

ARX neliniar

Nume studenți:

Croitoriu Andreea-Beatrice
Demean Vlad-Ionuț
Filimon Antonia-Maria

Cuprins

1. Descrierea proiectului
 2. Structura modelului ARX neliniar
 3. Procedura de găsire a parametrilor
 4. Rezultate de reglare
 5. Interpretarea rezultatelor
 6. Concluzii
 7. Anexa cu listingul codului dezvoltat
-

Descrierea proiectului

Scopul proiectului este dezvoltarea unui model parametric tip cutie neagră, folosind o structură ARX (model Auto-Regresiv cu intrare eXogenă) neliniară de tip polinomial.

Structura modelului ARX neliniar

- $\hat{y}(k) = p(y(k-1), \dots, y(k-na), u(k-nk), u(k-nk-1), \dots, u(k-nk-nb+1)) = p(d(k))$
 - ~ $d(k)$ - vectorul de ieșiri și intrări întârziate;
 - ~ na - numărul coeficienților modelului auto-regresiv;
 - ~ nb - numărul coeficienților corespunzători variabilelor exogene;
 - ~ nk - întârzierea modelului;
 - p - un polinom de grad variabil m ;
 - u - intrarea sistemului;
 - y - ieșirea sistemului.
-

Procedura de găsire a parametrilor

- Deoarece modelul ARX neliniar conține pătrate și produse între variabilele întârziate, implementăm o funcție puteri care ne va genera un vector de puteri pe care îl vom folosi pentru elementele vectorului $d(k)$;
 - Știind că modelul ARX are proprietatea esențială de a fi liniar în parametrii, înseamnă că parametrii pot fi găsiți prin intermediul regresiei liniare;
 - Vom utiliza procedura de regresie prin găsirea vectorului de parametrii θ cu formula **$\theta = \phi_{id} \backslash y_{id}$** ;
 - Modelul cu intrări noi îl vom utiliza în două moduri:
 - ~ predicție(pas înainte): `yid_pred`;
 - ~ simulare(în care ieșirile precedente ale modelului nu sunt disponibile): `yid_sim`.
-

Rezultate de reglare

m	na	nb	MSEid_pred	MSEid_sim	MSEval_pred	MSEval_sim
1	1	1	0.3285	1.5968	0.2772	1.5434
1	1	2	0.3280	1.5979	0.2761	1.5522
1	2	1	0.3247	1.5859	0.2645	1.5219
2	1	1	0.2345	NaN	0.1994	NaN
2	1	2	0.2071	NaN	0.1948	NaN
2	2	1	0.2309	NaN	0.1961	NaN
3	1	1	0.1382	0.4772	0.1205	0.4972
3	1	2	0.1144	0.3612	0.1288	0.5899
3	2	1	0.1115	NaN	0.1938	1.0625

