



PROIECT IDENTIFICAREA SISTEMELOR

Partea 2. ARX neliniar

Grupa 30135 semigrupa 1

Studenti:

- Bârgăoanu Bogdan Alexandru

Profesor îndrumător: Lucian Buşoniu

Cuprins

<i>INTRODUCERE</i>	3
<i>ALGORITM</i>	4
<i>REZULTATE</i>	6
<i>CONSIDERENTE</i>	9

Introducere

- **Scop** – Aproximarea comportamentului unui sistem neliniar cu iesirea afectata de zgomote.
- **Date utilizate** - Sunt date un set de date de intrare , si un set de date de iesire.
- **Cerinta** - Dezvoltarea unui model ARX neliniar de tip polinomial cu ordinele n_a , n_b si m configurabile, care sa reprezinte comportamentul unui sistem dinamic.

Algoritm

Vom porni de la structura modelului ARX neliniar pentru gasirea regresorilor:

$$\hat{y}(k) = p(y(k-1), \dots, y(k-n_a), u(k-n_k), u(k-n_k-1), \dots, u(k-n_k-n_b+1)) = p(d(k))$$

Considerand $n_a = n_b$, si $n_k = 1$ vom construi matricea puterilor corespunzatoare termenilor din polinom in functie de gradul m (<4).

Pentru calcularea regresorilor, se vor insuma termenii din polinom, la puterile lor corespunzatoare.

$$regresor(i, j) = \sum pol(i, p)^{m_{puteri}(j, p)}$$

unde $i \rightarrow$ indexul liniei pentru iesirea referinta;

$j \rightarrow$ indexul liniei din matricea de puteri;

$p \rightarrow$ indexul coloanei din matricea de puteri care reprezinta puterea elementului curent din polinom;

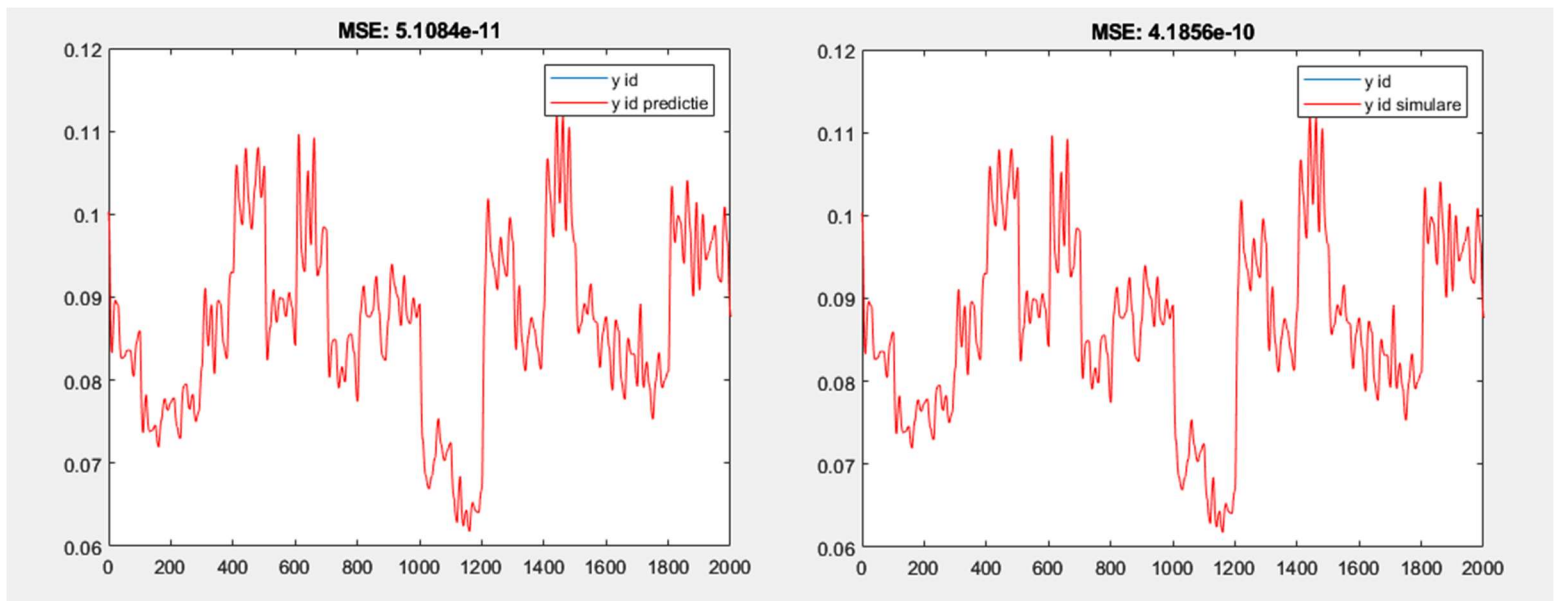
Dupa ce matricea de regresori este gasita, putem folosi principiul regresiei liniare pentru a obtine parametri modelului.

