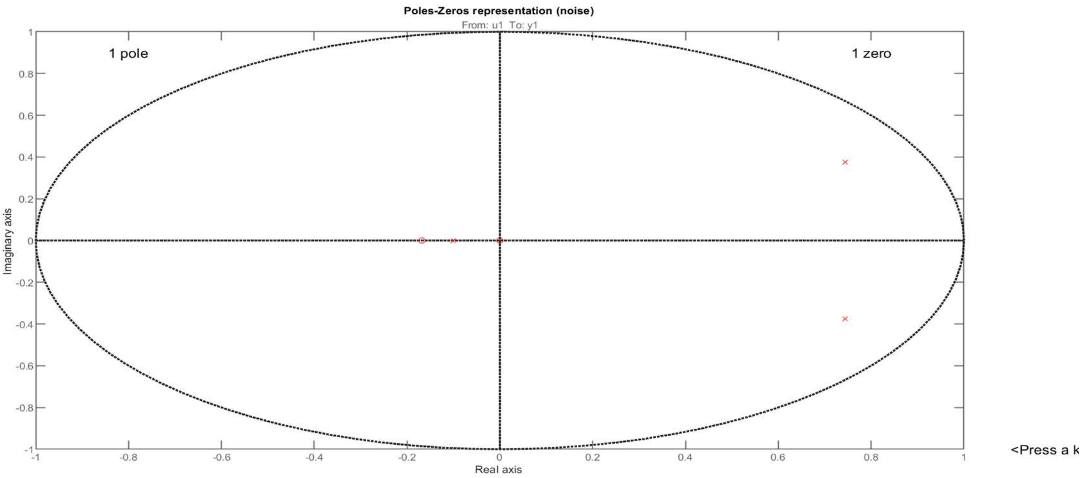

Fig 12. Reprezentarea poli-zerouri (zgomot) – ISLAB_6B

Concluzii:

Modelele BJ au 4 indici structurali, in loc de 3 cum este in cazul ARMAX, astfel si timpul de rulare pentru o asemenea metoda de identificare este mai mare.

Se observa si in acest caz diferente intre rulari cand vine vorba de indicii optimi, dar acestea sunt in continuare mici.

Diferentele de performanta sunt minime intre BJ si ARMAX, dar lambda^2 tinde mai aproape de 1 in cazul BJ.

Pentru o rulare mai rapida am redus valoarea maxima a indicilor structurali la 3, in loc de 5. Astfel, se obtin rezultate putin mai slabe, dar timpul de executie este mult mai mic.

PROBLEMA 3(MCMMPE pentru modelele ARMAX si BJ)

ISLAB_6C

In aceasta rutina voi folosi toti indicii structurali posibili, urmand sa aleg ce am nevoie la pasul urmator.

```
pf = 0 ;
Na = 5 ;
Nb = 5 ;
Nc = 5 ;
Nd = 5 ;
Nf = 5 ;
Ts = 1 ;
```

Am implementat selectia metodei dorite: ARMAX sau BJ pentru cazul MCMMPE

```
% Alegere metoda: ARMAX/BJ
disp('Alegeti metoda MCMMPE:');
disp('1: ARMAX');
disp('2: BJ');
disp('3: Stop;');
metoda = input('Metoda: ', 's');
```

Se creaza rutinele armax_e si bj_e.

Cazul 1. ARMAX

In mare parte se bazeaza pe ISLAB_6A, dar apelul este pentru functia **armax_e**.

