Generarea unui model ARX neliniar de tip polinomial

Alina Claudia Precup   Răzvan-Vasile Vanca

Universitatea Tehnică din Cluj-Napoca, Facultatea de Automatică și Calculatoare,
Specializarea Automatică și Informatică Aplicată, Grupa 30134
data-index: 01

Conținut

Conținut

Elementele Problemei

- Un set de date $u(k)$, $y(k)$, $k = 1, \ldots, N$ măsurate de la un sistem dinamic cu o intrare și o ieșire.
- Există o întârziere *nk* între intrări și ieșiri, utilă în modelarea sistemelor cu timp mort.
- Vectorul de regresori $\varphi \in \mathbb{R}^{nm}$, combinații ale ieșirilor și intrărilor.
- Vectorul de parametri $\theta \in \mathbb{R}^{nm}$ necunoscut.
- Antrenarea modelului prin metoda regresiei liniare.

Conținut

Obiectivul

- Dezvoltarea unui model de tip cutie neagră pentru sistemul dinamic, folosind o structură ARX neliniară de tip polinomial.
- Programarea procedurii de regresie pentru identificarea parametrilor (coeficienții reali).
- Utilizarea modelului cu intrări noi. Această utilizare fiind făcută în două moduri: predicție și simulare.
- Validarea modelului obținut pe setul de validare.
- Alegerea ordinului modelului, întârzierea și gradul polinomului în vederea obținerii unei aproximări optime.

Conținut

1. Motivație
   - Elementele Problemei
   - Obiectivul
   - Datele de Intrare

2. Particularități ale Soluției Alese
   - Modelul Matematic
   - Modelul de Aproximare
   - Alegerea Gradului Polinomului și a Ordinului Sistemului

Setul de antrenare a modelului

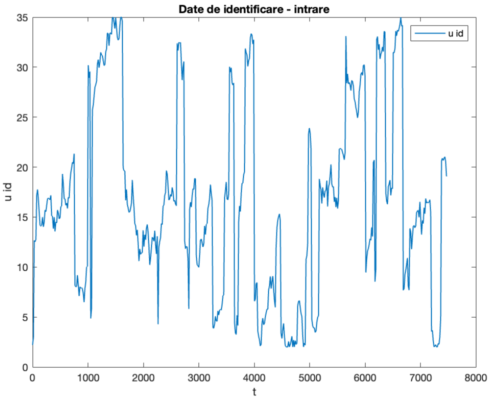

Figure 1: Reprezentarea datelor de identificare (intrarea sistemului)

Setul de antrenare a modelului

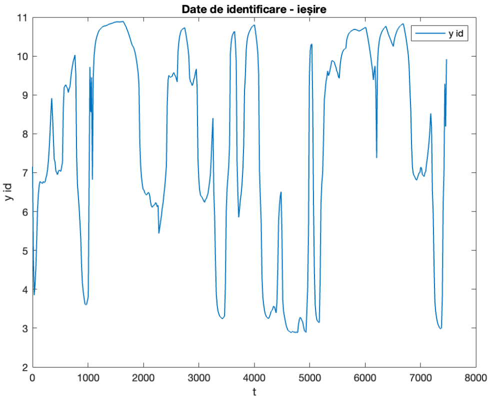

Figure 2: Reprezentarea datelor de identificare (ieșirea sistemului)

Setul de validare a modelului

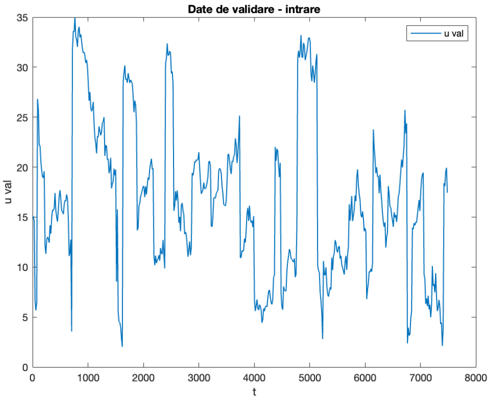

Figure 3: Reprezentarea datelor de validare (intrarea sistemului)

Setul de validare a modelului

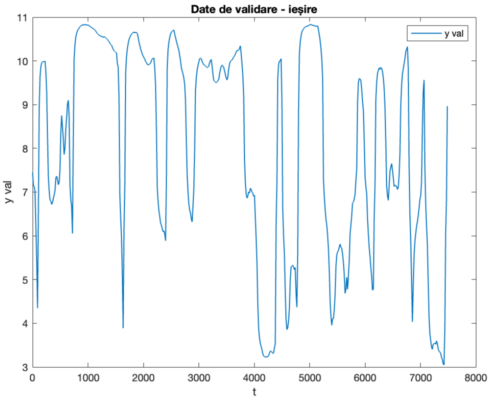

Figure 4: Reprezentarea datelor de validare (ieșirea sistemului)

Conținut

Modelul ARX neliniar

Considerăm un model cu ordinele *na*, *nb* și întârzierea *nk*. Modelul ARX neliniar are atunci structura:

$$\begin{aligned}\hat{y}(k) &= p(y(k-1), \dots, y(k-na), u(k-nk), u(k-nk-1), \dots,\\ &\quad u(k-nk-nb+1); \theta)\\ &= p(d(k))\end{aligned}$$

unde vectorul de semnale precedente (ieșiri și intrări întârziate) este:

$$\begin{aligned}d(k) = [&y(k-1), \dots, y(k-na), u(k-nk), u(k-nk-1), \dots,\\ &u(k-nk-nb+1))]^T\end{aligned}$$

Funcția *p* este parametrizată de $\theta \in \mathbb{R}^n$, iar acești parametri pot fi aleși pentru a recupera datele de identificare, identificând astfel un sistem dinamic neliniar.

