Identificarea Sistemelor

LABORATOR 6

Mărgăritescu Vlad - 342B3

**PROBLEMA 1 (Identificare recursiva comparativa cu algoritmul de baza)**

Cu ajutorul mini-simulatorului ISLAB\_7A se efectueaza o comparatie intre cele 4 metode, folosind setul de date Dc (generate de procesul cu parametri constanti).

**ISLAB\_7A**



Fig. 1 – MCMMP-R



Fig. 2 – MVI-R



Fig. 3 – MMEP-R



Fig. 4 – MRPL-R

**Concluzii:**

a) Care ar fi explicaţiile performanţelor mai slabe ale MCMMP-R în estimarea parametrului părţii AR?

Solutie: Se observa cum in figura 1, componenta AR nu este identificata corect de catre metoda MCMMP-R, deoarece primele 2 metode sunt aplicate unui model ARX[1,1] extras dintr-un model ARMAX, astfel apar erori de identificare si aproximare.

Ultimele 2 opereaza cu modelul ARMAX[1,1,1] si se vede cum estimatiile sunt consistente si mai precise.

**ISLAB\_7B**

La acest punct trebuie sa apelam functia gdata\_vp, dar cu cv!=0, intrucat cv este parametrul care face schimbarea dintre parametri constanti si parametri variabili.

A close up of a computer screen

Description automatically generated

% Generating the identification data (ARMAX)

% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

[D,~,P] = gdata\_vp(1,N,sigma,lambda,0) ;



Fig. 5 – MCMMP - R – param variabili



Fig. 6 – MVI - R – param variabili



Fig. 7 – MMEP - R – param variabili



Fig. 8 – MRPL - R – param variabili

**Concluzii:**

Se poate observa cum aceasta metoda ofera niste solutii foarte slabe.

Aproape toate estimatiile sunt inconsistente. Doar in cazul MVI avem estimatii consistente.

Astfel, folosind parametri variabili, duce la un rezultat chiar mai slab fata de metodele cu parametri constanti.

In concluzie, trebuie evitata aceasta abordare.

**PROBLEMA 2 (Identificare recursiva cu diferite initializari)**

Se adauga in cod vectorul alpha si se modifica functiile rarx si riv pentru a include si P0.

Se combina cele 2 subrutine precedente: ISLAB\_7A si ISLAB\_7B pentru a include cazurile cu parametri constanti si parametri variabili. (parametrul cv=0/1)

Pentru fiecare alpha se afiseaza cele 4 grafice, adica un total de 28 de grafice.

****

Fig. 9 – MCMMP - R – param constanti

Toate graficele arata la fel pentru orice valoare a lui alpha



Fig. 10 – MVI - R – param constanti

Toate graficele arata la fel pentru orice valoare a lui alpha



Fig. 11 – MCMMP - R – param variabili



Fig. 12 – MVI - R – param variabili, alpha = 0.001



Fig. 13 – MVI - R – param variabili, alpha = 1



Fig. 14 – MVI - R – param variabili, alpha = 1000

**Concluzii:**

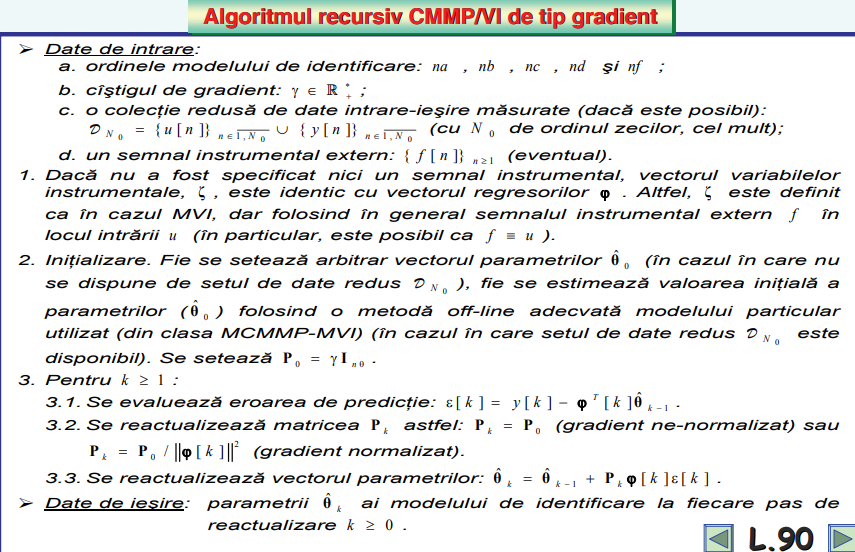
Pentru metodele cu parametri constanti cele 7 grafice sunt la fel pentru orice alpha. Acelasi lucru se observa si pentru MCMMP-R cu param variabili.

Pentru MVI-R cu parametrii variabili in schimb, graficele se schimba pentru fiecare valoare a lui alpha.

La valori mici ale lui alpha, cu cat valoarea lui alpha creste mai mult, cu atat graficul se apropie mai mult de referinta. Dar atunci cand alpha devine supraunitar graficul atinge un apogeu si ramane aproximativ constant indiferent de cat creste valoarea lui alpha.

