Universitatea Tehnică din Cluj-Napoca

Facultatea de Automatica și Calculatoare

Disciplina:

Identificarea Sistemelor

Proiect:

Identificarea si Simularea unui sistem de ordin 2

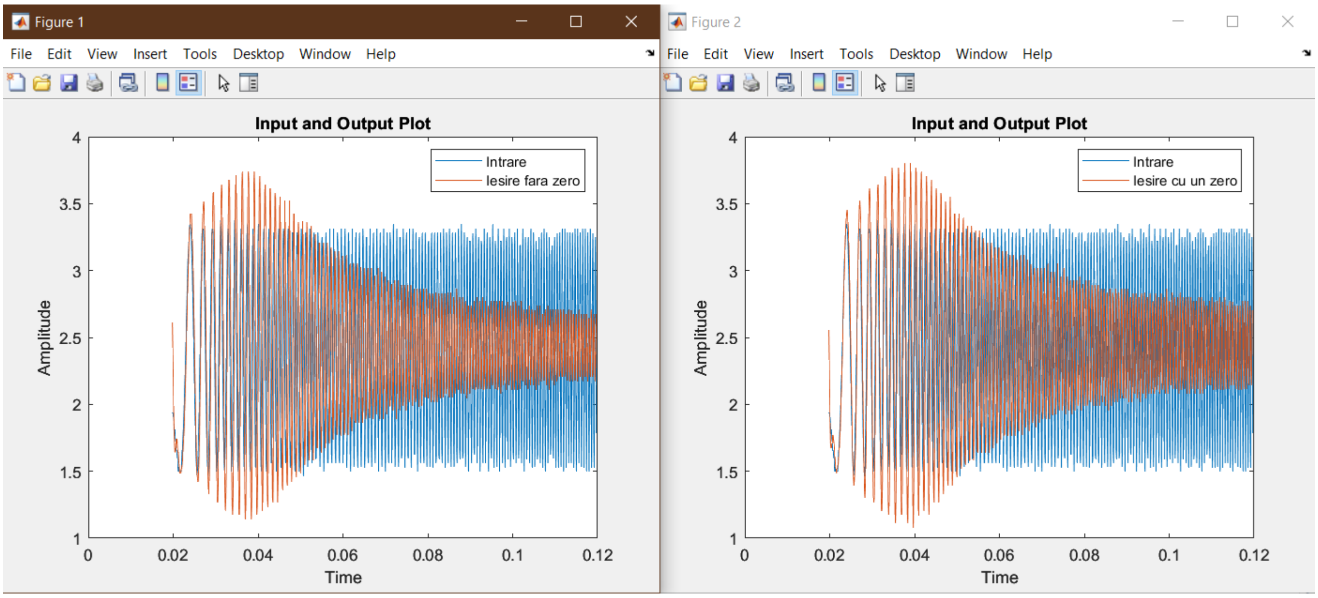
Studentul: Moldovan Dan-Alexandru

Grupa: 30132_2

Coordonator: Prof. univ. dr. Dobra Petru

Pentru proiect am avut de identificat si simulat un sistem de ordin 2 fără zerouri și un alt sistem de ordin 2 cu un zerou cu ajutorul metodelor neparametrice si parametrice, unde ni s-a oferit un set de date in care avem 4 vectori ce conțineau: timpul, semnalul de intrare și cele 2 sisteme.

Cu ajutorul funcției din MATLAB, plot(t, u, t, y) pentru afișarea semnalului sistemului fără zerouri și plot(t, u, t, x) pentru cel cu un zerou, putem obține o copie a reprezentări de pe osciloscopul din laborator.



(Fig. 1&2 intrarea cu albastru – u, ieșirea sis. ord. 2 cu/fără zerouri cu portocaliu – y,x)

```
%Plot intrare/iesire y
figure
plot(t, u, t, y)
xlabel('Time')
ylabel('Amplitude')
legend('Intrare', 'Iesire fara zero' )
title('Input and Output Plot')

%Plot intrare/iesire x
figure
plot(t, u, t , x)
xlabel('Time')
ylabel('Amplitude')
legend('Intrare', 'Iesire cu un zero')
title('Input and Output Plot')
```

1. Identificarea sistemului de ordin 2 fără zerouri printr-o metoda neparametrică

In continuare am făcut identificarea sistemului pe ieșirea y fără zero prin ***Metoda de Rezonanta***:

Am determinat 4 puncte de pe grafic pe perioada rezonantei:

- 2 pe perioada cu amplitudine maxima pozitiva
- 2 pe perioada cu amplitudine maxima negativa

```
ymin=180; %minim iesire
ymax=187; %maxim iesire
umin=178; %minim intrare
umax=185; %maxim intrare
```

Fenomenul de rezonanță se produce atunci când un sistem răspunde la o frecvență specifică cu o amplitudine mai mare decât la alte frecvențe, iar această reacție puternică la o anumită frecventă se datorează sincronizării undelor cu frecvența naturală a sistemului.