$$\Theta = \phi \backslash y ;$$

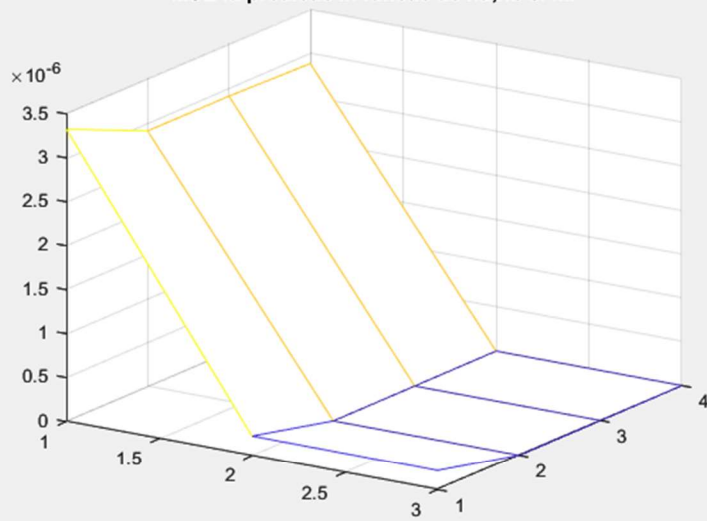
Pentru predictie se va folosi fix principiul mentionat anterior, iar pentru simulare, se va inlocui iesirea reala y cu iesirea generata de model, obtinuta prin predictie.

S-a ales na = nb 1:4 si m = 1:3.