**PROBLEMA 3 (Identificare recursiva comparativa)**

**Rutina arx\_nabla:** Se pleaca de la algoritmul prezentat la ISLAB\_7B si se modifica conform paginii 90 din laborator.



%%% ARX\_NABLA

%1 Input

gain = input('Castigul de gradient = ') ;

[theta,ypred] = rarx(D,[1 1 1],'ng',gain) ; % Se schimba linia aici

% Definire phi (Date iesire, Date intrare)

%Calcul Parametri

phi = [D.y; D.u];

Rn = 0;

rn = 0;

for n = 1:N

Rn = Rn+(1/N)\*(phi\*phi');

rn = rn+(1/N)\*(phi\*D.y(n));

end

theta = Rn^(-1)\*rn ;

%2 Initializare

Priv = gain\*eye(length(theta)) ;

%3

for k = 1 :N

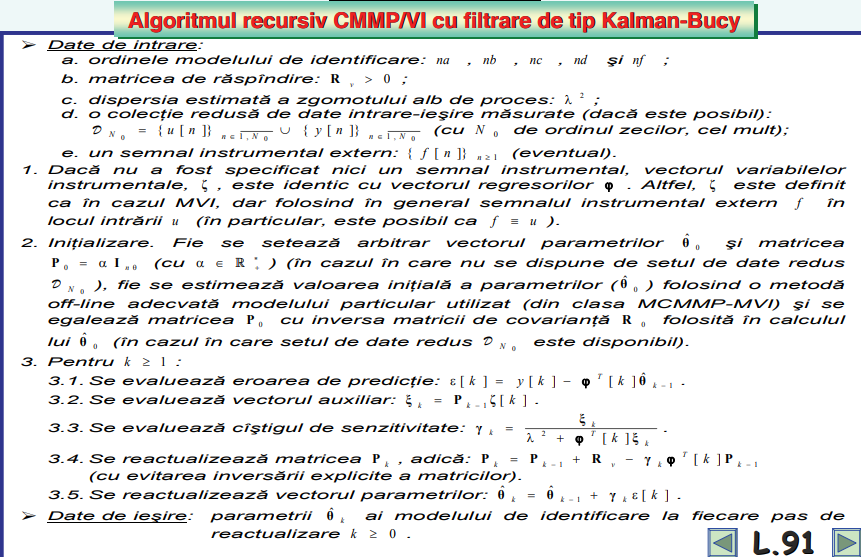
eps(k) = D.y(k) - phi'\*theta;

Priv = Priv/(norm(phi))^2;

theta = theta + Priv\*phi\*eps(k);

end

**Rutina arx\_KB:** Se pleaca de la algoritmul prezentat la ISLAB\_7B si se modifica conform paginii 91 din laborator.



%%% ARX\_KB

% Input

lambdaPatrat = 1 ;

[theta,ypred] = rarx(D,[1 1 1],'kf',Rv) ; %Se modifica aici

% Definire phi + Calcul Parametri

phi = [D.y; D.u];

Rn = 0;

rn = 0;

tau = phi;

for n = 1:N

Rn = Rn+(1/N)\*(phi\*phi');

rn = rn+(1/N)\*(phi\*D.y(n));

end

theta = Rn^(-1)\*rn ;

%2 Initializare

alpha = 1; %Se alege alpha

Priv = alpha\*eye(length(theta)) ;

Priv = Rn^(-1) ;

Rv =ones(size(Priv)) ;

%3

for k = 1 : N

eps(k) = D.y(k) - phi'\*theta ;

tzeta = Priv\*tau ;

gain = tzeta/(lambdaPatrat + phi'\*tzeta);

Priv = Priv + Rv - gain\*phi'\*Priv ;

theta = theta + gain\*eps(k) ;

**ISLAB\_7D:** Se proiecteaza asemanator cu ISLAB\_7B, dar se adauga toate variantele:

disp('Lista celor 6 variante: ');

disp('1 = MCMMP-R cu parametri variabili');

disp('2 = MVI-R cu parametri variabili');

disp('3 = MMEP-R cu parametri variabili');

disp('4 = MRPL-R cu parametri variabili');

disp('5 = CMMP/VI de tip gradient');

disp('6 = CMMP/VI cu filtrare de tip Kalman-Bucy');

optiune = input('Alegere varianta: ') ;

Se foloseste switch si case pt toate variantele.



Fig. 15 – MCMMP – R cu param variabili



Fig. 16 – MVI – R cu param variabili



Fig. 17 – MMEP – R cu param variabili



Fig. 18 – MRPL – R cu param variabili



Fig. 19 – CMMP/VI de tip gradient cu castigul = 1



Fig. 20 – CMMP/VI cu filtrare de tip Kalman-Bucy cu alpha = 1

**Concluzii:**

Se observa cum MCMMP are cea mai inconsistenta urmarire.

Cea mai buna urmarire o are CMMP/VI de tip gradient.

Si CMMP/VI de tip KB are o consistenta buna.

Asadar, opinia mea este ca ordinea solutiilor (de la cea mai buna la cea mai slaba) este:

CMMP/VI-gradient > CMMP/VI-KB > MVI-R > MRPL-R > MMEP-R > MCMMP-R.