Explicitarea Polinomului

În cazul nostru particular, p este un polinom de gradul m în ieșirile și intrările precedente.
De exemplu, pentru câteva valori particulare ale lui na, nb, nk, vectorul d și polinomul p au forma:

$$
\begin{aligned}
na = nb &= 1, m = 2, nk = 0 \\
d &= [y(k-1), u(k-1)]^T \\
\hat{y}(k) &= [1, u(k-1), u(k-1)^2, y(k-1), y(k-1)u(k-1), y(k-1)^2]^T \\
&\quad \cdot \theta \\
&= \theta_1 + \theta_2 u(k-1) + \theta_3 u(k-1)^2 + \theta_4 y(k-1) + \\
&\quad + \theta_5 y(k-1)u(k-1) + \theta_6 y(k-1)^2
\end{aligned}
$$

unde θ_1, θ_2, θ_3, θ_4, θ_5, θ_6 sunt coeficienții reali și parametrii modelului.

Modelul neliniar

Modelul este neliniar, conține produse între variabile întârziate, nu doar termeni liniari în $y(k-1)$ și $u(k-1)$ ca și în cazul ARX-ului standard.

Modelul are deci proprietatea esențială că este liniar în parametri, motiv pentru care aceștia pot fi găsiți folosind metoda regresiei liniare.

Forma Generală a Sistemului Liniar

Pentru aproximatorul polinomial $\hat{y}$ de grad m în $na + nb$ variabile numărul de monoame nm este dat de formula combinatorică:

$$nm = \binom{m + na + nb}{na + nb}$$

unde na, nb reprezintă ordinul sistemului, iar m gradul polinomului de aproximare (na, nb, m fiind configurabile)

Pentru simplitate, vom fixa nk=1 și vom considera na=nb în căutarea gradului și a ordinului optim.

Sistemul de Ecuații

Scriind modelul pentru datele de intrare, obținem următorul sistem de ecuații liniare:

y(1)=$\varphi_1(1)\theta_1 + \varphi_2(1)\theta_2 + \cdots + \varphi_{nm}(1)\theta_{nm}$
y(2)=$\varphi_1(2)\theta_1 + \varphi_2(2)\theta_2 + \cdots + \varphi_{nm}(2)\theta_{nm}$
$\ldots$
y(n)=$\varphi_1(n)\theta_1 + \varphi_2(n)\theta_2 + \cdots + \varphi_{nm}(n)\theta_{nm}$

n$-$numărul datelor de intrare

$\varphi_1, \varphi_2, \ldots, \varphi_{nm}$ $-$ regresorii (produse între variabile întârziate)

$\theta_1, \theta_2, \ldots, \theta_{nm}$ $-$ parametrii modelului (coeficienții reali)

Observație: Când $k < na, nb$, valorile lui u și y la momentele de timp negative și 0, necesare pentru construirea lui φ vor fi luate 0 (presupunând condiții inițiale nule).

Forma Matriceală a Sistemului de Ecuații

Sistemul se poate scrie sub formă matriceală astfel:

$$\begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(n) \end{bmatrix} = \begin{bmatrix} \varphi_1(1) & \varphi_2(1) & \cdots & \varphi_{nm}(1) \\ \varphi_1(2) & \varphi_2(2) & \cdots & \varphi_{nm}(2) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_1(n) & \varphi_2(n) & \cdots & \varphi_{nm}(n) \end{bmatrix} \cdot \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_{nm} \end{bmatrix}$$

$$Y = \Phi\theta$$

$$\hat{\theta} = (\Phi^T\Phi)^{-1}\Phi^T Y$$

noile variabile fiind: $Y \in \mathbb{R}^n$ și $\Phi \in \mathbb{R}^{n\times nm}$.

- Problema se reduce la a găsi vectorul de parametri care minimizează media erorilor pătratice.
- Vectorul $\hat{\theta}$ este deci soluția reală a problemei de optimizare, dar rămâne totuși o estimare datorită zgomotului din datele de intrare.

Modelul Matematic

Pentru simplitate, în explicitarea modelului matematic vom considera vectorul de semnale precedente ca fiind:

$$[x_1, x_2, \ldots, x_{nv}]$$

iar *nk* îl vom considera 0.
nv - numărul de variabile ale polinomului ($nv = na + nb$)

Astfel, au loc schimbările de variabile:

$$\begin{aligned} y(k-1) &\leftrightarrow x_1 \\ y(k-2) &\leftrightarrow x_2 \\ &\vdots \\ y(k-na) &\leftrightarrow x_{na} \end{aligned}$$

$$\begin{aligned} u(k-1) &\leftrightarrow x_{na+1} \\ u(k-2) &\leftrightarrow x_{na+2} \\ &\vdots \\ u(k-nb) &\leftrightarrow x_{nv} \end{aligned}$$