armax_e:

```
function [Mid] = armax_e(Did, si)
      %armax_e implementeaza MCMMPE pentru ARMAX
      %si = [na nb nc nk]

    Ts = 1;
    N = 250;
      if (nargin < 1)
             si = [2 2 2 1];
      end
      if (isempty(si) || length(si) < 4) %este o eroare care zice maxim 2 elemente
             si = [2 2 2 1];
      end

      %Indicii structurali (na,nb,nc,nk=1):
      na = si(1);
      nb = si(2);
      nc = si(3);
      nk = si(end);

    %Declarare n_alpha & n_beta
    %Se impune conditia min(n_alpha,n_beta) >> max(na,nb,nc)
      n_alpha = max([na,nb,nc])*2;
      n_beta = 2*n_alpha;

    %Identificare model ARX cu setul de date Did si n_alpha si n_beta
      Mid = arx(Did,[n_alpha n_beta nk]);

    % Estimarea zgomotului ARX:
    e = pe(Mid, Did);

    %Crearea celor 3 componente y,u,e:
    e = e.y;
    y = Did.y;
    u = Did.u;
    y = [zeros(na, 1); y];
    e = [zeros(nc, 1); e];
    u = [zeros(nb, 1); u];
```

```
    %Structura R_N si r_n
    R_N = zeros(na+nb+nc, na+nb+nc);
    r_n = zeros(na+nb+nc, 1);

    %Calcul efectiv

    for i = 1:N %pentru fiecare n pana la N
        phi_y = -y(i+na-1:-1:i); %iesirea
        phi_u = u(i+nb-1:-1:i); %intrarea
        phi_e = e(i+nc-1:-1:i); %zgomotul

        %Explicatie: Se cer u[n-1] u[n-2] ... u[n-nb], adica nb termeni
        %Se considera ca n>nb pentru ca nu avem voie cu indici negativi
        %Astfel, daca nb este 4 => n incepe de la 5, astfel termenii sunt:
        %Avem u[5-1], u[5-2], u[5-3], u[5-nb] adica u[5-4].

        % Construim phi prin concatenarea secțiunilor
        phi = [phi_y; phi_u; phi_e];

        %Se calculeaza recursiv R_N si r_n pentru fiecare pas din for:
        R_N = R_N + 1/N *(phi * phi');
        r_n = r_n + 1/N * phi * Did.y(i);
    end

    %Se afla R_N si r_n finale si se afla theta
    theta = R_N\r_n; %theta = inv(R_N)*r_n

    %Definire vectori A,B,C folositi pentru ARMAX
    A = [1; theta(1:na)]'; %primul coeficient trebuie 0 (nu merge altfel)
    B = [0; theta(na+1:na+nb)]'; %primul coeficient trebuie 1 (nu merge altfel)
    C = [1; theta(na+nb+1:end)]'; %primul coeficient trebuie 0 (nu merge altfel)

    Mid = idpoly(A,B,C,1,1,1,Ts); %Model ARMAX (A,B,C,D=1,F=1)

end
```

* Proposed optimal indices:

<F-test on prediction error>: [na nb nc] = [3 2 2]

<F-test on fitness (identification data)>: [na nb nc] = [3 2 2]

<F-test on fitness (validation data)>: [na nb nc] = [3 2 2]

<GAIC-Rissanen criterion>: [na nb nc] = [0 0 1]

Insert optimal indices [na nb nc]: [3 2 2]

o Optimum model:

Mid =
Discrete-time ARMAX model: A(z)y(t) = B(z)u(t) + C(z)e(t)
A(z) = 1 - 1.149 z^-1 + 0.1556 z^-2 + 0.2473 z^-3