Pe baza punctelor selectate am determinat:

*Intervalul de timp:	$T_r = \left(t(y_{max}) - t(y_{min})\right) \cdot 2$ $\boldsymbol{T_r = 0.0014}$
*Pulsația de rezonanta:	$\omega_r = 2 \cdot \frac{\pi}{T_r}$ $\boldsymbol{\omega_r = 4.4880 \cdot 10^3}$
*Modul rezonanta:	$M_r = \frac{y_1(y_{\min}) - y_1(y_{\max})}{u(u_{\min}) - u(u_{\max})}$ $\boldsymbol{M_r = 1.4068}$
*Factor de amortizare:	$\zeta = \sqrt{\frac{M_r - \sqrt{M_r^2 - 1}}{2 \cdot M_r}}$ $\boldsymbol{\zeta = 0.4544}$
*Pulsația naturala:	$\omega_n = \frac{\omega_r}{\sqrt{1 - 2 \cdot \zeta^2}}$ $\boldsymbol{\omega_n = 5.8575 \cdot 10^3}$
*Factor de proporționalitate:	$K = \frac{\bar{y}}{\bar{u}}$ $\boldsymbol{K = 1.0081}$

- Modulul de rezonanță (M_r) a fost calculat cu ajutorul a 4 indexi de pe grafic acolo unde apare fenomenul de rezonanță;

- Perioada de oscilație (T) a fost calculată cu ajutorul celor 2 indexi de la modulul de rezonanță cu care putem face o jumate de perioadă și după o dublăm și avem perioada;
- Pulsația de rezonanță (ω_r) este frecvența la care un sistem prezintă un răspuns maxim sau o amplitudine maximă în cazul unei stimulări la acea frecvență;
- Factorul de amortizare (tita sau ζ) a fost calculat cu ajutorul modulului de rezonanță și pentru a fi o valoare valida trebuie să fie cuprinsă între 0 și 1;
- Pulsația naturală de oscilație (ω_n), este o mărime caracteristică a unui sistem de ordin 2 și prezintă viteza de oscilație a sistemului;
- Factorul de proporționalitate (K), este o constanta care are ca valoare locul unde se stabilizează ieșirea sistemul, astfel în cazul nostru se v-a face o medie a tuturor valorilor de pe ieșire și respectiv de pe intrare și se va face raportul acestor valori;

Matlab:

```
Mr=(y(ymin)-y(ymax))/(u(umin)-u(umax)) %modulul de rezonanta
Tr=(t(ymax)-t(ymin))*2 %perioada de rezonanta
zeta=(sqrt((Mr-sqrt(Mr^2-1)))/2*Mr)%factorul de amortizare
wr = (2*pi/Tr) %pulsatia de rezonanta
wn = wr/sqrt(1-2*zeta^2) %pulsatia naturala
K=mean(y)/mean(u); %=1.0081
```

Astfel obținând toate valorile descrise anterior putem realiza funcția de transfer care reprezintă un model matematic a relației între intrare și ieșire ale unui sistem care de obicei este în domeniul frecvențial și poate evidenția modul în care sistemul răspunde la diferite frecvențe sau diferite semnale la intrare. În cazul nostru funcția de transfer fiind că este de ordinul 2 va arta în genul acesta:

$$H_s = K\frac{\omega_n^2}{s^2 + 2 * \zeta * \omega_n s + \omega_n^2}$$

$$H(s) = \frac{3.459 \cdot 10^7}{s^2 + 5323s + 3.431 \cdot 10^7}$$

Matlab:

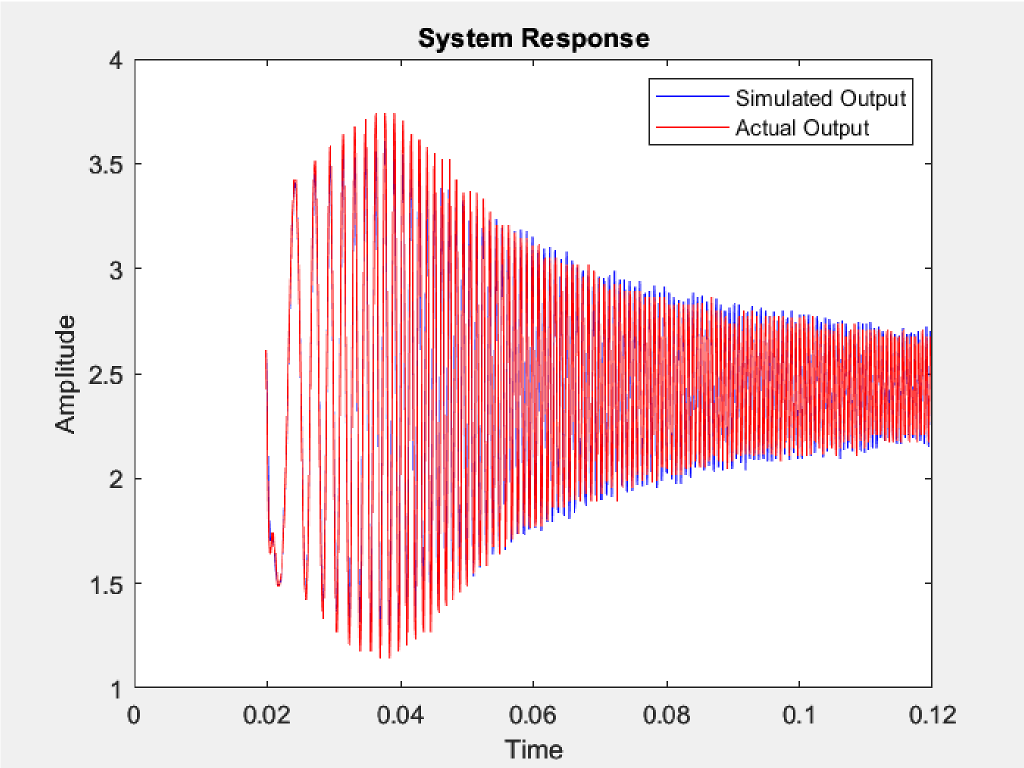
```
num=K*wn^2;
den=[K 2*zeta*wn wn^2];
Hdes=tf(num,den)
```