Generarea Monoamelor Polinomului de grad m în $na + nb$ variabile

Presupunem că $m = 2$, iar $na = nb = 2$. Atunci, vectorul de variabile (în cazul nostru semnale precedente) arată astfel:

$$[x_1, x_2, x_3, x_4] \leftrightarrow [y(k-1), y(k-2), u(k-1), u(k-2)]$$

Regresorii fiind următorii:

$$\begin{array}{ccccccc}
0 & 0 & 0 & 0 & \rightarrow & 1 \\
0 & 0 & 0 & 1 & \rightarrow & x_4 \\
0 & 0 & 0 & 2 & \rightarrow & x_4^2 \\
0 & 0 & 1 & 0 & \rightarrow & x_3 \\
0 & 0 & 1 & 1 & \rightarrow & x_3 x_4 \\
0 & 0 & 2 & 0 & \rightarrow & x_3^2 \\
0 & 1 & 0 & 0 & \rightarrow & x_2 \\
0 & 1 & 0 & 1 & \rightarrow & x_2 x_4
\end{array}
\qquad
\begin{array}{ccccccc}
0 & 1 & 1 & 0 & \rightarrow & x_2 x_3 \\
0 & 2 & 0 & 0 & \rightarrow & x_2^2 \\
1 & 0 & 0 & 0 & \rightarrow & x_1 \\
1 & 0 & 0 & 1 & \rightarrow & x_1 x_4 \\
1 & 0 & 1 & 0 & \rightarrow & x_1 x_3 \\
1 & 1 & 0 & 0 & \rightarrow & x_1 x_2 \\
2 & 0 & 0 & 0 & \rightarrow & x_1^2
\end{array}$$

Explicitarea Modelului Matematic

$$
\begin{bmatrix} \hat{y}(1) \\ \hat{y}(2) \\ \vdots \\ \hat{y}(n) \end{bmatrix} =
\begin{bmatrix}
1 & x_{nv}(1) & \cdots & x_{nv}(1)^m & x_{nv-1}(1) & x_{nv-1}(1)x_{nv}(1) & \cdots \\
1 & x_{nv}(2) & \cdots & x_{nv}(2)^m & x_{nv-1}(2) & x_{nv-1}(2)x_{nv}(2) & \cdots \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots \\
1 & x_{nv}(n) & \cdots & x_{nv}(n)^m & x_{nv-1}(n) & x_{nv-1}(n)x_{nv}(n) & \cdots
\end{bmatrix.}
$$

$$
\begin{matrix}
x_{nv-1}(1)x_{nv}(1)^{m-1} & x_{nv-1}(1)^2 & \cdots & x_{nv-1}(1)^2 x_{nv}(1)^{m-2} & \cdots & x_{nv-1}(1)^m \\
x_{nv-1}(2)x_{nv}(2)^{m-1} & x_{nv-1}(2)^2 & \cdots & x_{nv-1}(2)^2 x_{nv}(2)^{m-2} & \cdots & x_{nv-1}(2)^m \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
x_{nv-1}(n)x_{nv}(n)^{m-1} & x_{nv-1}(n)^2 & \cdots & x_{nv-1}(n)^2 x_{nv}(n)^{m-2} & \cdots & x_{nv-1}(n)^m
\end{matrix}
$$

$$
\left.\begin{matrix}
\cdots & x_1(1) & x_1(1)x_{nv}(1) & \cdots & x_1 x_{nv}(1)^{m-1} & \cdots & x_1(1)^m \\
\cdots & x_1(2) & x_1(2)x_{nv}(2) & \cdots & x_1 x_{nv}(2)^{m-1} & \cdots & x_1(2)^m \\
\ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
\cdots & x_1(n) & x_1(n)x_{nv}(n) & \cdots & x_1 x_{nv}(n)^{m-1} & \cdots & x_1(n)^m
\end{matrix}\right] \cdot
\begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_{nm} \end{bmatrix}
$$

Ilustrarea Modelului Matematic

Presupunem că gradul polinomului este m=2, iar ordinul na=nb=1, deci vor exista nm=6 monoame în componența acestuia.

$$\begin{bmatrix} \hat{y}(1) \\ \hat{y}(2) \\ \vdots \\ \hat{y}(n) \end{bmatrix} = \begin{bmatrix} 1 & x_2(1) & x_2(1)^2 & x_1(1) & x_1(1)x_2(1) & x_1(1)^2 \\ 1 & x_2(2) & x_2(2)^2 & x_1(2) & x_1(2)x_2(1) & x_1(2)^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_2(n) & x_2(n)^2 & x_1(n) & x_1(n)x_2(n) & x_1(n)^2 \end{bmatrix} \cdot \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_6 \end{bmatrix}$$

Concretizarea Modelului Matematic

$$\begin{bmatrix} \hat{y}(1) \\ \hat{y}(2) \\ \vdots \\ \hat{y}(n) \end{bmatrix} = \begin{bmatrix} 1 & u(1\text{-}1) & u(1\text{-}1)^2 & y(1\text{-}1) & y(1\text{-}1)u(1\text{-}1) & y(1\text{-}1)^2 \\ 1 & u(2\text{-}1) & u(2\text{-}1)^2 & y(2\text{-}1) & y(2\text{-}1)u(1\text{-}1) & y(2\text{-}1)^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & u(n\text{-}1) & u(n\text{-}1)^2 & y(n\text{-}1) & y(n\text{-}1)u(n\text{-}1) & y(n\text{-}1)^2 \end{bmatrix} \cdot \theta$$

$$\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_6 \end{bmatrix}$$

Utilizarea Modelului - Predicție

Ieșirea reală a sistemului este cunoscută, deci este disponibil vectorul de semnale întârziate $d(k)$:

$$d(k) = [y(k-1), \dots, y(k-na), u(k-nk), u(k-nk-1), \dots, \\ u(k-nk-nb+1))]^T$$

Predicție cu un pas înainte: $\hat{y}(k) = p(d(k); \hat{\theta})$

Observație: Semnalele la momente negative de timp vor fi luate 0.

