UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

Proiect Identificarea Sistemelor -partea 2

Arx neliniar

Ianuarie-2024

Nume studenți:

Florean Ioana-Valentina

Tăbîrcea Georgiana-Maria

Ciceu Robert

Grupa 30135/1

Setul de date: iddata-18.mat

Cuprins

Descrierea problemei

- Acest proiect are ca o obiectiv dezvoltarea unui model ARX neliniar bazat pe un polinom cu regresie. Configurațiile modelului , precum ordinele pentru termenii liniari (na și nb) și gradul polinomului neliniar (n), sunt parametrizabile,permițând ajustarea flxibilă a modelului în funcție de caracteristicile specifice modelului.
- Proiectul se bazează pe două seturi distincte de date
  - Date de identificare (notat cu id)

  -folosit pentru antrenarea datelor

  - Date de validare (notat cu val)

  -folosit pentru evaluarea performanței modelului dezvoltat

Descrierea structurii aproximatorului si a procedurii de găsire a parametrilor

- Gradul aproximatorului poate fi configurat, iar alegerea gradului optim depinde de complexitatea sistemului care trebuie identificat.
- Procedura de găsire a parametrilor constă în dezvoltarea unei matrici d() care este compusă din termenii liniari și neliniari ai unui model ARX. Apoi găsim vectorul optim de parametrii θ pentru care polinomul se apropie cât mai mult de funcția necunoscută.
- Modelul ARX neliniar are structura:

yˆ(k) = p(y(k-1), . . . , y(k-na), u(k-nk), u(k-nk-1), . . . , u(k-nk-nb+1)) = p(d(k)) , unde

  - d(k) este vectorul de ieșiri și intrări întârziate, notat d(k) = [y(k-1), . . . , y(k-na), u(k-nk), u(k-nk-1), . . . , u(k-nk-nb+1)]^T
  - p() este un polinom de grad m ale variabilelor din d(k)

- Antrenarea modelului constă în găsirea vectorului optim de parametrii θ pentru care funcția p() se apropie cât mai bine de relația dintre semnalele de intrare și ieșire.Aceasta minimizează discrepanța dintre datele observate și cele prezise.

În continuare vă vom prezenta părțile esențiale ale codului. În primul rând vă vom arăta matricea de identificare d a unui model ARX neliniar cu regresie.

```
% Construirea matricei de regresie
d = [];
for i = 1:N
% Prima coloană întotdeauna 1
d(i,1) = 1;
% Se calculeaza termenii liniari ARX
for j=2:na+1
if i-j>0
d(i,j)=-y_id(i-j);
else
d(i,j)=0;
end
end
% Se calculeaza partea 2 a matricii cu termeni ARX
for j=na+2:na+nb+1
if i-(j-na)>0
d(i,j)=u_id(i-(j-na));
else
d(i,j)=0;
end
end
```

```
% Termeni polinomiali neliniari pentru datele de identificare
index=na+nb+2;
for m=2:n % m putere
for w=0:m % w putere
if i-w>0&&i-(j-na)>0
parte_liniara = d(i,j);
parte_neliniara = d(i-w,j-na);
d(i, index) = -parte_neliniara^(m - w) * parte_liniara^w;
% Se trece la urmatoarea coloană
index=index+1;
else
d(i,index)=0;
index=index+1;
end
end
end
end
```

O alta parte importanta este validarea prin simulare

```
% Validare prin simulare
ysim=zeros(Nval,1);
for i=2:Nval
y1=0; % Suma pentru toate y-urile
y2=0; % Suma pentru toate u-urile
y3=0; % Suma pentru toate y-urile si u-urile neliniare
% Termeni liniari ARX pentru y
for j=1:na
if i-j>0
y1=y1-theta(j)*ysim(i-j);
end
end
% Termeni liniari ARX pentru u
for j=1:nb
if i-j>0
y2=y2+theta(na+j)*u_val(i-j);
end
end
% Termeni neliniari ARX
index=na+nb+2;
for m =2:n % m putere
for w=0:m % w putere
if i-w>0 && i-(na+1) > 0
parte_liniara = d_val(i,j);
parte_neliniara = d_val(i-w, na+1);
y3 = y3-theta(index)*parte_neliniara^(m - w)*parte_liniara^w;
index = index + 1;
end
end
end
%ysim(i) = y1 + y2 + y3;
ysim(i)=y3;
end
```