Pentru a putea genera semnalul simulat, ne vom folosi de comanda **ss** din **Matlab** împreună cu cele 4 matrici A,B,C,D, care ne va oferi spațiul stărilor a semnalului simula, si comanda **lsim** pentru a genera semnalul simulat:

```
A=[0 1;-wn^2 -2*zeta*wn];
B=[0;K*wn^2];
C=[1 0];
D=0;
%convertire in spatiul starilor
sys=ss(A,B,C,D);
%generare semnal simulat
ysim=lsim(sys, u , t,[y(1),0]);
```

Validare semnal simulat

Pentru a putea face validarea semnalului simulat in interfața **Matlab** vom folosi comanda **plot**, iar ca si parametri vom alege semnalul de ieșire si semnalul simulat:



Astfel pentru a determina eroarea medie pătratica relativa am folosit formula:

$$\varepsilon_{MPN} = \frac{\|y - y_M\|}{\|y - \bar{y}\|} \cdot 100$$

$$\varepsilon_{MPN} = 20.2976\%$$

```
% Calcularea eroare patratica normalizata
empn = norm(y - ysim1) / norm(y - mean(y))
* 100;
```

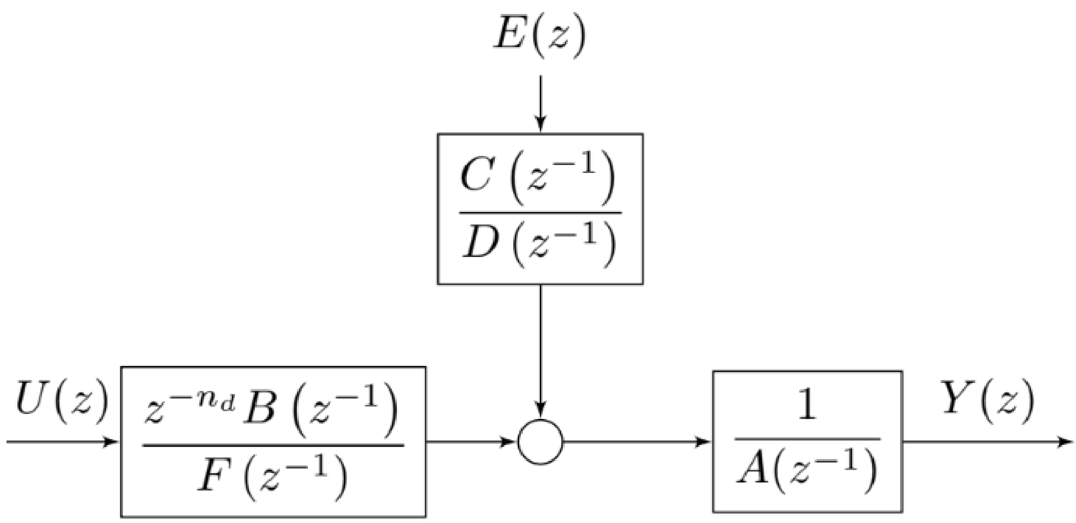
2. Identificarea sistemului de ordin 2 fără zerouri prin 2 metode parametrice

Pentru a putea face simularea sistemului , va trebui ca datele de intrare si datele de ieșire sa le grupez intr-un singur obiect cu ajutorul comenzi de tip **iddata**. Si pentru a determina ordinul sistemului vom folosi **n4sid.**

```
Te=t(2)-t(1);%perioada esantionare
date=iddata(y, u, Te);
%verificare ordin
n4sid(date,1:10)
```