Utilizarea Modelului - Simulare

Ieșirea reală a sistemului este necunoscută, deci vom folosi ieșirile simulate anterior pentru a construi o aproximare $\tilde{d}(k)$ a lui $d(k)$:

$$\tilde{d}(k) = [\tilde{y}(k-1), \ldots, \tilde{y}(k-na), u(k-nk), u(k-nk-1), \ldots, u(k-nk-nb+1))]^T$$

$$\text{Simulare :} \quad \tilde{y}(k) = p(\tilde{d}(k); \hat{\theta})$$

Observație: Ieșirile simulate la pași negativi și zero vor fi luate 0.

Conținut

Predicție: Aproximarea datelor de identificare pentru grad 2 și ordin 8

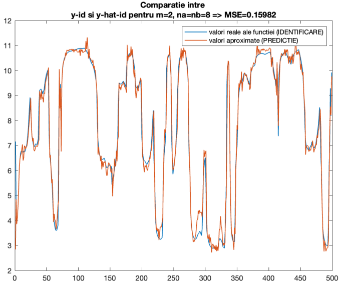

Figure 5: Comparația între y_{hat} și y pentru m=2, na=nb=8

Rezultat bun! Parametrii aleși duc la o aproximare bună a datelor de identificare.

Predicție: Aproximarea datelor de validare pentru grad 2 și ordin 8

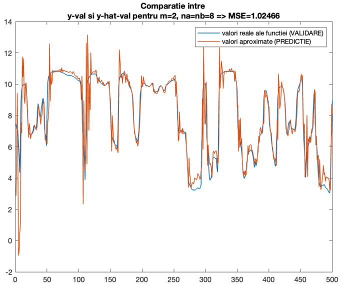

Figure 6: Comparația între y_{hat} și y pentru m=2, na=nb=8

Rezultat prost! Ordinul mare duce la fenomenul de supraantrenare.

Predicție: Aproximarea datelor de identificare pentru grad 4 și ordin 1

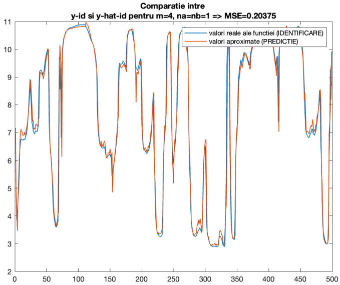

Figure 7: Comparația între y_{hat} și y pentru m=4, na=nb=1

Rezultat bun! Parametrii aleși duc la o aproximare bună a datelor de identificare.

Predicție: Aproximarea datelor de validare pentru grad 4 și ordin 1

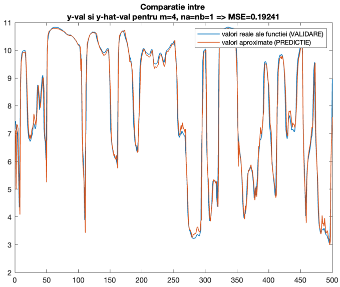

Figure 8: Comparația între y_{hat} și y pentru m=4, na=nb=1

Rezultat bun! Parametrii aleși duc la o aproximare optimă a datelor de validare.

Simulare: Aproximarea datelor de identificare pentru grad 3 și ordin 1

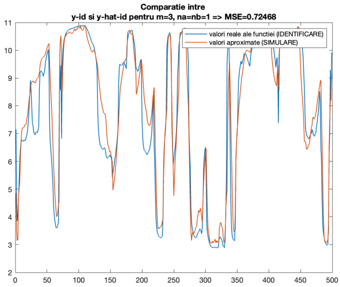

Figure 9: Comparația între y_{hat} și y pentru m=3, na=nb=1

Rezultat bun! Parametrii aleși duc la o aproximare optimă a datelor de identificare.

Simulare: Aproximarea datelor de validare pentru grad 3 și ordin 1

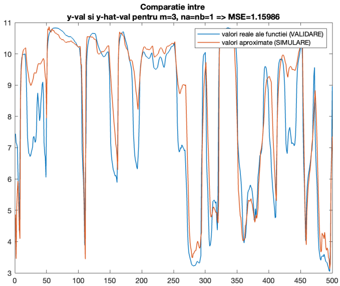

Figure 10: Comparația între y_{hat} și y pentru m=3, na=nb=1

Rezultat bun! Parametrii aleși duc la o aproximare optimă a datelor de validare.

Simulare: Aproximarea datelor de identificare pentru grad 1 și ordin 1

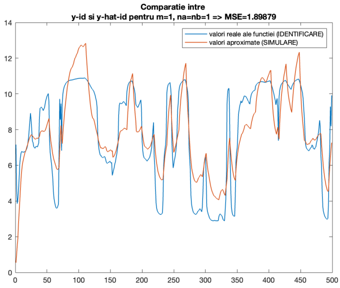

Figure 11: Comparația între y_{hat} și y pentru m=1, na=nb=1

Rezultat prost! Gradul și ordinul mic duc la fenomenul de subantrenare.