Caracteristici esențiale ale soluției

- Sunt specificate ordinele polinomului de ieșire (na) și polinomului de intrare (nb), de exemplu na=3 și nb=3 si n=4.
- Matricea de regresie (d) este creată pentru datele de identificare (u_id, y_id) , respectiv pentru datele de validare (u_val,y_val)

d(i, j) = -y_id(i - j); - pentru termenii liniari ai ieșirii

d(i, j) = u_id(i - (j - na)); - pentru termenii liniari ai intrării

Cu ajutorul partii liniare vom creea partea neliniara: *parte_neliniara = d(i-w,j-na)* apoi acestea ne ajuta in gasirea matricii d() finale: *d(i, index) = -parte_neliniara^(m - w) * parte_liniara^w;*

Parametrii modelului (θ) sunt calculați utilizând: *tetha = d \ y_id;*

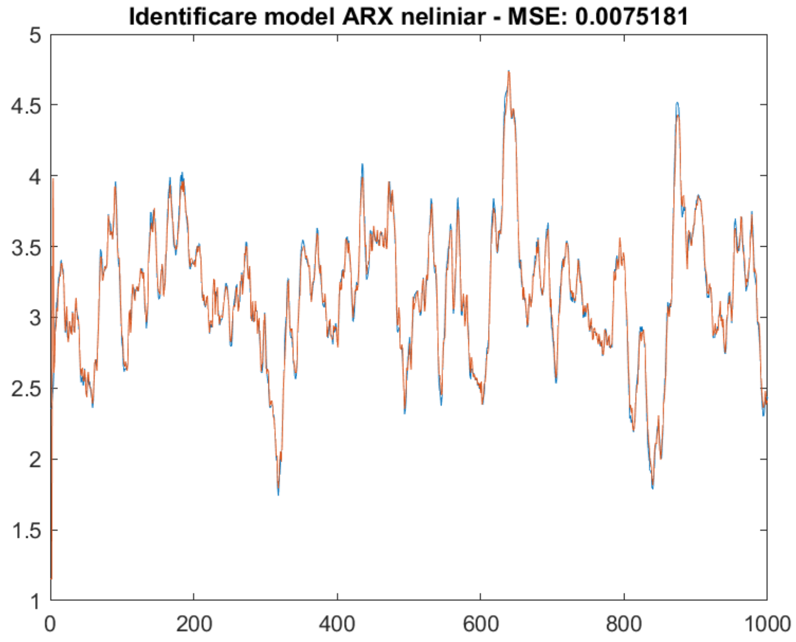
Se simulеză modelul identificat pe datele de validare:*(y_hat) y_hat = d_val * tetha;*