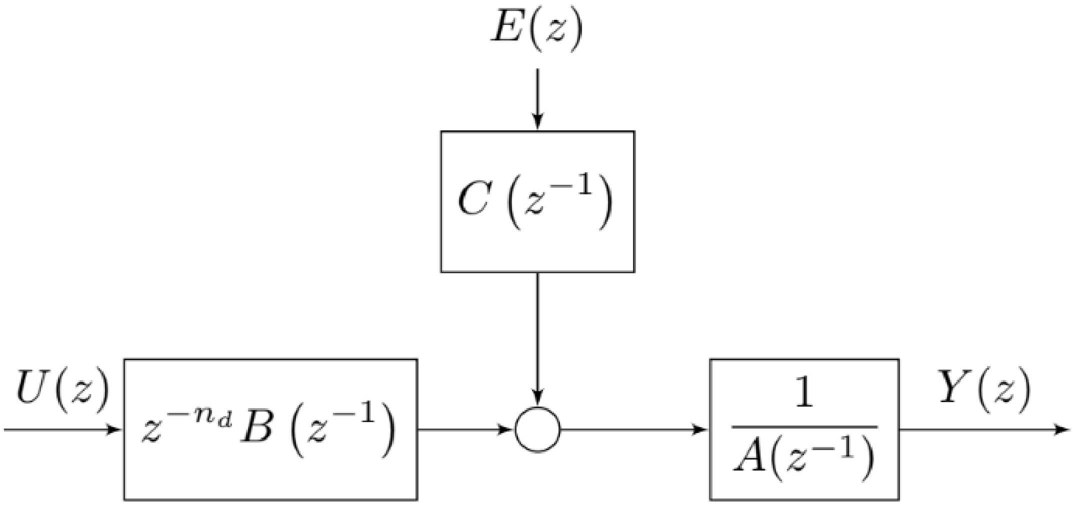
Cele două metode parametrice trebuie ca una dintre ele să treacă testul de autocorelație adică metodele „*armax*”(Metoda celor mai mici pătrate extinsa) și „*arx*”(Metoda celor mai mici pătrate recursiva) și cealaltă metodă trebuie să treacă testul de intercorelație, metodele „*iv4*”(Metoda variabilelor instrumentale) și „*oe*”(Metoda erorii de ieșire).

O metoda parametrică este reprezentat de un model general proces+perturbație al unui sistem polinomial discret, iar identificarea sistemului constă în determinarea coeficienților polinomilor A, B, C, D și F.



Ca un model să treacă testul de autocorelație trebuie ca eroarea de predicție formează o secvență de tip zgomot alb, iar ca un model să treacă testul de intercorelație trebuie ca eroarea de predicție să fie decorelată de predicțiile obținute ale ieșiri.

- **Metoda ARMAX**



Modelul ARMAX artă așa A(z)y(t) = B(z)u(t) + C(z)e(t) și polinoamele din componență au valorile:

A(z) = 1 - 1.45 z^-1 + 0.6667 z^-2

B(z) = 0.2316 z^-1 - 0.01368 z^-2

C(z) = 1 - 1.045 z^-1 + 0.2744 z^-2

În MATLAB funcția a fost declarată Marmax=armax(date,[na nb nc nk]) care are ca parametri in această ordine:

nA = 2 (numărul polilor),

nB = 2 (numărul zerourilor),

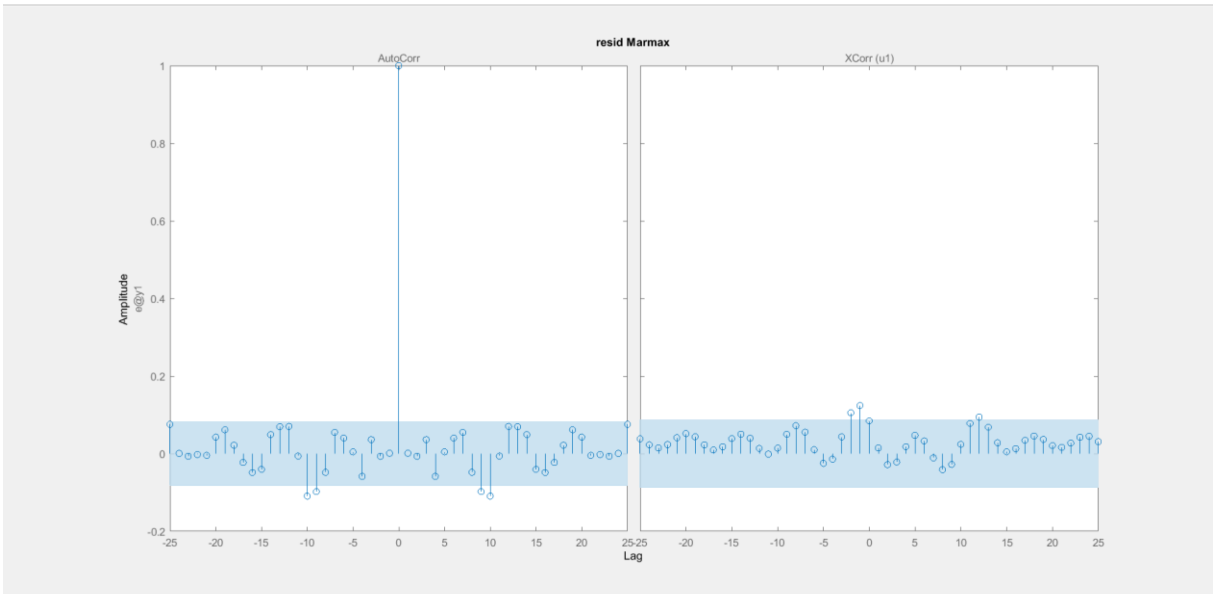
nC = 2 (numărul tacților de întârziere),