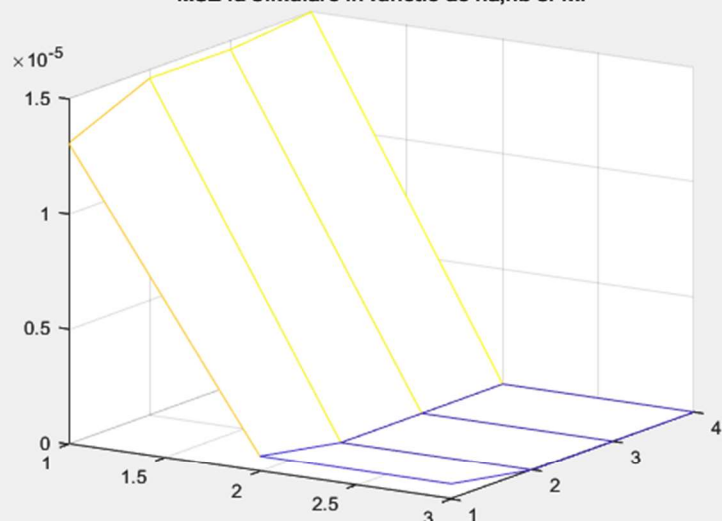
Rezultate



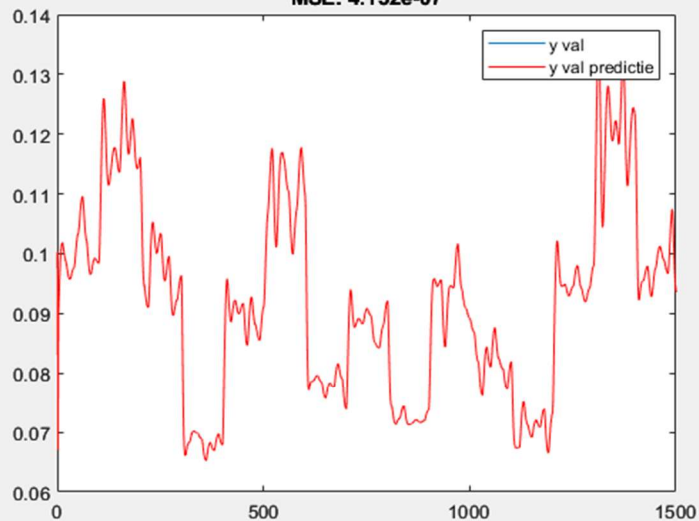
MSE id predictie in functie de na,nb si m.



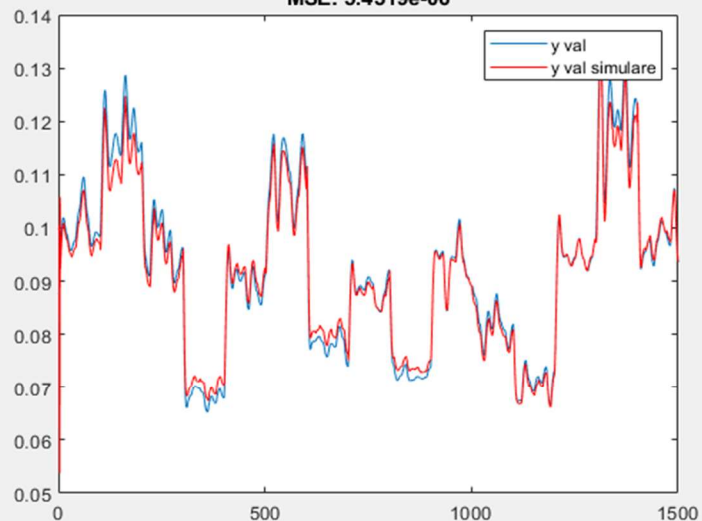
MSE id simulare in functie de na,nb si m.



