Arx Neliniar

Profesor indrumator:
Lucian Busoniu

Gherghinescu Dragos
Barsan Ilie
Gota Radu

Universitatea Tehnica
Din Cluj-Napoca
FACULTATEA DE AUTOMATICA SI CALCULATOARE
DEPARTAMENTUL AUTOMATICA

Grupa 30135
Indici 5/16

Descrierea problemei

- Se da un set de date cu o intrare si o iesire
- Dinamica poate fi neliniara si iesirea afectata de zgomot
- Se cere
  - programarea unei functii care genereaza un model ARX neliniar de tip polinomial cu ordinele na,nb,nk,m configurabile.
  - o procedura de regresie pentru gasirea parametrilor
  - utilizarea modelului prin predictie si simulare

Structura aproximatorului

- Aproximatorul este reprezentat de doua functii numite “generatePoly”
- Atat pentru predictie cat si pentru simulare se construieste o matrice de puteri prin functia “mop”
- Aceasta matrice se parcurge cu mai multe loop-uri de tip for care construiesc termenii polinomului cerut.

```
function P = mop(grade,na,nb)  %%matrice de puteri ale regresorilor

nr_elem = na+nb;

powers = unique(nchoosek(repmat(0:grade, 1, nr_elem), nr_elem), 'row');
P = powers(sum(powers, 2) <= grade, :);
```

```
for k = 1:length(d)
   for i = 1:length(P)
      a(i) = 1;
      for j=2:na+nb+1
         a(i) =  a(i)*d(k,j) .^(P(i,j-1));
      end
      PHI(k,i) = a(i);
   end
end
```

Predictie vs Simulare

- Pentru predictie matricea de regresori se construieste o singura data
- Se aplica algoritmul de creare a polinomului
- Se afla parametrii prin regresie si aproximarile

- Pentru simulare matricea de regresori se construieste treptat
- La fiecare pas se aplica algoritmul de creare a polinomului
- La fiecare pas se afla parametrii prin regresie si aproximarea curenta

Rezultate de reglare

MSE in functie de na,nb,nk,m

Pentru simplitate na=nb,iar nk=1,m configurabil

- Predictie vs Id

grad \ na+nb		
0.0106	0.0031	0.0024
0.0093	9.7518e-04	2.1746e-04
0.0085	1.7994e-04	2.8390e-05
0.0084	5.0795e-05	1.8070e-05

- Simulare vs Id

grad \ na+nb		
0.2544	0.0395	0.0252
0.2616	NaN	NaN
0.2091	0.0336	NaN
0.2168	0.0168	NaN