Simulare: Aproximarea datelor de validare pentru grad 1 și ordin 1

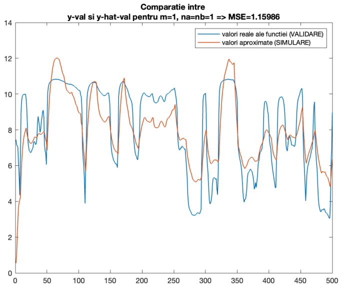

Figure 12: Comparația între y_{hat} și y pentru m=1, na=nb=1

Rezultat prost! Gradul și ordinul mic duc la fenomenul de subantrenare.

Cele mai bune modele obținute pentru aproximarea datelor de identificare

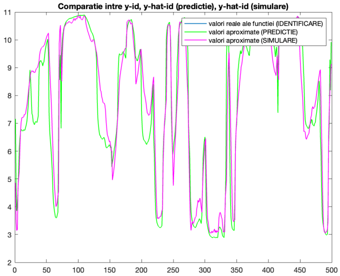

Figure 13: Comparația între y_{hat}(predicție), y_{hat}(simulare) și y

Utilizarea modelului cu intrări noi prin predicție duce la rezultate mai bune decât prin simulare.

Cele mai bune modele obținute pentru aproximarea datelor de validare

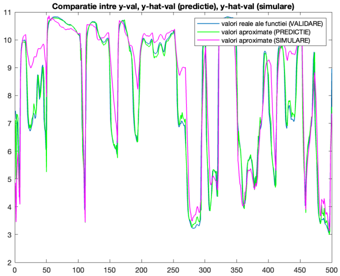

Figure 14: Comparația între y_{hat}(predicție), y_{hat}(simulare) și y

Utilizarea modelului cu intrări noi prin predicție duce la rezultate mai bune decât prin simulare.

Conținut

Predicție: Evoluția erorii medii pătratice în raport cu m

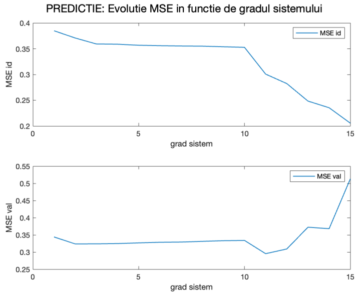

Figure 15: Reprezentarea erorii medie pătratice în funcție de m

Pentru $m > 10$ eroarea pe datele de validare începe să crească, iar pe datele de identificare să scadă.

Predicție: Evoluția erorii medii pătratice în raport cu na ($na = nb$)

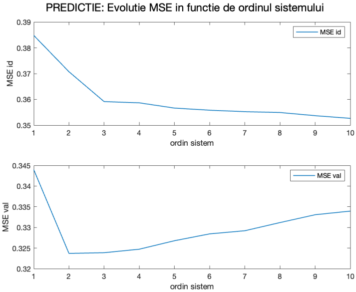

Figure 1: Reprezentarea erorii medie pătratice în funcție de na

Pentru $na > 5$ eroarea pe datele de validare începe să crească, iar pe datele de identificare să scadă.

Simulare: Evoluția erorii medii pătratice în raport cu m

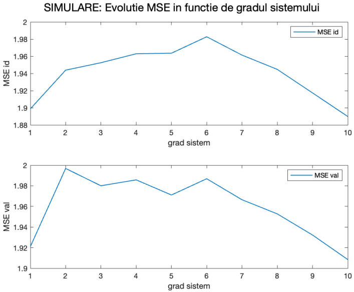

Figure 1: Reprezentarea erorii medie pătratice în funcție de m

Pentru m mare apar nedeterminări. Se constată că pentru m suficient de mic erorile sunt mici, atât pe validare, cât și pe identificare.

Simulare: Evoluția erorii medii pătratice în raport cu na ($na = nb$)

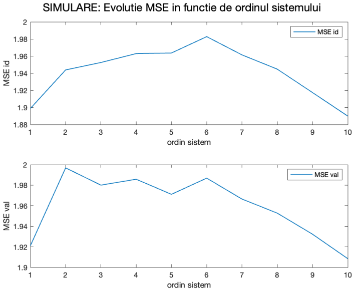

Figure 1: Reprezentarea erorii medie pătratice în funcție de na

Pentru na mare apar nedeterminări. Se constată că pentru na mic erorile sunt mici, atât pe validare, cât și pe identificare.

Concluzii

- Predicție: gradul optim al polinomului este 4, iar ordinul optim al sistemului este 1.
- Simulare: gradul optim al polinomului este 3, iar ordinul optim al sistemului este 1.
- Pentru un grad și ordin prea mici apare fenomenul de subantrenare la utilizarea modelului în modul simulare.
- Datele sunt afectate de zgomot, deci creșterea exagerată a gradului și ordinului va duce la supraantrenare: performanțe bune pe datele de identificare, dar proaste pe datele de validare la utlizarea modelului în modul predicție.

Bibliografie I

Arun K. Tangirala
Principles of System Identification.
CRC Press, 2015.

Lennart Ljung
System Identification Theory for the User.
Prentice Hall PTR, 1999.

