

Identificarea ARX Neliniară folosind Aproximare Polinomială

Curs: Identificarea Sistemelor

Universitatea Tehnică din Cluj-Napoca

Numele Studenților:

Nistor Razvan-Iulian

Sitea Alexandru-Ionut

Pop Mihnea-Daniel

Indexul Unic al Proiectului: 5/15

Introducere – Descrierea Problemei

- **Scop:** Identificarea unui **model ARX neliniar polinomial** pentru un sistem dinamic necunoscut.
- **Metodă:** Regresie liniară aplicată pe structura ARX neliniară.
- **Date disponibile:**
 - id:** Set de identificare (antrenare).
 - val:** Set de validare (testare).
- **Obiectiv:** Obținerea unui model optim pentru **predicție** și **simulare**.
- **Importanță:** Predicții precise, simulări sigure și optimizarea performanței sistemului.

Structura Aproximatorului și Procedura de Găsire a Parametrilor

- **Model ARX Neliniar:**

$$y(k) = p(y(k-1), \dots, y(k-na), u(k-1), \dots, u(k-nb)) + e(k)$$

- **Vector de regresori:**

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

- **Metodă de Estimare:**

Regresie liniară pentru parametri.

Minimizarea **MSE**.

```
function phi = build_regressors(y, u, na, nb, nk)
    N = length(y);
    phi = zeros(N, na + nb);
    for k = 1:N
        for col = 1:na
            if k > col
                phi(k, col) = -y(k - col);
            end
        end
        for col = 1:nb
            if k > col + nk - 1
                phi(k, na + col) = u(k - col - nk + 1);
            end
        end
    end
end
```

```
function phi_nl = build_nonlinear_regressors(phi, m)
    [N, num_features] = size(phi);
    phi_nl = [];
    for i = 1:num_features
        for p = 2:m
            phi_nl = [phi_nl, phi(:, i).^p];
        end
    end
    for i = 1:num_features
        for j = i+1:num_features
            for p1 = 1:m
                for p2 = 1:m
                    if p1 + p2 <= m
                        phi_nl = [phi_nl, (phi(:, i).^p1) .* (phi(:, j).^p2)];
                    end
                end
            end
        end
    end
end
```

Caracteristici Esențiale ale Soluției

Date de Identificare

Seturi de date:
id și val

Parametri Configurabili:
na, nb, m

Vector de Regresori

Vector de Regresori:

Ieșiri întârziate: $y(k-1) \rightarrow y(k-na)$
Intrări întârziate: $u(k-1) \rightarrow u(k-nb)$

Structura vectorului:

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

Estimarea Parametrilor

Folosirea **regresiei liniare** pentru calcularea coeficienților polinomului.

Vectorul parametrilor:

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

Obiectiv: Minimizarea erorii medii pătratice (**MSE**).

Analiza Performanței

Predicție: Date reale.

Simulare: Ieșiri model

Obiectiv: Evaluarea preciziei cu **MSE**.

Validare

Identificare: Ajustare parametri.

Validare: Testare pe date noi.

Obiectiv: Precizie ridicată, fără supra-antrenare.