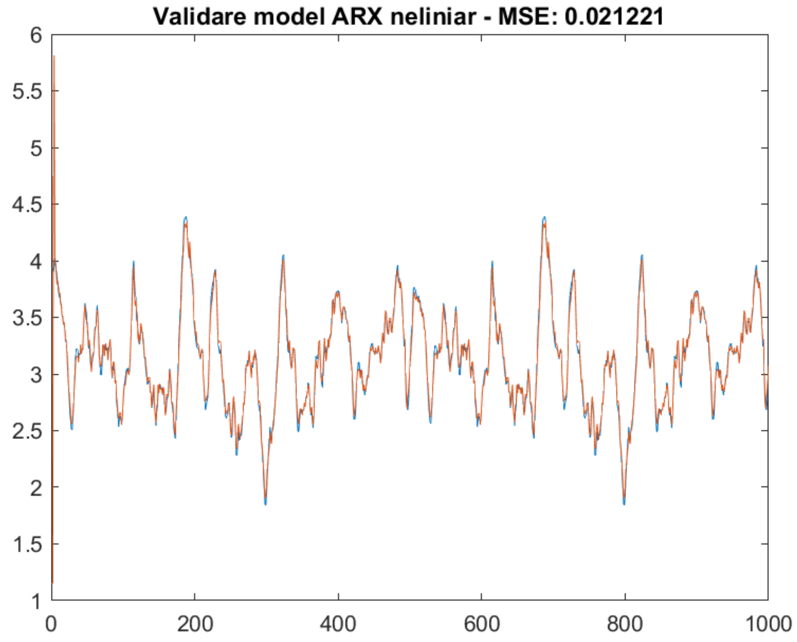
O altă opțiune este de a simula este de a folosi valoarea simulată precedent atunci când nu se pot folosi decât ieșirile anterioare ale sistemului.

În final se afișează graficele pentru identificare și validare evidețiindu-se erorile medii pătratice (MSE).

Grafice reprezentative

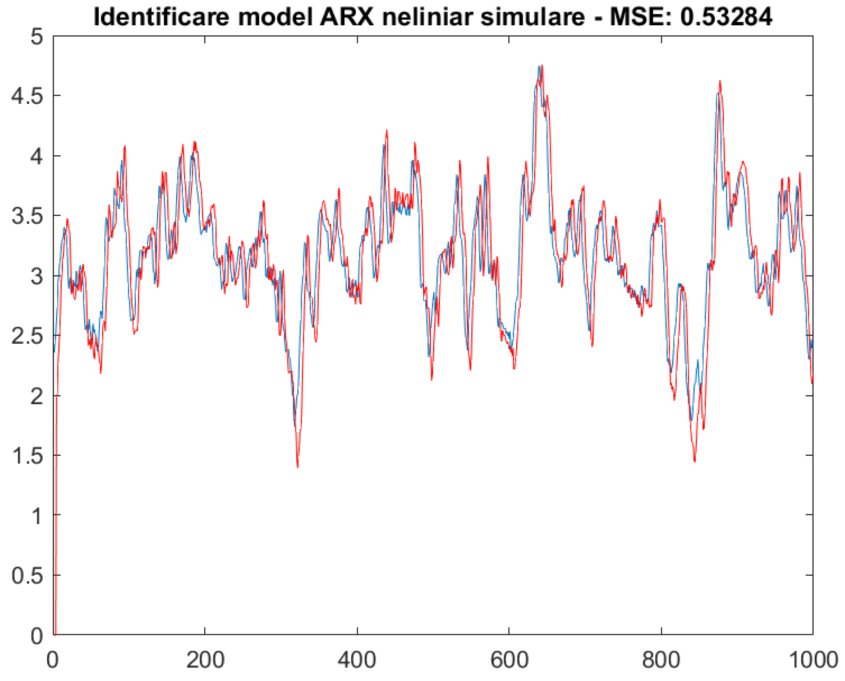
- Prima dată vom prezenta graficele pentru prezicerea unui sistem ARX neliniar cu regresie.