Torsten Söderström, Petre Stoica
System Identification.
Prentice Hall, 2001.

Christopher M. Bishop
Pattern Recognition and Machine Learning.
Springer Science, 2006.

Bibliografie II

Ian Goodfellow, Yoshua Bengio and Aaron Courville
Deep Learning.
The MIT Press Cambridge, 2016.

Lucian Bușoniu
Identificarea sistemelor - Ingineria Sistemelor, anul 3
Universitatea Tehnica din Cluj-Napoca.
Partea V - Metoda ARX

```
% x - vector simbolic care reprezinta variabilele polinomului
% m - gradul polinomului
% current_powers - puterile curente ale fiecarei variabile
function monomials = generate_monomials(x, m, current_powers)
    if length(current_powers) == length(x)
        prod = 1;
        for i = 1:length(x)
            prod=prod*x(i)^current_powers(i);
        end

        monomials = prod;
    else
        monomials = [];
```

```
            % power-potentiala putere care inca nu a fost asignata variabilei
            % curente
        for power = 0:m - sum(current_powers)
            new_current_powers=[current_powers, power]; %concatenarea valorii power la setul curent de puteri

            new_result=generate_monomials(x, m, new_current_powers); %apel recursiv

            monomials = [monomials, new_result];
        end
    end
end
```

```
function [phi_row]=create_phi_row(m,na,nb,y, u
    , number_of_row)
        variables=[];
        k=number_of_row;

        for i=1:na
            if k-i>0
                variables(i)=y(k-i);
            else
                variables(i)=0;
            end
        end

        nk=1; %timpul mort

        for i=1:nb
```

```
            if k-nk-i>0
                variables(na+i)=u(k-nk-i);
            else
                variables(na+i)=0;
            end
        end

        current_powers=[];

        phi_row=generate_monomials(variables,
            m, current_powers);
end




.
```

```
function [phi]=create_phi_matrix(N,m,na,nb,y,
    u)
    for k=1:N
        variables=[];

        for i=1:na
            if k-i>0
                variables(i)=y(k-i); %iesirea
                    sistemului
            else
                variables(i)=0;
            end
        end

        nk=1; %timpul mort