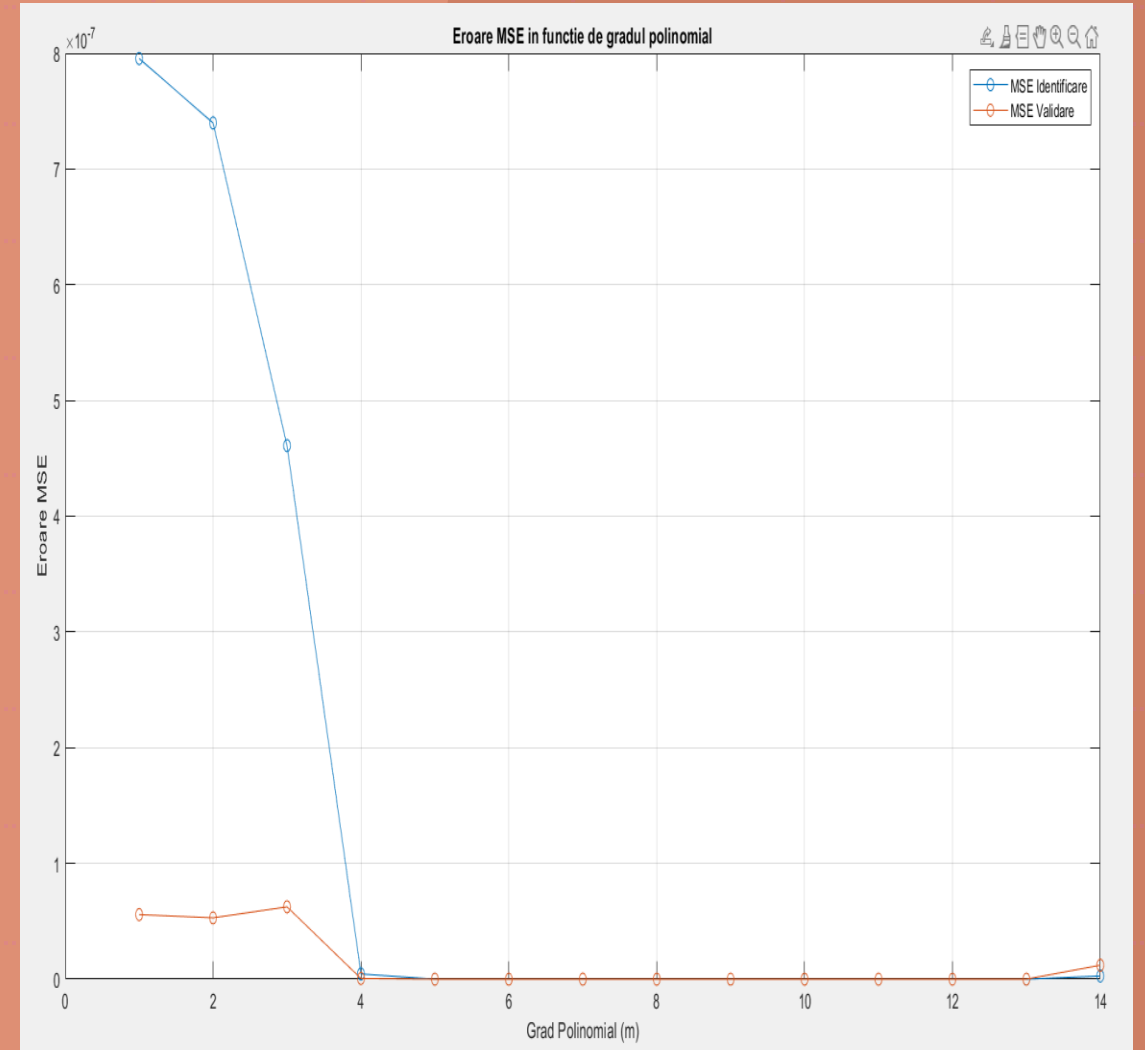
Minimizarea MSE

Obiectiv: Reducerea diferenței între ieșirile reale și cele estimate.