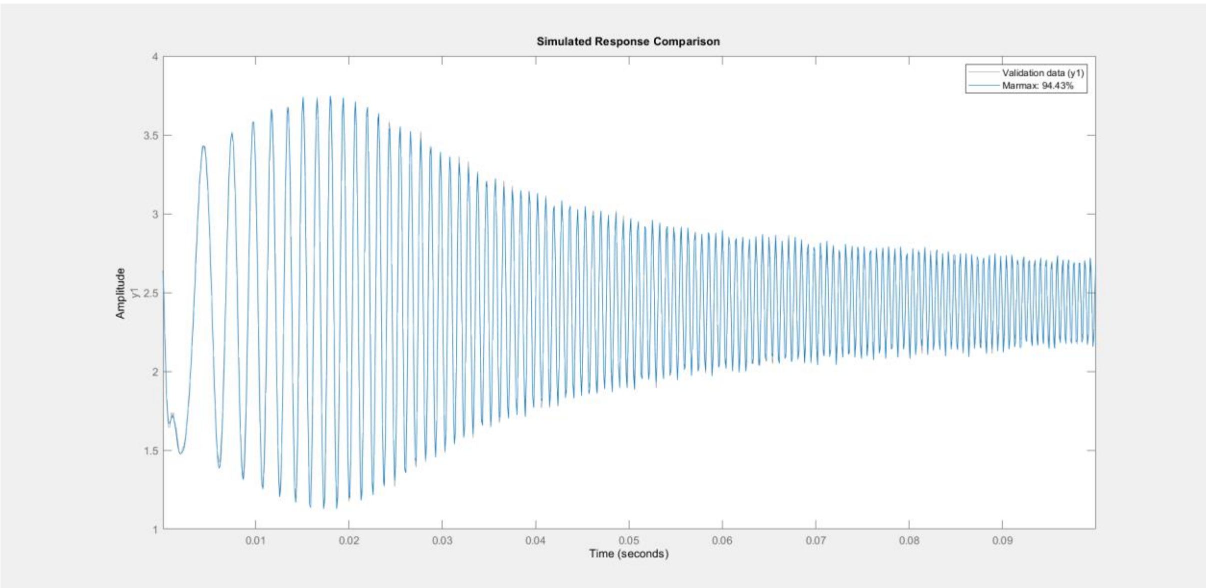
nk = 1(timp mort) și funcția de transfer în discret este:

Hz $=\frac{0.2316\, z^{-1} - 0.01368\, z^{-2}}{1 - 1.45\, z^{-1} + 0.6667\, z^{-2}}$ și funcția de transfer => Hs = $\frac{1466\, s + 2.711e07}{s^2 + 4054s + 2.69e07}$

Trecerea din discret în continuu se face cu ajutorul funcției „*d2c*“ (d2c(Hz_armax,'zoh') unde zoh înseamnă zero-order hold).

Următoarele grafice unu dintre ele v-a conține compararea sistemelor și în celălalt grafic v-a fi testul de autocorelație:

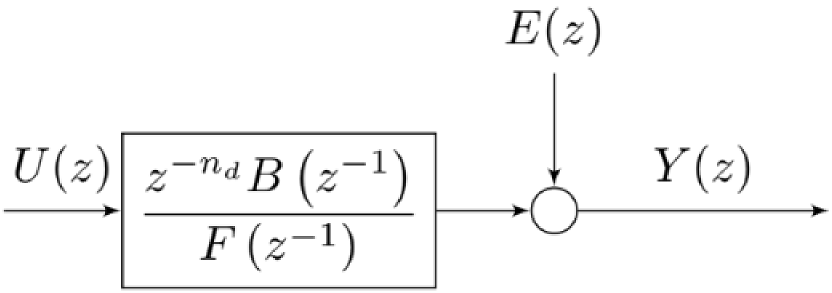




După cum se poate observa comparația are eroare de 5,57%, iar testul de autocorelație îl trece cu succes.

```
%% Metoda armax

Marmax=armax(date, [2 2 2 1])

figure,resid(date, Marmax),shg,title('resid Marmax')
figure,compare(date,Marmax),shg

Hz_arx1 = tf(Marmax.B,Marmax.A,Te,'variable','z^-1') %functia de transfer in
discret
Hs_arx1 = d2c(Hz_arx1,'zoh') %functia de transfer in continu
```