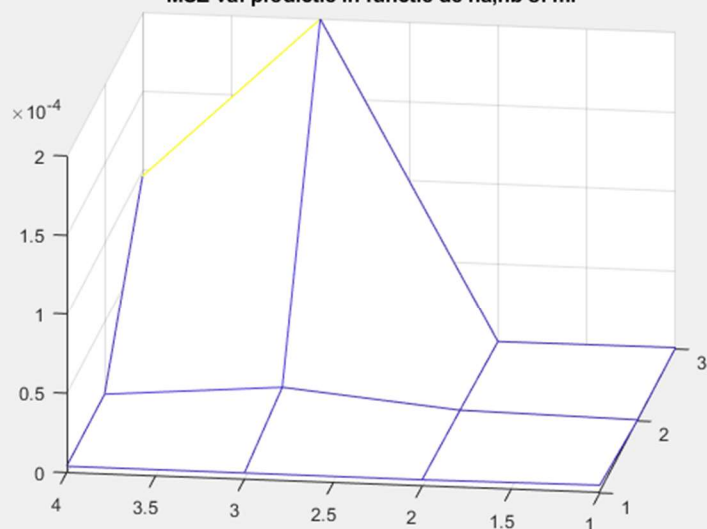
MSE: 4.132e-07



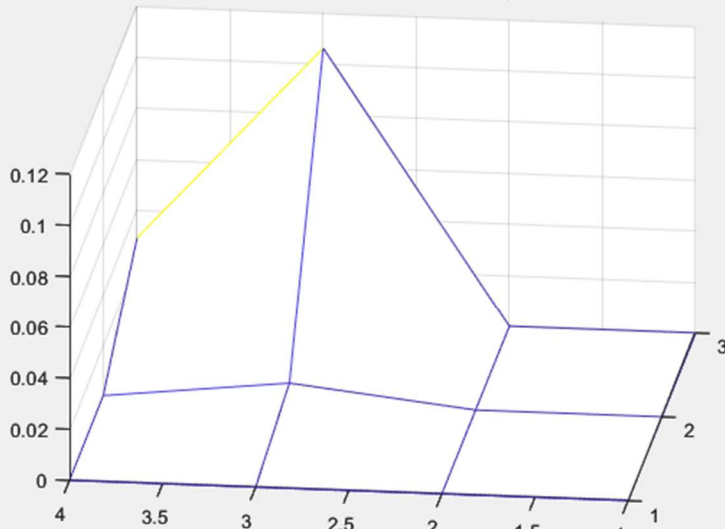
MSE: 5.4319e-06



MSE val predictie in functie de na,nb si m.



MSE val simulare in functie de na,nb si m.



Considerente

- Se poate observa minimizarea MSE la ambele seturi de date (identificare si validare) cu cresterea n_a si n_b , respectiv m .
- Ambele iesiri de predictie sunt aproape perfecte, totusi eroarea pe identificare este mult mai mica, ceea ce este si de asteptat.
- Simularea pe datele de identificarea este din nou, aproape perfecta, dar cu o eroare putin mai mare din cauza trecerii prin model a unei iesiri generate tot de model.
- Aceasta crestere a erorii este totusi destul de mica in practica , putand fi observata la simularea de pe setul de date de validare.
- In concluzie, modelul ARX generat aproximeaza iesirea neliniara destul de bine, dar trebuie luati in calcul si parametrii n_a, n_b si m , deoarece modelul ar putea intra in overfitting la prezenta a prea multi regresori.