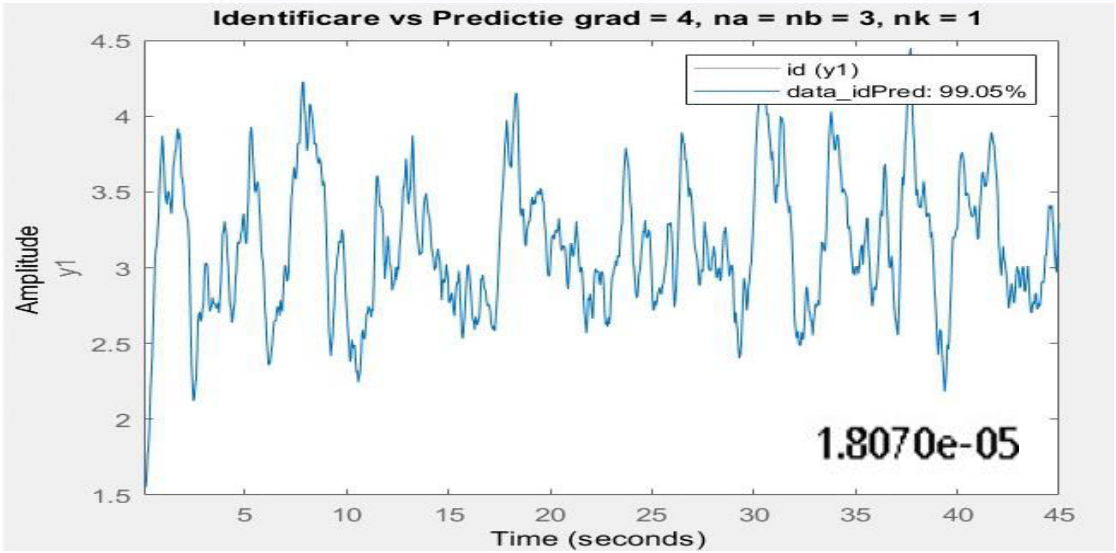
- Predictie vs Val

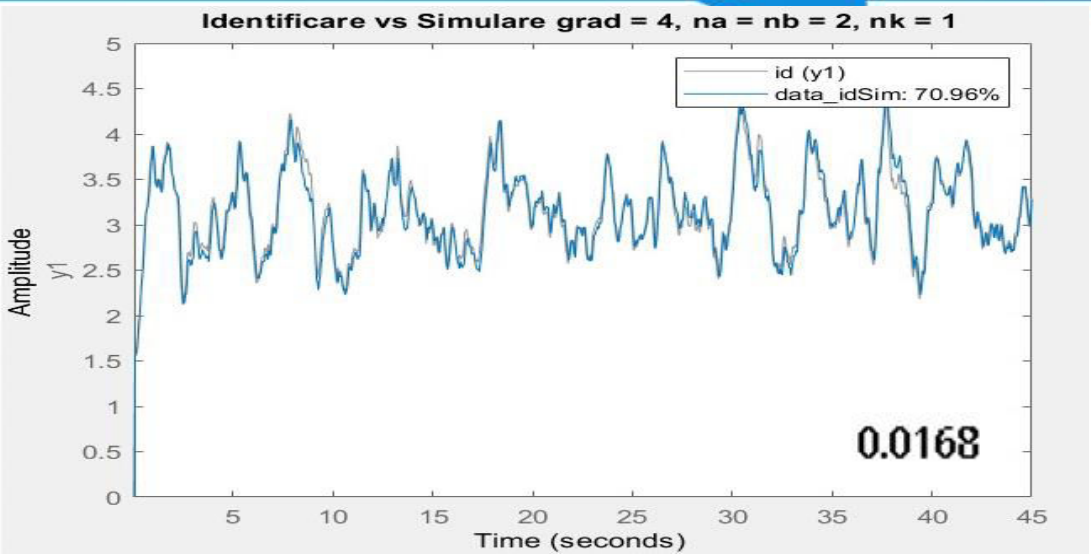
grad \ na+nb		
0.0040	0.0029	0.0023
0.0026	8.8894e-04	2.8370e-04
0.0018	1.5154e-04	0.0022
0.0017	3.2990e-05	94.3409