B(z) = 1.148 z^-1 + 0.7631 z^-2

C(z) = 1 - 0.6096 z^-1 - 0.2194 z^-2

Sample time: 1 seconds

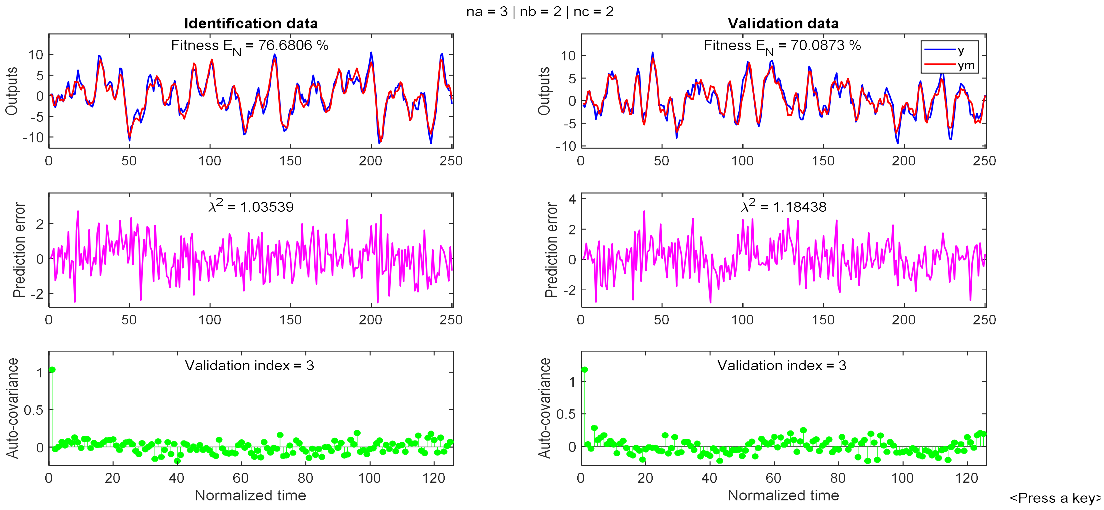

Fig 13. Performante model BJ identificat cu MMEP – ISLAB_6C_ARMAX

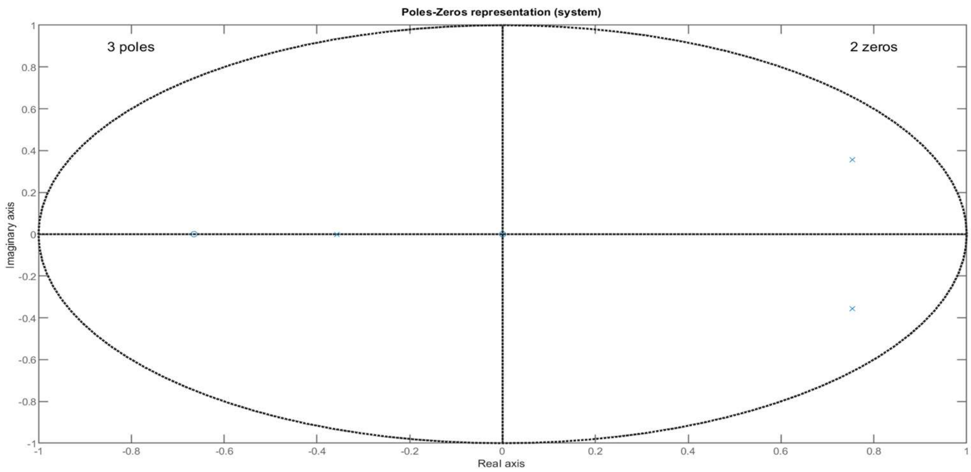

Fig 14. Reprezentarea poli-zerouri (intrare) – ISLAB_6C_ARMAX

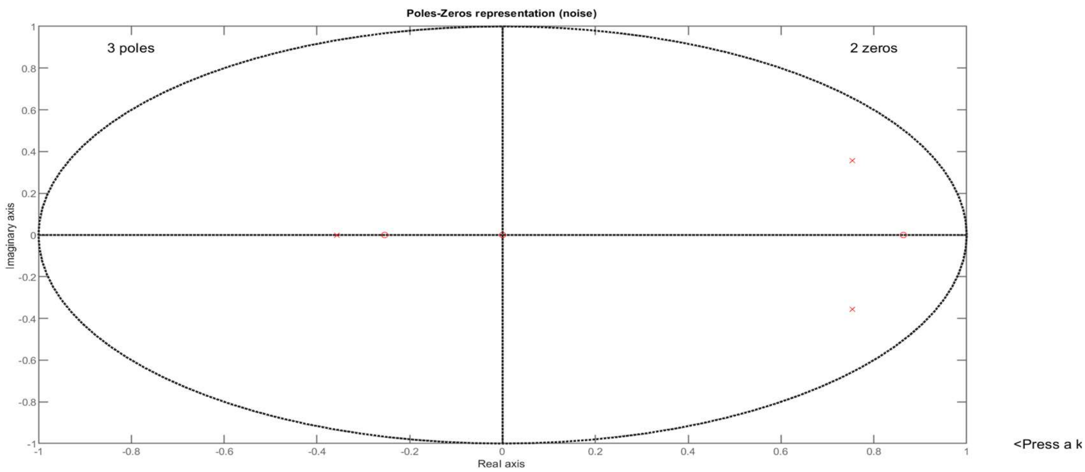

Fig 15. Reprezentarea poli-zerouri (zgomot) – ISLAB_6C_ARMAX

Cazul 2. BJ

In mare parte se bazeaza pe ISLAB_6B, dar apelul este pentru functia **bj_e**.