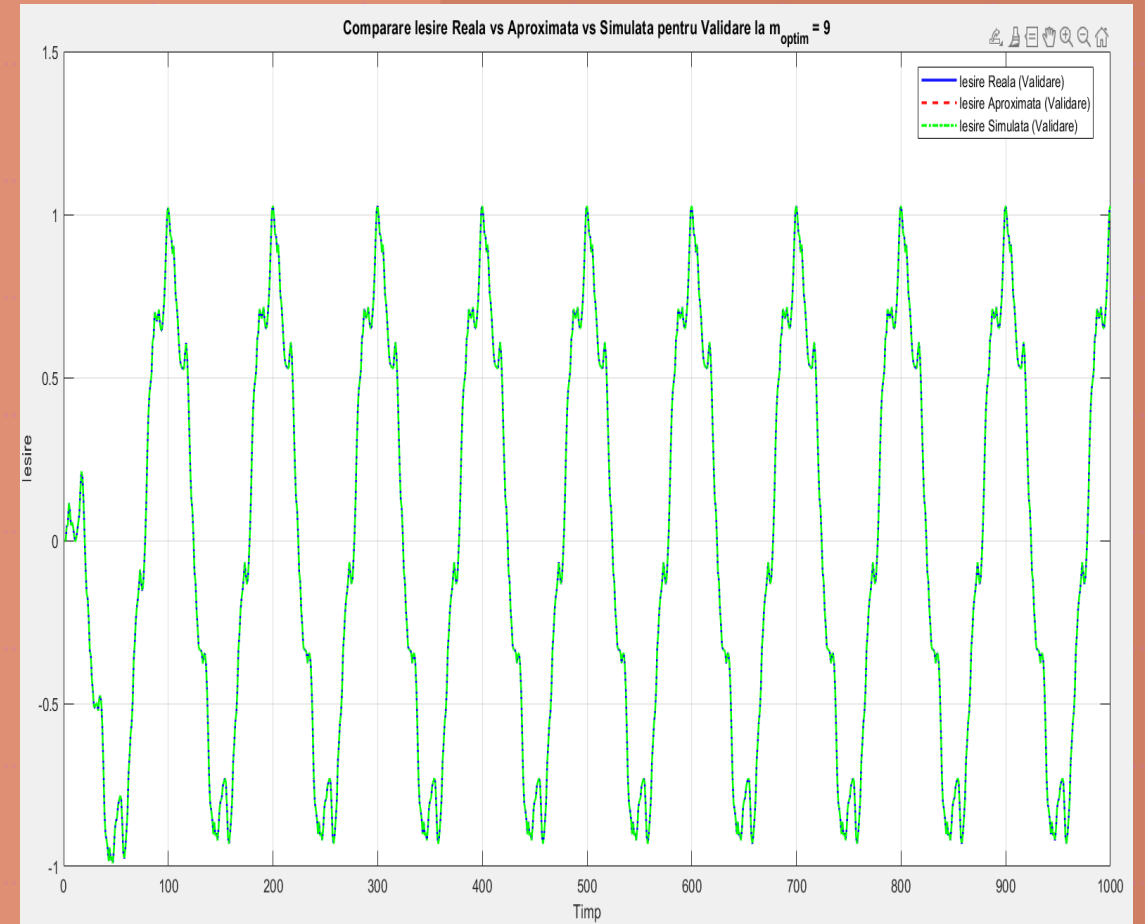
Rezultat: Parametri optimizați pentru o precizie maximă a modelului.

Rezultate de Reglare

– MSE în funcție de
 n_a , n_b și m



Compararea leșirii Aproximare vs. leșire Reală



Eroare de Predicție și Simulare – Identificare vs. Validare

MSE Identificare: 1.384e-17

MSE Validare: 2.0858e-17

MSE Simulare: 9.9015e-15

M_optim = 9

```
% Calculul erorii MSE pentru simulare  
mse_sim = mean((y_val - y_sim).^2);
```

```
% Calculul erorilor MSE pentru grad optim  
mse_id_optim = mean((y_id - Y_pred_id).^2);  
mse_val_optim = mean((y_val - Y_pred_val).^2);
```

Discuție asupra Rezultatelor

- **Performanța Modelului:**

Identificare: Eroarea **MSE** este **Mică** ($1.384e-17$ pentru grad optim $m=9$).

Validare: Eroarea **MSE** este **Mică** ($2.0858e-17$ pentru grad optim $m=9$).

- **Stabilitate Model:**

Modelul prezintă **consistență** între seturile de identificare și validare.

Diferența dintre **eroarea de predicție** și **eroarea de simulare** este **Mică**, ceea ce indică o bună adaptare a modelului pe setul de validare.

- **Supra-antrenare (Overfitting):**

Nu există semne de supra-antrenare?

Performanța pe setul de validare **confirmă generalizarea** modelului.

- **Observații Generale:**

Gradul polinomial **$m = 9$** și ordinele **$na = nb = 3$** influențează semnificativ precizia modelului.

Modelul este mai precis pentru **Predicție** decât pentru **Simulare** pe setul de validare.

Anexă - Codul Dezvoltat

```
% Incarcarea datelor de identificare si validare
```

```
load('iddata-15.mat');
```

```
u_id = id.u;
```

```
y_id = id.y;
```

```
u_val = val.u;
```

```
y_val = val.y;
```

```
% Structura optima ARX folosind criteriul AIC
```

```
data_id = iddata(y_id, u_id, 1);
```

```
data_val = iddata(y_val, u_val, 1);
```

```
orders = struc(1:3, 1:3, 1);
```

```
V = arxstruc(data_id, data_val, orders);
```

```
best_structure = selstruc(V, 'aic');
```

```
na = best_structure(1); % Ordinul iesirilor intarziate
```

```
nb = best_structure(2); % Ordinul intrarilor intarziate
```

```
nk = best_structure(3); % Intarzierea initiala
```

```
% Initializare parametri pentru analiza MSE in functie de  
gradul polinomial
```

```
m_max = 14;
```

```
mse_id = zeros(1, m_max);
```

```
mse_val = zeros(1, m_max);
```