Interpretarea rezultatelor

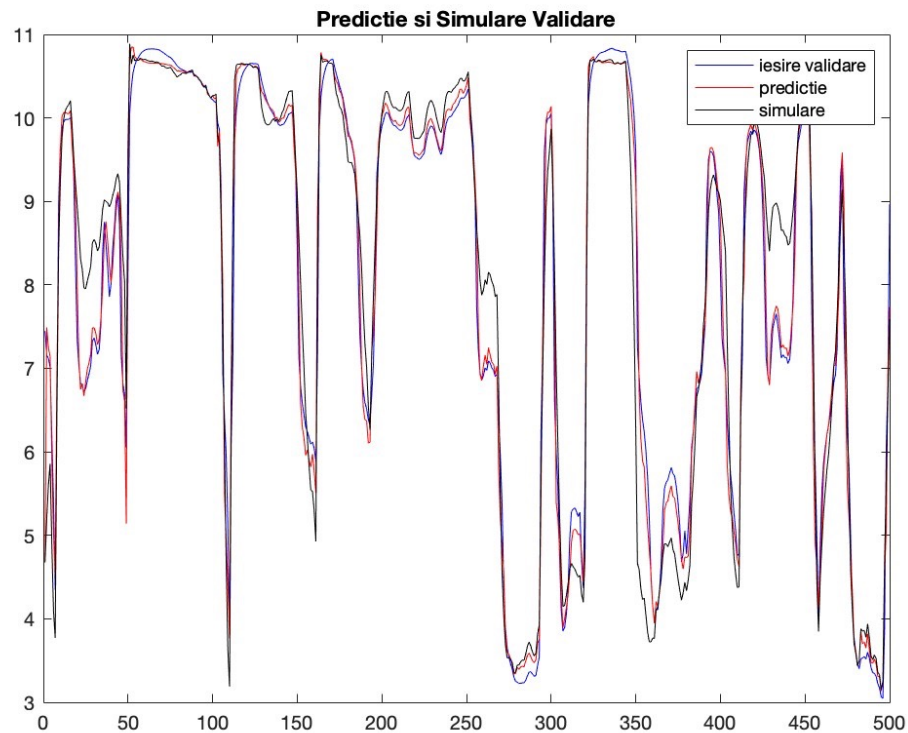


Fig.1 Predictie și Simulare Validare

Interpretarea rezultatelor

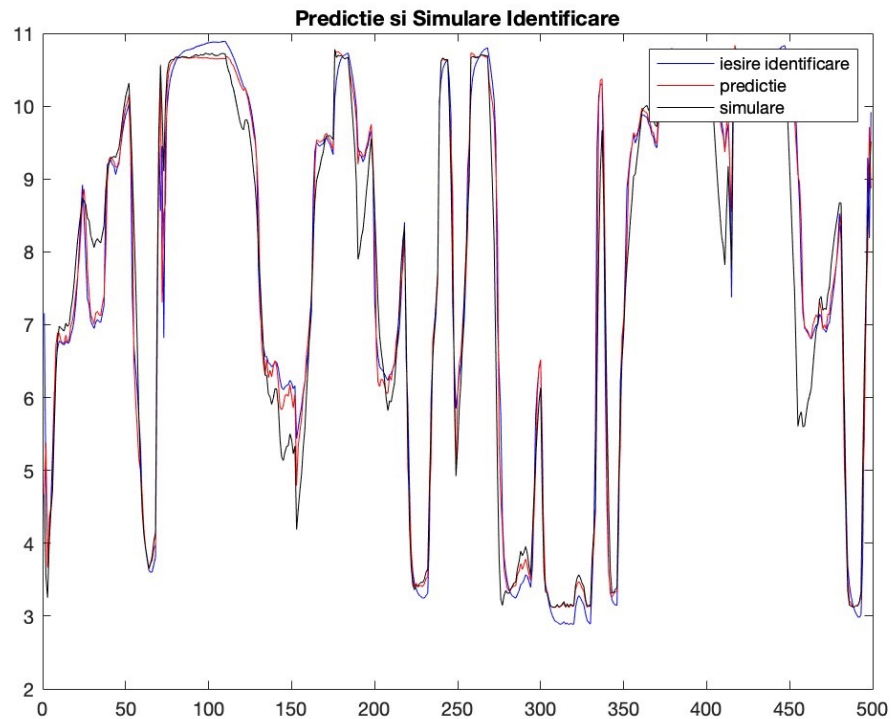


Fig.2 Predictie și Simulare Identificare

Concluzii

În concluzie, în urma generării graficelor, am observat comportamentul modelului în cazul ordinelor n_a, n_b și gradul m optime (în cazul nostru, $n_a = n_b = n_k = 1$ și $m = 3$).

De asemenea, pentru un m mai mare decât 4 apare limitarea reglării ARX-ului neliniar.

Anexa cu listiingul codului dezvoltat

```

clear all
close all
clc

load('iddata-01.mat')

uid = id.u;
yid = id.y;

uval = val.u;
yval = val.y;

%subplot(2, 1, 1)
%plot(uid)
%subplot(2, 1, 2)
%plot(yid)
%figure
%subplot(2, 1, 1)
%plot(uval)
%subplot(2, 1, 2)
%plot(yval)
%figure

na = 1;
nb = 1;
nk = 1;
m = 3;

vp = puteri(na, nb, m);

%predictie identificare

d = zeros;
phi_id = zeros;

```

```

for k = 1 : length(uid)
    for i = 1 : na
        if k - i <= 0
            d(k, i) = 0;
        else
            d(k, i) = -yid(k - i);
        end
    end

    for j = 1 : nb
        if k - j - nk + 1 <= 0
            d(k, na + j) = 0;
        else
            d(k, na + j) = uid(k - j - nk + 1);
        end
    end

    for p = 1 : length(vp)
        phi_id(k, p) = 1;
        for j = 1 : na + nb
            phi_id(k, p) = phi_id(k, p) * (d(k, j) ^ vp(p, j));
        end
    end
end

theta = phi_id \ yid;

yid_pred = phi_id * theta;

s = 0;
for i = 1 : length(yid)
    s = s + (yid(i) - yid_pred(i))^2;
end
MSEid_pred = 1/length(yid) * s;