bj_e:

```
function Mid  = bj_e(Did, si)
      %bj_e implementeaza MCMMPE pentru BJ
      %si = [nb nc nd nf nk]

    if (nargin < 1) % si
       si = [5 5 5 5 1];
    end
    if (isempty(si))
       si = [5 5 5 5 1];
    end

    %Indicii structurali (nb,nc,nd,nf,nk=1):
    nb = si(1);
    nc = si(2);
    nd = si(3);
    nf = si(4);
    nk = si(5);

    %Se foloseste functia armax_e proiectata anterior
    Mid = armax_e(Did,[nf+nd nb+nd nc+nf nk]);

    %Pornind de la radacinile A,B,C pt ARMAX, se cauta radacinile B,C,D,F pt BJ
    radacini_A = roots(Mid.A);
    radacini_B = roots(Mid.B);
    radacini_C = roots(Mid.C);

    %Polinomul D (Radacinile comune A si B)
    radacini_D = intersect(radacini_A, radacini_B);
    D = poly(radacini_D);

    %Polinomul F (Radacinile comune A si C)
    radacini_F = intersect(radacini_A, radacini_C);
    F = poly(radacini_F);

    %Polinomul C (Se extrag radacinile F din C)
    radacini_C = setdiff(radacini_C, radacini_F);
    C = poly(radacini_C);

    %Polinomul B (Se extrag radacinile D din B)
    radacini_B = setdiff(radacini_B, radacini_D);
    B = poly(radacini_B);

    Mid = idpoly(1,B,C,D,F,1,1); %Model BJ (A=1,B,C,D,F)

end
```

* Proposed optimal indices:

<F-test on prediction error>: [nb nc nd nf] = [1 0 2 0]

<F-test on fitness (identification data)>: [nb nc nd nf] = [1 1 1 0]

<F-test on fitness (validation data)>: [nb nc nd nf] = [1 1 0 2]

<GAIC-Rissanen criterion>: [nb nc nd nf] = [2 2 2 3]

Insert optimal indices [nb nc nd nf]: [2 2 2 3]

o Optimum model:

Mid =
Discrete-time Polynomial model: y(t) = B(z)u(t) + C(z)e(t)
 B(z) = 1 + 1.535 z^-1 + 0.8263 z^-2 + 0.4109 z^-3

 C(z) = 1 - 0.1756 z^-1 - 0.3001 z^-2 - 0.1293 z^-3 + 0.07818 z^-4 - 0.04088 z^-5

Sample time: 1 seconds

Parameterization:
 Polynomial orders: nb=4 nc=5 nk=0
 Number of free coefficients: 9
 Use "polydata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Created by direct construction or transformation. Not estimated.