- **Metoda OE**



Modelul OE arată așa y(t) = [B(z)/F(z)]u(t) + e(t) și polinoamele din componență au valorile:

B(z) = 0.2334 z^-1 + 0.01292 z^-2

F(z) = 1 - 1.445 z^-1 + 0.6638 z^-2

În MATLAB funcția a fost declarată oe(data_id,[nb nf nk]) care are ca parametri in această ordine

nb = 2 (numărul zerourilor),

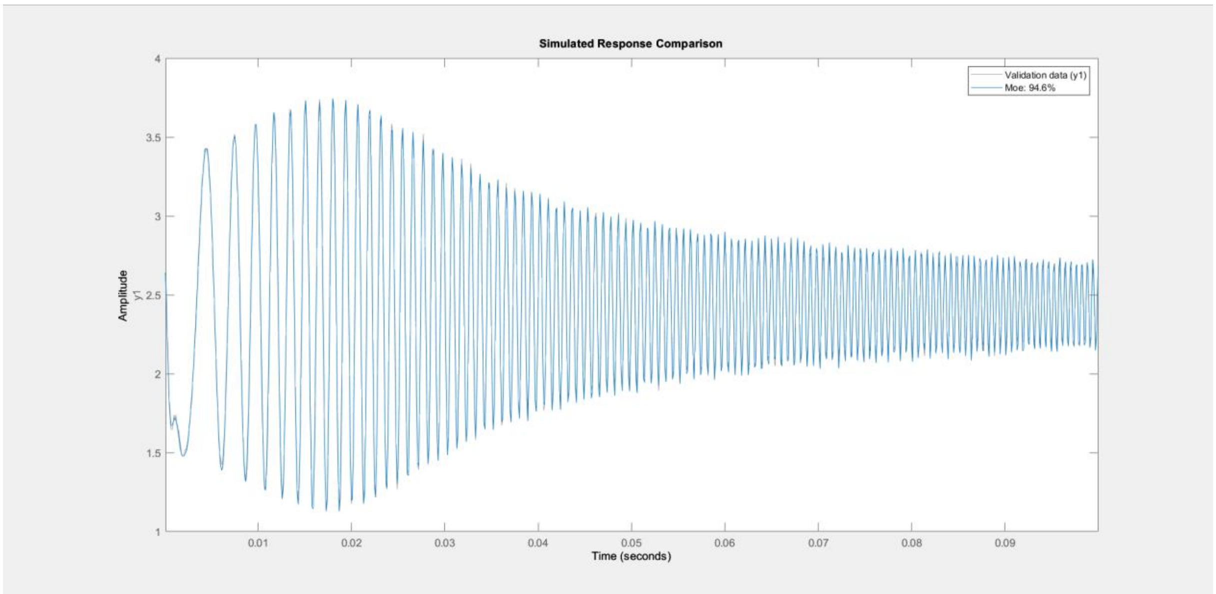
nf = 2 (numări poli),

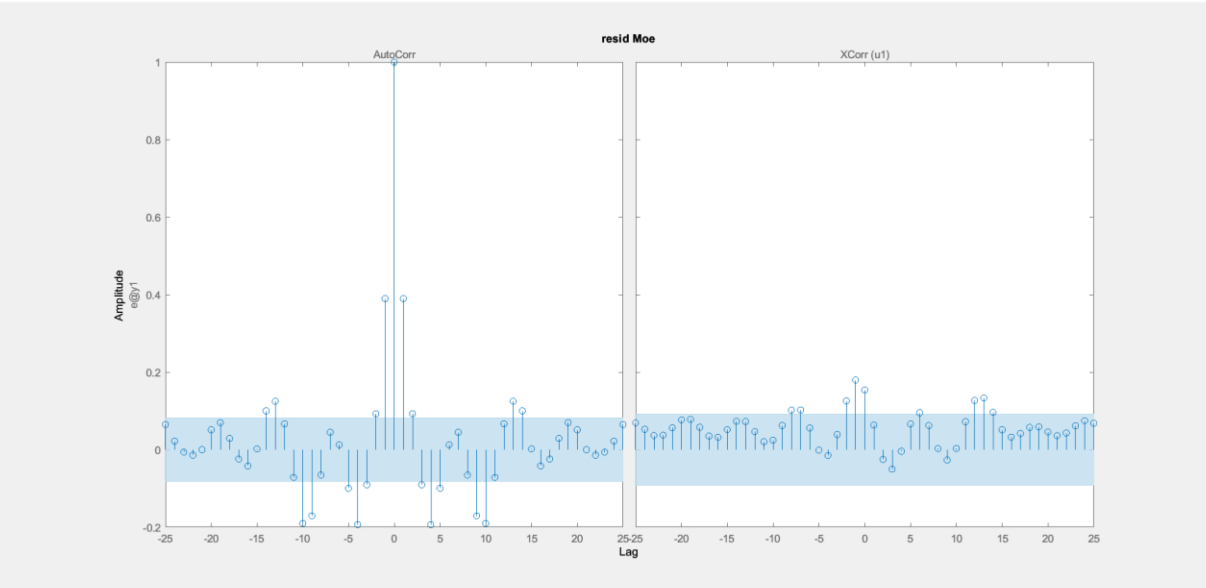
nk = 1(numărul tacților de întârziere)