Cod complet

```
load('iddata-11.mat');
u_id = id.InputData;
y_id = id.OutputData;
u_val = val.InputData;
y_val = val.OutputData;

e_id_predictie = zeros(length(y_id),1);
e_id_simulare = zeros(length(y_id),1);
e_val_predictie = zeros(length(y_val),1);
e_val_simulare = zeros(length(y_val),1);
MSE_id_predictie = zeros(4,3);
MSE_id_simulare = zeros(4,3);
MSE_val_predictie = zeros(4,3);
MSE_val_simulare = zeros(4,3);
min_MSE_id_p = inf;
min_MSE_id_s = inf;
min_MSE_val_p = inf;
min_MSE_val_s = inf;

for i=1:4
    for m = 1:3
        na = i;
        nb = i;

        [y_id_predictie,y_id_simulare] = getPredictieSimulare(na,nb,m,u_id,y_id,u_id,y_id);
        e_id_predictie = y_id - y_id_predictie;
        MSE_id_predictie(na,m) = 1/length(e_id_predictie) * sum(e_id_predictie.^2);

        if(MSE_id_predictie(na,m) < min_MSE_id_p)
            min_MSE_id_p = MSE_id_predictie(na,m);
            y_id_optim_p = y_id_predictie;
        end

        e_id_simulare = y_id - y_id_simulare;
        MSE_id_simulare(na,m) = 1/length(e_id_simulare) * sum(e_id_simulare.^2);
```



```

if(MSE_id_simulare(na,m) < min_MSE_id_s)
min_MSE_id_s = MSE_id_simulare(na,m);
y_id_optim_s = y_id_simulare;
end

[y_val_predictie,y_val_simulare] = getPredictieSimulare(na,nb,m,u_id,y_id,u_val,y_val);
e_val_predictie = y_val - y_val_predictie;
MSE_val_predictie(na,m) = 1/length(e_val_predictie) * sum(e_val_predictie.^2);

if(MSE_val_predictie(na,m) < min_MSE_val_p)
min_MSE_val_p = MSE_val_predictie(na,m);
y_val_optim_p = y_val_predictie;
end

e_val_simulare = y_val - y_val_simulare;
MSE_val_simulare(na,m) = 1/length(e_val_simulare) * sum(e_val_simulare.^2);

if(MSE_val_simulare(na,m) < min_MSE_val_s)
min_MSE_val_s = MSE_val_simulare(na,m);
y_val_optim_s = y_val_simulare;
end
end
end

figure;
plot(y_id);
hold on;
plot(y_id_optim_p,'Color','r');
legend('y id','y id predictie');
title(['MSE: ',num2str(min_MSE_id_p)]);

figure;
mesh(MSE_id_predictie);
title('MSE id predictie in functie de na,nb si m.');
```

```

figure;
plot(y_id);
hold on;
plot(y_id_optim_s,'Color','r');
legend('y id','y id simulare');
title(['MSE: ',num2str(min_MSE_id_s)]);

figure;
mesh(MSE_id_simulare);
title('MSE id simulare in functie de na,nb si m.');
```

```

figure;
plot(y_val);
hold on;
```

```

plot(y_val_optim_p,'Color','r');
legend('y val','y val predictie');
title(['MSE: ',num2str(min_MSE_val_p)]);

figure;
mesh(MSE_val_predictie);
title('MSE val predictie in functie de na,nb si m.');
```

```

figure;
plot(y_val);
hold on;
plot(y_val_optim_s,'Color','r');
legend('y val','y val simulare');
title(['MSE: ',num2str(min_MSE_val_s)]);

figure;
mesh(MSE_val_simulare);
title('MSE val simulare in functie de na,nb si m.');
```

```
function [y_predictie,y_simulare] = getPredictieSimulare(na,nb,m,u_id,y_id,u_val,y_val)
```

```
%generare set de puteri
```

```

i = 1;
puteri = zeros(1, na + nb);
combinari = [];
matrice_puteri = [];
```

```
while true
```

```

    combinari(i, :) = puteri;
    i = i + 1;
    j = 1;
```

```
    while j <= na + nb && puteri(j) == m
```

```

        puteri(j) = 0;
        j = j + 1;
```

```
    end
```

```
    if j > na + nb
```

```
        break;
```

```
    else
```

```
        puteri(j) = puteri(j) + 1;
```

```
    end
```

```
end
```

```
k = 1;
```

```
for i = 1:length(combinari)
```

```
    if sum(combinari(i, :)) <= m
```

```
        matrice_puteri(k, :) = combinari(i, :);
```

```

        k = k + 1;
    end
end

%generare phi_id pentru aflarea parametrilor theta
phi_id = zeros(length(y_id),na + nb);
phiLiniar = zeros(length(y_id),na + nb);
for i=1:length(y_id)
    for j=1:na
        if(i > j)
            phiLiniar(i,j) = y_id(i-j);
            phiLiniar(i,j+na) = u_id(i-j);
        end
    end
end

for k=1:length(matrice_puteri)
    element_phi_id = 1;
    for p=1:na + nb
        element_phi_id = element_phi_id*(phiLiniar(i,p)^matrice_puteri(k,p));
    end
    phi_id(i,k) = element_phi_id;
end
end

theta = phi_id \ y_id;

%generare phi_val pentru predictie
phi_val = zeros(length(y_val),na + nb);
phiLiniarValidare = zeros(length(y_val),na + nb);
for i=1:length(y_val)
    for j=1:na
        if(i > j)
            phiLiniarValidare(i,j) = y_val(i-j);
            phiLiniarValidare(i,j+na) = u_val(i-j);
        end
    end
end

for k=1:length(matrice_puteri)
    element_phi_val = 1;
    for p=1:na + nb
        element_phi_val = element_phi_val*(phiLiniarValidare(i,p)^matrice_puteri(k,p));
    end
    phi_val(i,k) = element_phi_val;
end
end

y_predictie = phi_val*theta;

%generare phi_sim pentru simulare

```

```

phi_sim = zeros(length(y_predictie),na + nb);
phiLiniarPredictie = zeros(length(y_predictie),na + nb);
for i=1:length(y_predictie)
    for j=1:na
        if(i > j)
            phiLiniarPredictie(i,j) = y_predictie(i-j);
            phiLiniarPredictie(i,j+na) = u_id(i-j);
        end
    end
    for k=1:length(matrice_puteri)
        element_phi_predictie = 1;
        for p=1:na+nb
            element_phi_predictie = element_phi_predictie*(phiLiniarPredictie(i,p)^matrice_puteri(k,p));
        end
        phi_sim(i,k) = element_phi_predictie;
    end
end
y_simulare = phi_sim*theta;
end

```