Model Properties

<Press a key>

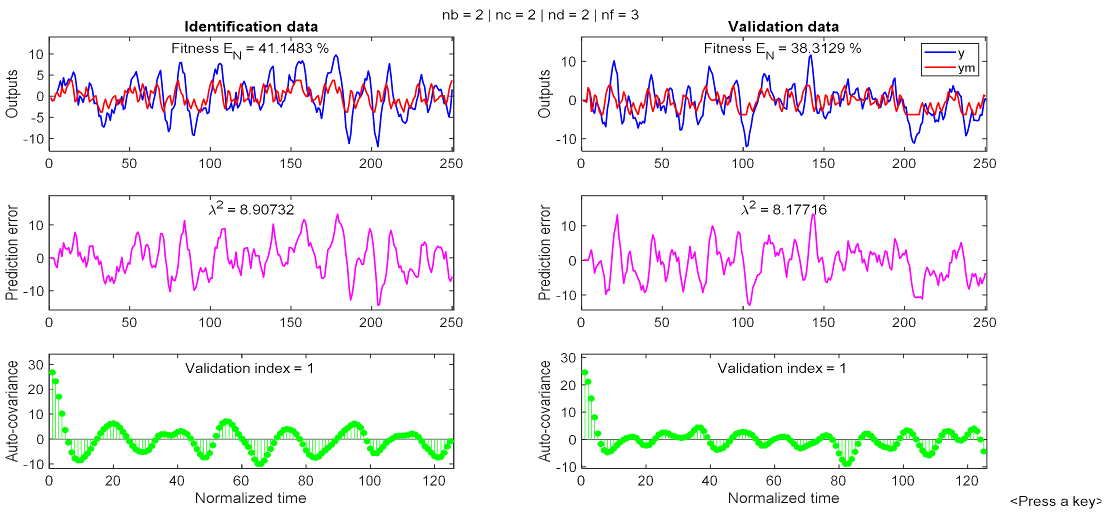

Fig 16. Performante model BJ identificat cu MMEP – ISLAB_6C_BJ

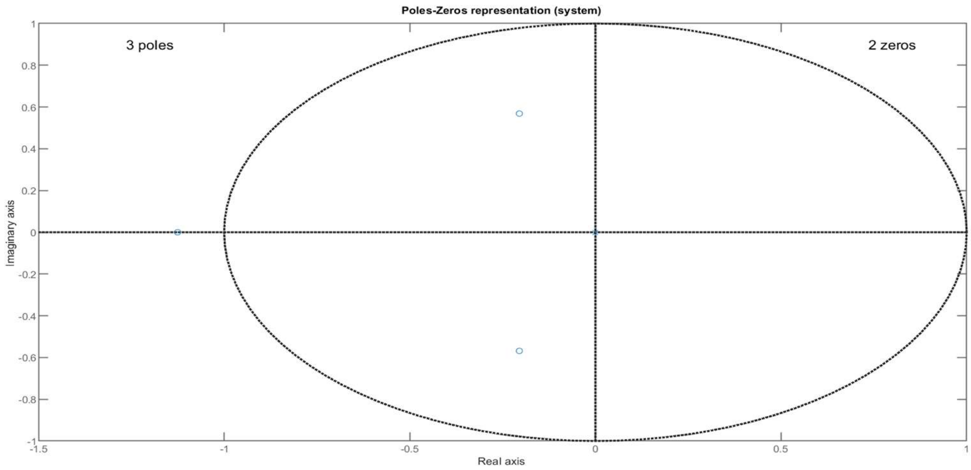

Fig 17. Reprezentarea poli-zerouri (intrare) – ISLAB_6C_BJ

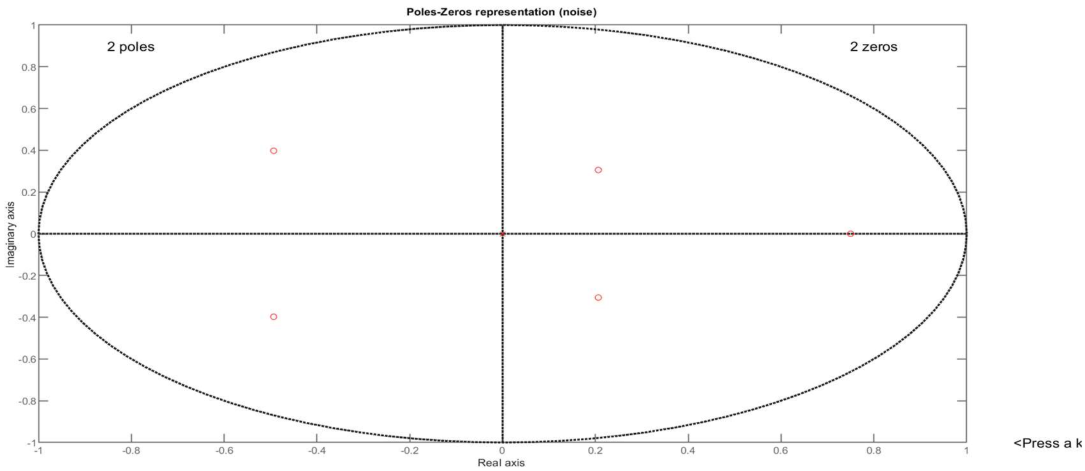

Fig 18. Reprezentarea poli-zerouri (zgomot) – ISLAB_6C_BJ

Concluzii:

Se observa cum varianta MCMMPE ofera performante mai scazute decat MMEP, tocmai de aceea varianta MMEP este mai eficienta.

Acest simulator ofera performante asemanatoare cu cele precedente in cazul ARMAX, insa in cazul BJ performantele sunt foarte slabe iar indexul de validare este in general 1. Asta inseamana un model valid, dar cu valididate slaba.

Pentru cazul MCMMPE – ARMAX, rezultatele sunt mai bune daca marim indicii structurali, insa in cazul MCMMPE – BJ, rezultatatele sunt slabe indiferent de indicii structurali.