Hz = $\frac{0.2334\, z^{-1} + 0.01292\, z^{-2}}{1 - 1.445\, z^{-1} + 0.6638\, z^{-2}}$ și funcția de transfer => Hs = $\frac{1475s + 2.75e07}{s^2 + 4097s + 2.729e07}$

Trecerea din discret în continuu se face cu ajutorul funcției „*d2c*“ (d2c(Hz_oe,'zoh') unde zoh înseamnă zero-order hold).

Următoarele grafice unu dintre ele v-a conține compararea sistemelor și în celălalt grafic v-a fi testul de intercorelație.





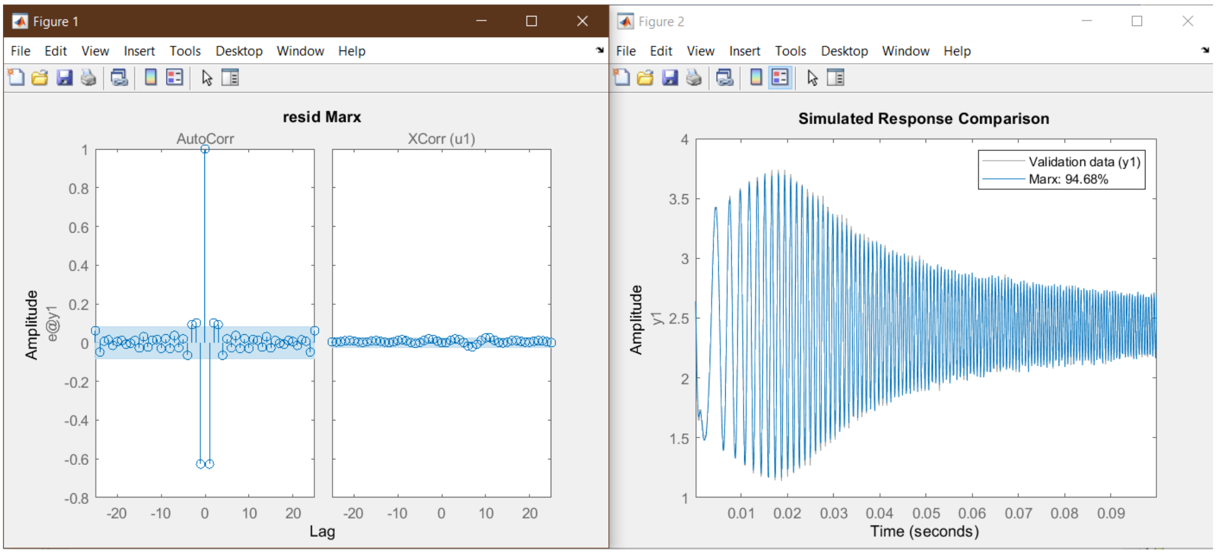
După cum se poate observa comparația are eroare de 5.4%, iar testul de intercorelație aproape îl trece cu succes.

```
%% Metoda oe
Moe=oe(date, [2 2 0])

figure,resid(date,Moe),shg,title('resid Moe')
figure,compare(date,Moe),shg

Hz_oe1 = tf(Moe.B,Moe.F,Te,'variable','z^-1')%functia de transfer in discret
Hs_oe1 = d2c(Hz_oe1,'zoh')%functia de transfer in continu
```