```

```
        for i=1:nb
            if k-nk-i>0
                variables(na+i)=u(k-nk-i); % intrarea intarziata
            else
                variables(na+i)=0;
            end
        end

        current_powers=[];
        monomials=generate_monomials(variables, m, current_powers);

        phi(k,:)=monomials;
    end
end
```

```
function [y_tilda,e,MSE] = calculate_for_simulation(N, m, na, nb, theta, u, y)

y_tilda=zeros(N,1);

for k=1:N
    phi_row=create_phi_row(m,na,nb,y_tilda,u,k);
    y_tilda(k)=phi_row*theta(1:length(phi_row));
end

e=y_tilda-y(1:length(y_tilda));
MSE=mean(e.^2);
end
```

```
function [MSE_id, MSE_val, theta, phi_id, phi_val, e_id, e_val, y_hat_id, y_hat_val]=calculate_for_prediction(N_id, N_val,m,na,nb,u_id, y_id, u_val, y_val)
phi_id=create_phi_matrix(N_id, m, na, nb, y_id, u_id);
theta=phi_id\y_id;
y_hat_id=phi_id*theta;
e_id=y_id(1:length(y_hat_id))-y_hat_id;
MSE_id=mean(e_id.^2);
phi_val=create_phi_matrix(N_val, m, na, nb, y_val, u_val);
y_hat_val=phi_val*theta;
e_val=y_val(1:length(y_hat_val))-y_hat_val;
MSE_val=mean(e_val.^2);
end
```

```
clear all; close all; clc;
load('iddata-01.mat')

% Datele de identificare
u_id=id.u;
y_id=id.y;
t_id=id_array(:,1);

% Datele de validare
u_val=val.u;
y_val=val.y;
t_val=val_array(:,1);

Ts=t_id(2)-t_id(1); % perioada de esantionare
% Reprezentarea grafica a intrarii de identificare
```

```
figure
plot(t_id, u_id,'LineWidth',1)
xlabel("t")
ylabel("u id")
title("Date de identificare – intrare")
legend("u id")
% Reprezentarea grafica a iesirii de
    identificare
figure
plot(t_id, y_id,'LineWidth',1)
xlabel("t")
ylabel("y id")
title("Date de identificare – ie ire")
legend("y id")
```

```
% Reprezentarea grafica a intrarii de validare
figure
plot(t_val, u_val,'LineWidth',1)
xlabel("t")
ylabel("u val")
title("Date de validare - intrare")
legend("u val")

% Reprezentarea grafica a iesirii de validare
figure
plot(t_val,y_val,'LineWidth',1)
xlabel("t")
ylabel("y val")
title("Date de validare - ie ire")
legend("y val")
```

```
N_id=length(u_id); % nr datelor identificare
N_val=length(u_val); % nr date validare
grad_maxim=10; %gradul maxim al polinomului
ordin_maxim=5; %ordinul maxim al sistemului
poz=1;
for m=1:grad_maxim
    for na=1:ordin_maxim
        nb=na; % consideram na=nb pentru simplitate
        % predictie
        [MSE_pr_id(poz), MSE_pr_val(poz), theta, phi_id, phi_val, e_id, e_val, y_hat_id, y_hat_val]= calculate_for_prediction(N_id,N_val,m,na,nb,u_id, y_id, u_val, y_val);
        % simulare
```

```
            [y_tilda_val,e_val, MSE_sim_val(poz)] = calculate_for_simulation(N_val, m, na, nb, theta, u_val, y_val);
            [y_tilda_id,e_id, MSE_sim_id(poz)] = calculate_for_simulation(N_id, m, na, nb, theta, u_id, y_id);
            % actualizare variabile
            y_hat_pr_id(:,poz)=y_hat_id;
            y_hat_pr_val(:,poz)=y_hat_val;
            y_hat_sim_id(:,poz)=y_tilda_id;
            y_hat_sim_val(:,poz)=y_tilda_val;
            theta(:,poz)=theta';
            order_vector(poz)=na;
            poz=poz+1;
        end
end
```

```
%% ----------------PREDICTIE-----------------
%% MSE minima pe datele de identificare
[MSE10, index10]=min(MSE_pr_id);
sys_degree10=ceil(index10/ordin_maxim);
sys_order10=order_vector(index10);
%=> m=4, na=nb=5
%=> MSE pe id: 1.063903e-17
%=> MSE pe val: 1.647723e+07
%=> FENOMEN SUPRAANTRENARE
figure
plot(y_id,'LineWidth',1)
hold on
plot(y_hat_pr_id(:, index10),'LineWidth',1)
title(sprintf("Comparatie intre \ny-id si y-
    hat-id pentru m=%d, na=nb=%d => MSE=%d",
    sys_degree10, sys_order10, MSE10))
```

```
legend("valori reale ale functiei (IDENTIFICARE)", "valori aproximate (PREDICTIE)")
figure
plot(y_val,'LineWidth',1)
hold on
plot(y_hat_pr_val(:, index10),'LineWidth',1)
title(sprintf("Comparatie intre \ny-val si y-hat-val pentru m=%d, na=nb=%d => MSE=%d", sys_degree10, sys_order10, MSE_pr_val(index10)))
legend("valori reale ale functiei (VALIDARE)", "valori aproximate (PREDICTIE)")
%% MSE minima pe datele de validare
[MSE11, index11]=min(MSE_pr_val);
sys_degree11=ceil(index11/ordin_maxim);
```

```
sys_order11=order_vector(index11);
%=> MSE pe id: 0.20375; MSE pe val: 0.19241
figure
plot(y_id,'LineWidth',1)
hold on
plot(y_hat_pr_id(:, index11),'LineWidth',1)
title(sprintf("Comparatie intre \ny-id si y-hat-id pentru m=%d, na=nb=%d => MSE=%.5f", sys_degree11, sys_order11, MSE_pr_id(index11)))
legend("valori reale ale functiei (IDENTIFICARE)", "valori aproximate (PREDICTIE)")
figure
plot(y_val,'LineWidth',1)
hold on
```

```
plot(y_hat_pr_val(:, index11),'LineWidth',1)
title(sprintf("Comparatie intre \ny-val si y-hat-val pentru m=%d, na=nb=%d => MSE=%.5f", sys_degree11, sys_order11, MSE11))
legend("valori reale ale functiei (VALIDARE)", "valori aproximate (PREDICTIE)")
%% grad mic, ordin mare => FENOMEN SUPRAANTRENARE
index30=18;
sys_degree30=ceil(index30/ordin_maxim);
sys_order30=order_vector(index30);
figure
plot(y_id,'LineWidth',1)
hold on
plot(y_hat_pr_id(:, index30),'LineWidth',1)
title(sprintf("Comparatie intre \ny-id si y-
```

```
hat-id pentru m=%d, na=nb=%d => MSE=%.5f", sys_degree30, sys_order30, MSE_pr_id(index30)))
legend("valori reale ale functiei (IDENTIFICARE)", "valori aproximate (PREDICTIE)")
figure
plot(y_val,'LineWidth',1)
hold on
plot(y_hat_pr_val(:, index30),'LineWidth',1)
title(sprintf("Comparatie intre \ny-val si y-hat-val pentru m=%d, na=nb=%d => MSE=%.5f", sys_degree30, sys_order30, MSE_pr_val(index30)))
legend("valori reale ale functiei (VALIDARE)", "valori aproximate (PREDICTIE)")
```

```
%%  ----------------SIMULARE----------------
%% MSE minima pe datele de identificare
[MSE20, index20]=min(MSE_sim_id);
sys_degree20=ceil(index20/ordin_maxim);
sys_order20=order_vector(index20);
%=> MSE pe id: 0.72468; MSE pe val: 1.15986
figure
plot(y_id,'LineWidth',1)
hold on
plot(y_hat_sim_id(:, index20),'LineWidth',1)
title(sprintf("Comparatie intre \ny-id si y-
    hat-id pentru m=%d, na=nb=%d => MSE=%.5f",
    sys_degree20, sys_order20, MSE20))
legend("valori reale ale functiei (
    IDENTIFICARE)", "valori aproximate (
    SIMULARE)")
```

```
figure
plot(y_val,'LineWidth',1)
hold on
plot(y_hat_sim_val(:, index20),'LineWidth',1)
title(sprintf("Comparatie intre \ny-val si y-
    hat-val pentru m=%d, na=nb=%d => MSE=%.5f",
     sys_degree20, sys_order20, MSE_sim_val(
    index20)))
legend("valori reale ale functiei (VALIDARE)",
     "valori aproximate (SIMULARE)")

%% MSE minima pe datele de validare
[MSE21, index21]=min(MSE_sim_val);
sys_degree21=ceil(index21/ordin_maxim);
sys_order21=order_vector(index21);
%=> MSE pe id: 0.72468; MSE pe val: 1.15986
```

```
figure
plot(y_id,'LineWidth',1)
hold on
plot(y_hat_sim_id(:, index21),'LineWidth',1)
title(sprintf("Comparatie intre \ny-id si y-hat-id pentru m=%d, na=nb=%d => MSE=%.5f", sys_degree21, sys_order21, MSE_sim_id(index21)))
legend("valori reale ale functiei (IDENTIFICARE)", "valori aproximate (SIMULARE)")
figure
plot(y_val,'LineWidth',1)
hold on
plot(y_hat_sim_val(:, index21),'LineWidth',1)
title(sprintf("Comparatie intre \ny-val si y-
```

```
    hat-val pentru m=%d, na=nb=%d => MSE=%.5f",
     sys_degree21, sys_order21, MSE21))
legend("valori reale ale functiei (VALIDARE)",
     "valori aproximate (SIMULARE)")

%% grad mic, ordin mic => FENOMEN SUBANTRENARE
index50=1;
sys_degree50=ceil(index50/ordin_maxim);
sys_order50=order_vector(index50);
figure
plot(y_id,'LineWidth',1)
hold on
plot(y_hat_sim_id(:, index50),'LineWidth',1)
title(sprintf("Comparatie intre \ny-id si y-
    hat-id pentru m=%d, na=nb=%d => MSE=%.5f",
    sys_degree50, sys_order50, MSE_sim_id(
```

```
    index50)))
legend("valori reale ale functiei (IDENTIFICARE)", "valori aproximate (SIMULARE)")
figure
plot(y_val,'LineWidth',1)
hold on
plot(y_hat_sim_val(:, index50),'LineWidth',1)
title(sprintf("Comparatie intre \ny-val si y-hat-val pentru m=%d, na=nb=%d => MSE=%.5f", sys_degree50, sys_order50, MSE_sim_val(index20)))
legend("valori reale ale functiei (VALIDARE)", "valori aproximate (SIMULARE)")