Anexă - Codul Dezvoltat

```
for m = 1:m_max
    phi_id = build_regressors(y_id, u_id, na, nb,
                             nk);
    phi_val = build_regressors(y_val, u_val, na, nb,
                               nk);

    phi_nl_id = build_nonlinear_regressors(phi_id,
                                             m);
    phi_nl_val = build_nonlinear_regressors(phi_val,
                                              m);

    phi = [phi_id, phi_nl_id];
    phiV = [phi_val, phi_nl_val];

    theta = phi \ y_id;

    Y_pred_id = phi * theta;
    Y_pred_val = phiV * theta;

    mse_id(m) = mean((y_id - Y_pred_id).^2);
    mse_val(m) = mean((y_val - Y_pred_val).^2);
end
```

Anexă - Codul Dezvoltat

```
figure;  
plot(1:m_max, mse_id, '-o', 'DisplayName', 'MSE  
    Identificare');  
hold on;  
plot(1:m_max, mse_val, '-o', 'DisplayName', 'MSE  
    Validare');  
xlabel('Grad Polinomial (m)');  
ylabel('Eroare MSE');  
legend;  
title('Eroare MSE in functie de gradul polinomial');  
grid on;
```

```
m_optim = 9;  
  
phi_id = build_regressors(y_id, u_id, na, nb, nk);  
phi_val = build_regressors(y_val, u_val, na, nb, nk);  
  
phi_nl_id = build_nonlinear_regressors(phi_id,  
    m_optim);  
phi_nl_val = build_nonlinear_regressors(phi_val,  
    m_optim);
```

Anexă - Codul Dezvoltat

```
phi = [phi_id, phi_nl_id];  
phiV = [phi_val, phi_nl_val];
```

```
theta = phi \ y_id;
```

```
Y_pred_id = phi * theta;
```

```
Y_pred_val = phiV * theta;
```

```
mse_id_optim = mean((y_id - Y_pred_id).^2);
```

```
mse_val_optim = mean((y_val - Y_pred_val).^2);
```

```
disp(['MSE Identificare la m_optim = ', num2str(m_optim), ': ',  
      num2str(mse_id_optim)]);
```

```
disp(['MSE Validare la m_optim = ', num2str(m_optim), ': ',  
      num2str(mse_val_optim)]);
```

```
figure;
```

```
plot(y_val, 'b', 'DisplayName', 'Iesire Reala (Validare)');
```

```
hold on;
```

```
plot(Y_pred_val, 'r--', 'DisplayName', 'Iesire  
      Aproximata (Validare)');
```

```
xlabel('Timp');
```

```
ylabel('Iesire');
```

```
legend;
```

```
title(['Comparare Iesire Reala vs Aproximata pentru  
      Validare la m_{optim} = ', num2str(m_optim)]);
```

```
grid on;
```

Anexă - Codul Dezvoltat

```
N_val = length(y_val);  
y_sim = zeros(N_val, 1);  
phi_sim = zeros(N_val, na + nb);  
  
y_sim(1:na) = y_val(1:na);
```

```
for k = (na + 1):N_val  
    for col = 1:na  
        if k > col  
            phi_sim(k, col) = -y_sim(k - col);  
        end  
    end  
    for col = 1:nb  
        if k > col  
            phi_sim(k, na + col) = u_val(k - col);  
        end  
    end  
    phi_nl_sim = build_nonlinear_regressors(phi_sim(k, :), m_optim);  
    phi_total_sim = [phi_sim(k, :), phi_nl_sim];  
    y_sim(k) = phi_total_sim * theta;  
end
```

Anexă - Codul Dezvoltat

```
mse_sim = mean((y_val - y_sim).^2);

disp(['MSE Simulare la m_optim = ', num2str(m_optim), ': ',
      num2str(mse_sim)]);

figure;

plot(y_val, 'b-', 'LineWidth', 1.5, 'DisplayName', 'Iesire Reala (Validare)');

hold on;

plot(Y_pred_val, 'r--', 'LineWidth', 1.5, 'DisplayName', 'Iesire Aproximata
      (Validare)');

plot(y_sim, 'g-.', 'LineWidth', 1.5, 'DisplayName', 'Iesire Simulata (Validare)');

xlabel('Timp');

ylabel('Iesire');

legend('Location', 'best');

title(['Comparare Iesire Reala vs Aproximata vs Simulata pentru Validare la
      m_{optim} = ', num2str(m_optim)]);

grid on;
```