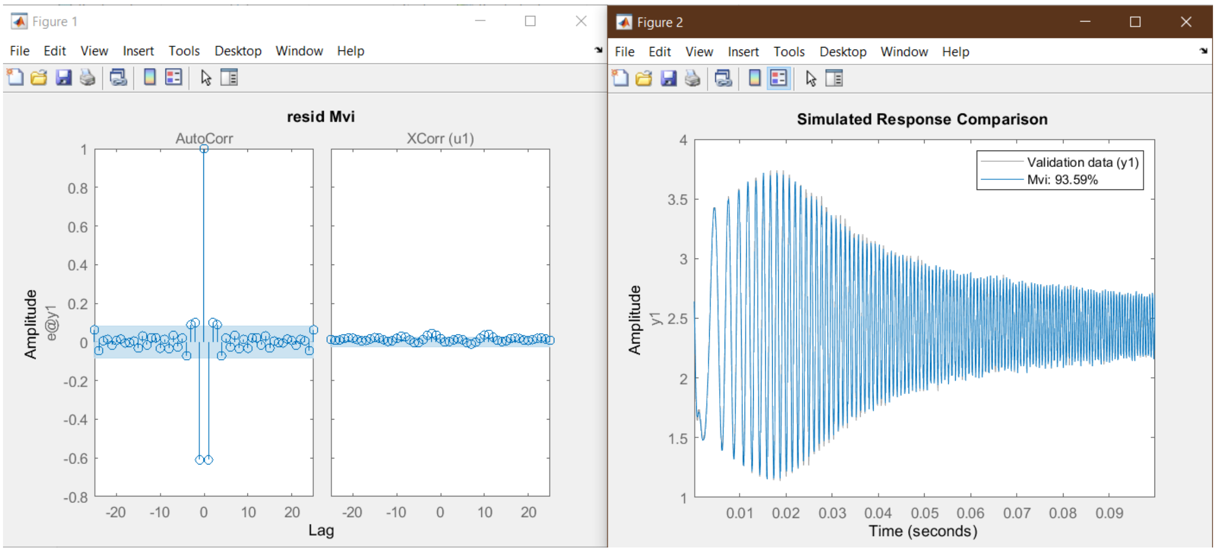
```
close all
clc
t = Moldovan(:,1);
u = Moldovan(:,2);
y = Moldovan(:,3);
x = Moldovan(:,4);

%Plot intrare/iesire y
figure
plot(t, u, t, y)
xlabel('Time')
ylabel('Amplitude')
legend('Intrare', 'Iesire fara zero' )
title('Input and Output Plot')

%Plot intrare/iesire x
figure
plot(t, u, t , x)
xlabel('Time')
ylabel('Amplitude')
legend('Intrare', 'Iesire cu un zero')
title('Input and Output Plot')

%%

ymin=180; %minim iesire
ymax=187; %maxim iesire
umin=178; %minim intrare
umax=185; %maxim intrare

Mr=(y(ymin)-y(ymax))/(u(umin)-u(umax)) %modulul de rezonanta
Tr=(t(ymax)-t(ymin))*2 %perioada de rezonanta
zeta=(sqrt((Mr-sqrt(Mr^2-1)))/2*Mr)%factorul de amortizare
wr = (2*pi/Tr) %pulsatia de rezonanta
wn = wr/sqrt(1-2*zeta^2) %pulsatia naturala
K=mean(y)/mean(u); %=1.0081


num=K*wn^2;
den=[K 2*zeta*wn wn^2];
Hdes=tf(num,den)

figure
bode(Hdes)


%Conditii initiale nenule=>este necesar modelul de tip spatiul starilor

A=[0 1;-wn^2 -2*zeta*wn];
B=[0;K*wn^2];
C=[1 0];
D=0;
%convertire in spatiul starilor
sys=ss(A,B,C,D);
%generare semnal simulat
ysim=lsim(sys, u , t,[y(1),0]);
```

```
figure
plot (t,ysim,'b',t,y,'r');
xlabel('Time')
ylabel('Amplitude')
legend('Simulated Output', 'Actual Output')
title('System Response')
hold on


J=1/sqrt(length(t))*norm(y-ysim)
empn=norm(y-ysim)/norm(y-mean(y))*100%eroare medie patratica normalizata
%%

Te=t(2)-t(1);%perioada esantionare
date=iddata(y, u, Te);
%verificare ordin
n4sid(date,1:10)

%% Metoda armax

Marmax=armax(date, [2 2 2 1])

figure,resid(date, Marmax),shg,title('resid Marmax')
figure,compare(date,Marmax),shg

Hz_arx1 = tf(Marmax.B,Marmax.A,Te,'variable','z^-1') %functia de transfer in
discret
Hs_arx1 = d2c(Hz_arx1,'zoh') %functia de transfer in continu

%% Metoda oe

Moe=oe(date, [2 2 1])

figure,resid(date,Moe),shg,title('resid Moe')
figure,compare(date,Moe),shg

Hz_oe1 = tf(Moe.B,Moe.F,Te,'variable','z^-1')%functia de transfer in discret
Hs_oe1 = d2c(Hz_oe1,'zoh')%functia de transfer in continu


%% Metoda arx

Marx=arx(date, [2,2,0]);%na,nb,timp mort
%2.2.0: 2-ord a,2-ord b,0-ord timp mort


figure,resid(date,Marx),shg,title('resid Marx')
figure,compare(date,Marx),shg


%% Metoda vi
Mvi=iv4(date, [2 2 1]);

figure,resid(date,Mvi),shg,title('resid Mvi')
figure,compare(date,Marx,Mvi),shg
```

Simularea cu ARX



Simularea cu IV4



*Identificarea sistemului de ordin 2 cu un zerou

Explicațiile sunt la fel ca la cel care nu are zerouri doar că acum avem alți parametri la armax (na = 2, nb = 3, nc=1, nk = 0) și oe (nb = 2, nf = 2, nk = 1)