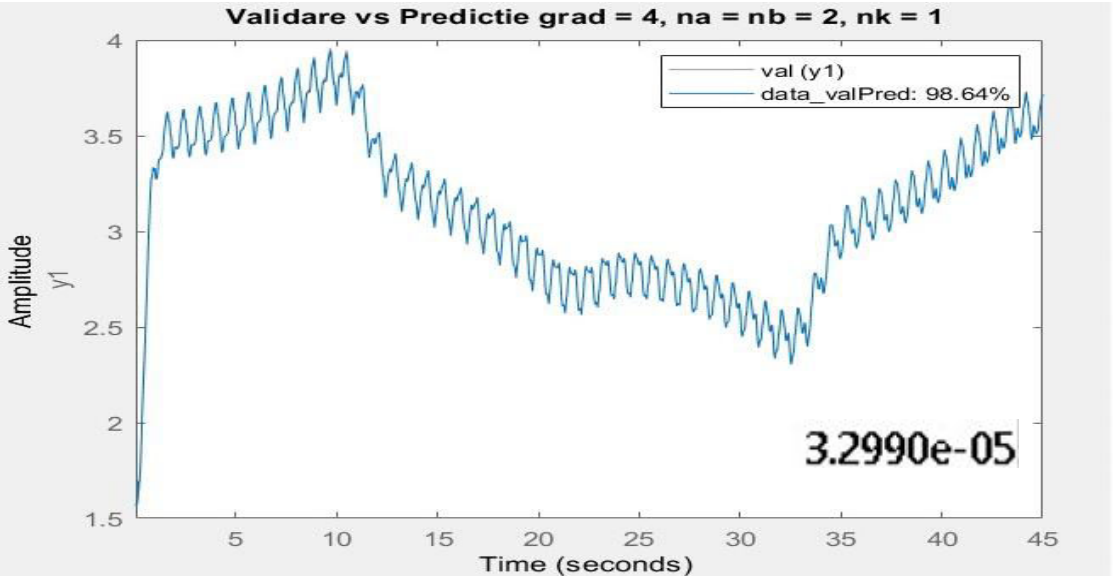
- Simulare vs Val

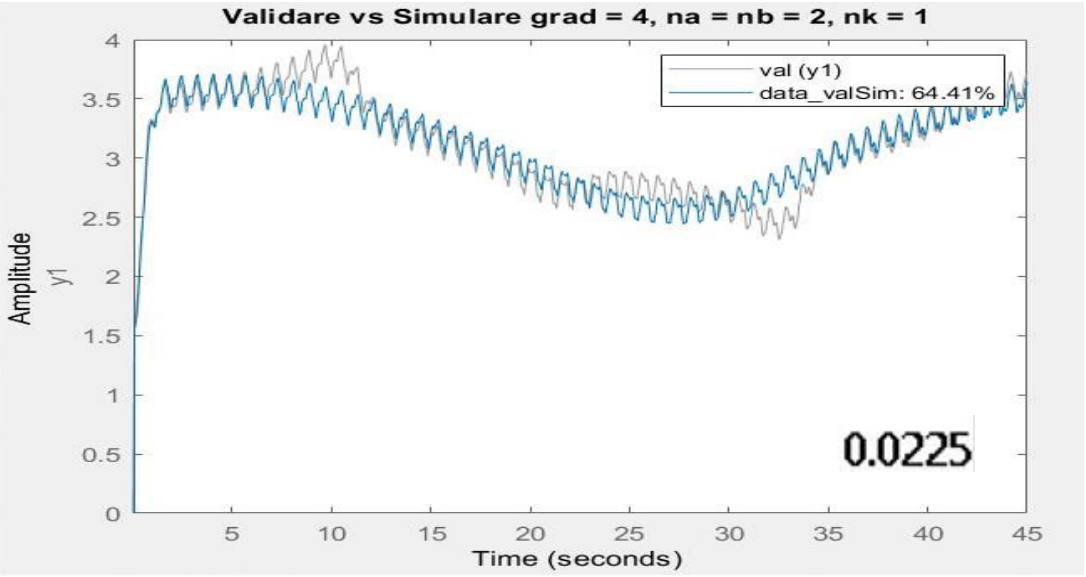
grad \ na+nb		
0.2279	0.0503	0.0385
0.2279	0.1015	NaN
0.2386	0.0320	NaN
0.3104	0.0225	NaN

Comparare iesiri









Probleme si avantaje ale algoritmului

- *Probleme:*
  - Consumator de memorie
  - Timp de rulare mare(aprox 47 secunde pentru na=nb=1:3,grad 1:4)
- *Avantaje:*
  - na,nb,nk,m configurabile
  - cod compact si usor de inteles
  - programul functioneaza

Concluzie

- In concluzie, aproximatorul functioneaza pe orice set de date de tipul intrare-iesire.
- Cele mai bune rezultate sunt la gradul 4, na=nb=2/3
- Chiar daca nk variaza
  - Cele mai bune aproximari sunt cele pe predictie
  - Simulare duce la un grad de potrivire mai mic
  - Gradul de potrivire se schimba cu maxim un procent