```

```
%% --------------------MSE--------------------
figure
subplot(211),plot(1:grad_maxim, MSE_pr_id(1:grad_maxim), LineWidth=1)
xlabel("grad sistem")
ylabel("MSE id")
legend("MSE id")
subplot(212),plot(1:grad_maxim, MSE_pr_val(1:grad_maxim), LineWidth=1)
xlabel("grad sistem")
ylabel("MSE val")
legend("MSE val")
sgtitle('PREDICTIE: Evolutie MSE in functie de gradul sistemului')
figure
subplot(211),plot(1:ordin_maxim, MSE_pr_id(1:
```

```
    ordin_maxim), LineWidth=1)
xlabel("ordin sistem")
ylabel("MSE id")
legend("MSE id")
subplot(212),plot(1:ordin_maxim, MSE_pr_val(1:
    ordin_maxim), LineWidth=1)
xlabel("ordin sistem")
ylabel("MSE val")
legend("MSE val")
sgtitle('PREDICTIE: Evolutie MSE in functie de
    ordinul sistemului')
figure
subplot(211),plot(1:grad_maxim, MSE_sim_id(1:
    grad_maxim), LineWidth=1)
xlabel("grad sistem")
ylabel("MSE id")
```

```
legend("MSE id")
subplot(212),plot(1:grad_maxim, MSE_sim_val(1:grad_maxim), LineWidth=1)
xlabel("grad sistem")
ylabel("MSE val")
legend("MSE val")
sgtitle('SIMULARE: Evolutie MSE in functie de gradul sistemului')
figure
subplot(211),plot(1:ordin_maxim, MSE_sim_id(1:ordin_maxim), LineWidth=1)
xlabel("ordin sistem")
ylabel("MSE id")
legend("MSE id")
subplot(212),plot(1:ordin_maxim, MSE_sim_val(1:ordin_maxim), LineWidth=1)
```

```
xlabel("ordin sistem")
ylabel("MSE val")
legend("MSE val")
sgtitle('SIMULARE: Evolutie MSE in functie de ordinul sistemului')

figure
plot(y_val,'LineWidth',1)
hold on
plot(y_hat_pr_val(:, index11),"green", 'LineWidth',1)
hold on
plot(y_hat_sim_val(:,index21),"magenta",'LineWidth',1)
title("Comparatie intre y-val, y-hat-val (predictie), y-hat-val (simulare)")
```

```
legend("valori reale ale functiei (VALIDARE)", "valori aproximate (PREDICTIE)", "valori aproximate (SIMULARE)")
figure
plot(y_id,'LineWidth',1); hold on
plot(y_hat_pr_id(:, index10),"green", 'LineWidth',1)
hold on
plot(y_hat_sim_id(:,index20),"magenta",'LineWidth',1)
title("Comparatie intre y-id, y-hat-id (predictie), y-hat-id (simulare)")
legend("valori reale ale functiei (IDENTIFICARE)", "valori aproximate (PREDICTIE)", "valori aproximate (SIMULARE)")
```