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]

A screenshot of a computer program

Description automatically generated

Fig 1. Identificarea indicilor structurali optimi – ISLAB\_6A\_1



Fig 2. Performante model ARMAX identificat cu MMEP – ISLAB\_6A\_1

A graph of a function

Description automatically generated

Fig 3. Reprezentarea poli-zerouri (intrare) – ISLAB\_6A\_1

A graph of a function

Description automatically generated

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]

A screenshot of a computer

Description automatically generated

Fig 5. Identificarea indicilor structurali optimi – ISLAB\_6A\_2



Fig 6. Performante model ARMAX identificat cu MMEP – ISLAB\_6A\_2

A graph of a function

Description automatically generated

Fig 7. Reprezentarea poli-zerouri (intrare) – ISLAB\_6A\_2

A graph of a function

Description automatically generated

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.

A number and letters on a white background

Description automatically generated

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]

**A screenshot of a computer

Description automatically generated**

Fig 9. Identificarea indicilor structurali optimi – ISLAB\_6B

****

Fig 10. Performante model BJ identificat cu MMEP – ISLAB\_6B

**A graph of a function

Description automatically generated**

Fig 11. Reprezentarea poli-zerouri (intrare) – ISLAB\_6B

A graph of a function

Description automatically generated

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.

A number and symbols on a white background

Description automatically generated

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

**A graph of a function

Description automatically generated**

Fig 14. Reprezentarea poli-zerouri (intrare) – ISLAB\_6C\_ARMAX

**A graph of a function

Description automatically generated**

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

A graph of a function

Description automatically generated with medium confidence

Fig 17. Reprezentarea poli-zerouri (intrare) – ISLAB\_6C\_BJ

A graph of a function

Description automatically generated

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.