```

```
%simulare identificare
```

```
xs_id = zeros;  
yid_sim = zeros(length(uid), 1);  
phis_id = zeros;  
  
for k = 1 : length(uid)  
    for i = 1 : na  
        if k - i <= 0  
            xs_id(k, i) = 0;  
        else  
            xs_id(k, i) = -yid_sim(k - i);  
        end  
    end  
  
    for j = 1 : nb  
        if k - j - nk + 1 <= 0  
            xs_id(k, na + j) = 0;  
        else  
            xs_id(k, na + j) = uid(k - j - nk + 1);  
        end  
    end  
  
    for p = 1 : length(vp)  
        phis_id(k, p) = 1;  
        for j = 1 : na + nb  
            phis_id(k, p) = phis_id(k, p) *  
                (xs_id(k, j) ^ vp(p, j));  
        end  
    end  
    yid_sim(k) = phis_id(k,:) * theta;  
end
```

```
s = 0;
```

```
for i = 1 : length(yid)  
    s = s + (yid(i) - yid_sim(i))^2;  
end  
MSEid_sim = 1/length(yid) * s;
```

```
plot(yid, 'b')  
hold on  
plot(yid_pred, 'r')  
hold on  
plot(yid_sim, 'black')  
legend('iesire identificare','predictie','simulare')  
title('Predictie si Simulare Identificare')
```

```
%predictie valida
```

```
d = zeros;  
phi_val = zeros;
```

```
for k = 1 : length(uval)  
    for i = 1 : na  
        if k - i <= 0  
            d(k, i) = 0;  
        else  
            d(k, i) = -yval(k - i);  
        end  
    end  
  
    for j = 1 : nb  
        if k - j - nk + 1 <= 0  
            d(k, na + j) = 0;  
        else  
            d(k, na + j) = uval(k - j - nk + 1);  
        end  
    end  
end
```

```

for p = 1 : length(vp)
    phi_val(k, p) = 1;
    for j = 1 : na + nb
        phi_val(k, p) = phi_val(k, p) * (d(k,
j) ^ vp(p, j));
    end
end

yval_pred = phi_val * theta;

s = 0;
for i = 1 : length(yval)
    s = s + (yval(i) - yval_pred(i))^2;
end
MSEval_pred = 1/length(yval) * s;

%simulare validare
xs_val = zeros;
yval_sim = zeros(length(uval), 1);
phis_val = zeros;

for k = 1 : length(uval)
    for i = 1 : na
        if k - i <= 0
            xs_val(k, i) = 0;
        else
            xs_val(k, i) = -yval_sim(k - i);
        end
    end
end

```

```

for j = 1 : nb
    if k - j - nk + 1 <= 0
        xs_val(k, na + j) = 0;
    else
        xs_val(k, na + j) = uval(k - j - nk + 1);
    end
end

for p = 1 : length(vp)
    phis_val(k, p) = 1;
    for j = 1 : na + nb
        phis_val(k, p) = phis_val(k, p) * (xs_val(k, j) ^ vp(p, j));
    end
end
yval_sim(k) = phis_val(k,:) * theta;

end

s = 0;
for i = 1 : length(yval)
    s = s + (yval(i) - yval_sim(i))^2;
end

MSEval_sim = 1/length(yval) * s;

figure
plot(yval, 'b')
hold on
plot(yval_pred, 'r')
hold on
plot(yval_sim, 'black')
legend('iesire validare','predictie','simulare')
title('Predictie si Simulare Validare')

```

```
function vp = puteri(na, nb, m)
    vp = zeros;

    for i = 1 : m + 1
        vp(i) = i - 1;
    end

    vp = unique(nchoosek(repmat(vp, 1, na + nb), na + nb), 'rows');

    k = 1;
    while(k <= length(vp))
        if(sum(vp(k,:)) > m)
            vp(k,:) = [];
            k = k - 1;
        end
        k = k + 1;
    end
end
```