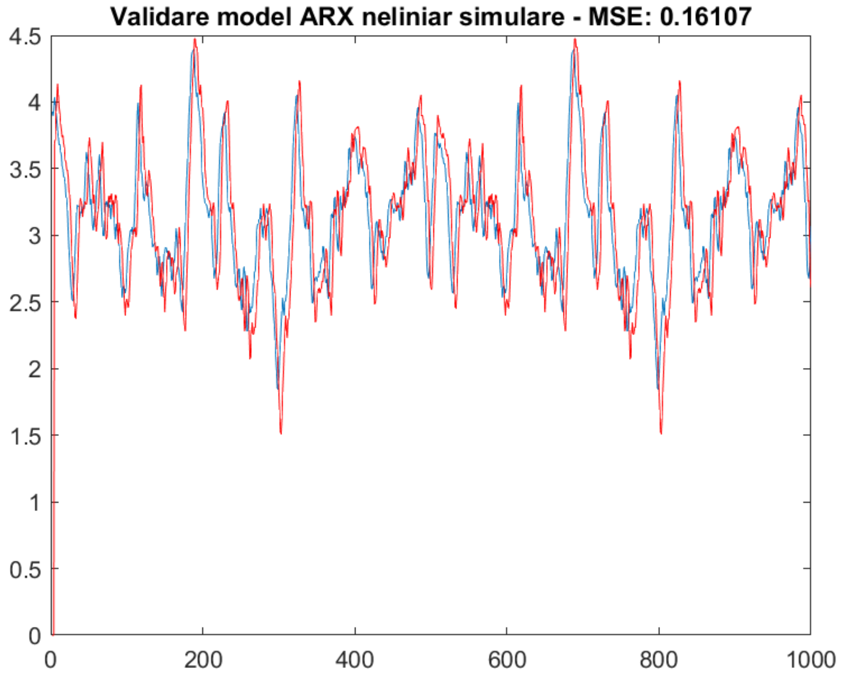




După cum se observă metoda noastră de prezicere este una buna având erori medii pătratice foarte mici, modelul fiind destul de bun.

Aceste grafice reprezintă a doua metodă ARX prin simulare





Metoda prin simulare aduce o evaluare a performanțelor destul de aproape de modelul dorit.

Concluzii

În concluzie, codul implementează o metodă de identificare a unui sistem ARX neliniar cu regresie polinomială de gradul n, evaluată prin eroarea medie pătratică (MSE). Evaluarea performanțelor se face atât prin predicție cât și prin simulare.

Sugestie de imbunătațire a codului:

->reglarea lui na,nb si n pentru rezultate mai bune

->folosirea unor funcții si operații specifice pentru eficiența codului.

```
% Încarcarea datelor
load("iddata-18.mat");
% Grafice initiale
figure;
plot(id)
title("Identificarea initiala")
figure;
plot(val)
title("Validarea initiala")
%na-ordinul polinomului ARX pentru ieșirea y
%nb-ordinul polinomului ARX pentru intrarea u
na = 3;
nb = 3;
% Date initiale
u_id = id.u;
y_id = id.y;
u_val = val.u;
y_val = val.y;
N = length(u_id);
```

```
% Gradul polinomului neliniar
n = 4;
% Construirea matricei de regresie
d = [];
for i = 1:N
% Prima coloană întotdeauna 1
d(i,1) = 1;
% Se calculeaza termenii liniari ARX
for j=2:na+1
if i-j>0
d(i,j)=-y_id(i-j);
else
d(i,j)=0;
end
end
% Se calculeaza partea 2 a matricii cu termeni ARX
for j=na+2:na+nb+1
if i-(j-na)>0
d(i,j)=u_id(i-(j-na));
else
d(i,j)=0;
end
end
```

```
% Termeni polinomiali neliniari pentru datele de identificare
index=na+nb+2;
for m=2:n % m putere
for w=0:m % w putere
if i-w>0&&i-(j-na)>0
parte_liniara = d(i,j);
parte_neliniara = d(i-w,j-na);
d(i, index) = -parte_neliniara^(m - w) * parte_liniara^w;
% Se trece la urmatoarea coloană
index=index+1;
else
d(i,index)=0;
index=index+1;
end
end
end
end
%predictia se face la fel pt validare
Nval=length(u_val);
d_val=[];
for i=1:Nval
% Prima coloană întotdeauna 1
d_val(i,1)=1;
% Se calculeaza termenii liniari ARX
for j=2:na+1
if i-j>0
d_val(i,j)=-y_val(i-j);
else
d_val(i,j) = 0;
end
end
% Se calculeaza partea 2 a matricii cu termeni ARX
for j =na+2:na+nb+1
if i-(j-na)>0
d_val(i, j)=u_val(i-(j-na));
else
d_val(i,j)=0;
end
end
% Termeni polinomiali neliniari pentru datele de validare
index = na + nb + 2;
for m = 2:n % m putere
for w = 0:m % w putere
if i-w>0 && i-(j-na) > 0
parte_liniara = d_val(i,j);
parte_neliniara = d_val(i-w,j-na);
d_val(i,index) = -parte_neliniara^(m-w)*parte_liniara^w;
% Se trece la urmatoarea coloană
index = index+1;
```

```
else
d_val(i,index)=0;
index = index+1;
end
end
end
end
% Calculul parametrilor theta
theta=d\y_id;
% Calculul iesirii modelului identificat
y_id1=d*theta;
y_hat=d_val*theta;
% Validare prin simulare
ysim=zeros(Nval,1);
for i=2:Nval
y1=0; % Suma pentru toate y-urile
y2=0; % Suma pentru toate u-urile
y3=0; % Suma pentru toate y-urile si u-urile neliniare
% Termeni liniari ARX pentru y
for j=1:na
if i-j>0
y1=y1-theta(j)*ysim(i-j);
end
end
% Termeni liniari ARX pentru u
for j=1:nb
if i-j>0
y2=y2+theta(na+j)*u_val(i-j);
end
end
% Termeni neliniari ARX
index=na+nb+2;
for m =2:n % m putere
for w=0:m % w putere
if i-w>0 && i-(na+1) > 0
parte_liniara = d_val(i,j);
parte_neliniara = d_val(i-w, na+1);
y3 = y3-theta(index)*parte_neliniara^(m - w)*parte_liniara^w;
index = index + 1;
end
end
end
%ysim(i) = y1 + y2 + y3;
ysim(i)=y3;
end
```

```
% Identificare prin simulare
ysimid=zeros(N, 1);
for i = 2:N
y1 = 0; % Suma pentru toate y-urile
y2 = 0; % Suma pentru toate u-urile
y3 = 0; % Suma pentru toate y-urile si u-urile neliniare
% Termeni liniari ARX pentru y
for j = 1:na
if i-j > 0
y1=y1-theta(j)*ysimid(i-j);
end
end
% Termeni liniari ARX pentru u
for j=1:nb
if i-j>0
y2=y2+theta(na + j)*u_id(i - j);
end
end
% Termeni neliniari ARX
index=na+nb+2;
for m = 2:n % m putere
for w = 0:m % w putere
if i-w > 0 && i-(na+1) > 0
parte_liniara = d(i,j);
parte_neliniara=d(i-w,na+1);
y3 = y3 - theta(index) * parte_neliniara^(m - w) *
parte_liniara^w;
index = index + 1;
end
end
end
%ysim(i) = y1 + y2 + y3;
ysimid(i)=y3;
end
% Eroare identificare
e1=y_id-y_id1;
MSE1 = 1/N*sum(e1.^2);
% Eroare validare prezicere
e2=y_val-y_hat;
MSE2 = 1/Nval*sum(e2.^2);
% Eroare validare simulare
e3=y_val-ysim;
MSE3=1/Nval*sum(e3.^2);
% Eroare identificare simulare
e4=y_val-ysimid;
```

```
MSE4=1/N*sum(e4.^2);
% Afisarea graficelor cu MSE
figure;
plot(y_id);
hold on;
plot(y_id1);
title(['Identificare model ARX neliniar - MSE: ',
num2str(MSE1)]);
figure;
plot(y_val);
hold on;
plot(y_hat);
title(['Validare model ARX neliniar - MSE: ', num2str(MSE2)]);
figure;
plot(y_val);
hold on;
plot(ysim,'r');
title(['Validare model ARX neliniar simulare - MSE: ',
num2str(MSE3)]);
figure;
plot(y_id);
hold on;
plot(ysimid,'r');
title(['Identificare model ARX neliniar simulare - MSE: ',
num2str(MSE4)]);
```