Proiect la Identificarea Sistemelor

Identificarea si Simularea unui Circuit de ordin 2, FTJ cu amplificator operațional

Student: Barbul Laurentiu-Gavril-Cezar

Grupa: 30132

Profesor: Petru Dobra

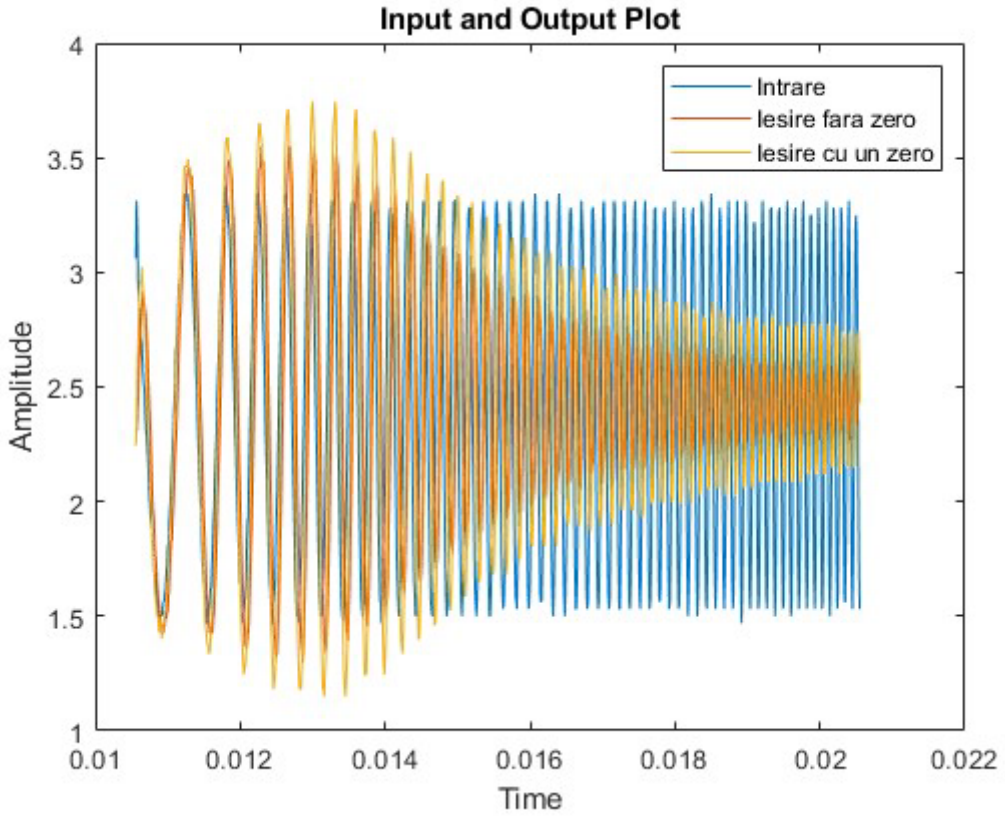
Data predări proiectului: 11.01.2024

Cuprins

Obținerea datelor experimentale

Aparatura utilizata : sursa de alimentare, multimetru, osciloscop si achiziție a datelor.

Pe baza a unui semnal de intrare, care trece printr-un filtru, se generează doua semnale de ieșire, unul este fără zerouri, iar celălalt este cu zerouri:



Vizualizarea datelor experimentale

Pentru vizualizarea datelor experimentale se utilizează mediul **Matlab**.

In **Matlab** vom importa fișierul cu care se lucrează, Semnal.mat. Pentru a afișa datele am rulat următorul cod in **Matlab**, unde in structura datelor avem doua câmpuri ‚X' si ‚Y'. Vectorul 'X' corespunde domeniului timp, iar in vectorul 'Y' se regăsesc valorile intrării notate cu "u", a primului semnal cu "y1" si a celui deal doilea semnal notata cu "y2".

Am afișat datele inițiale:

*Afișare semnal de intrare u, si semnalul de ieșire y1:	*Afișare semnal de intrare u, si semnalul de ieșire y2:

`%Plot intrare/iesire y1`
`figure`
`plot(t, u, t, y1)`
`xlabel('Time')`
`ylabel('Amplitude')`
`legend('Intrare', 'Iesire fara zero' )`
`title('Input and Output Plot')`
	`%Plot intrare/iesire y2`
	`figure`
	`plot(t, u, t , y2)`
	`xlabel('Time')`
	`ylabel('Amplitude')`
	`legend('Intrare', 'Iesire cu un zero')`
	`title('Input and Output Plot')`

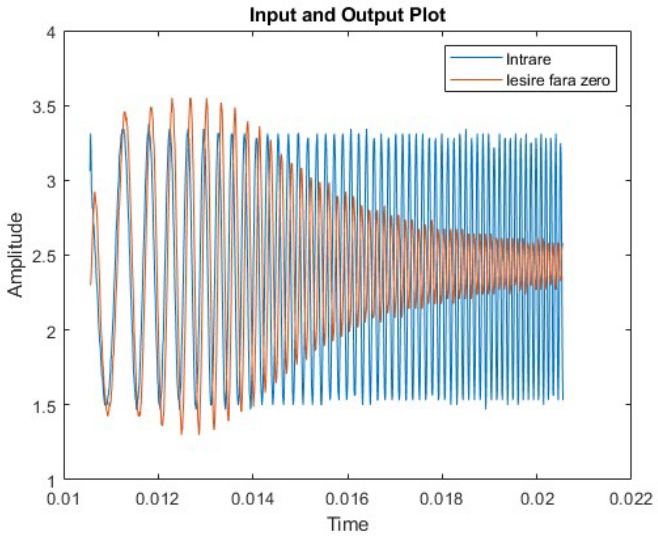


Fig. 1: Semnal inițial, intrare u si ieșire y1

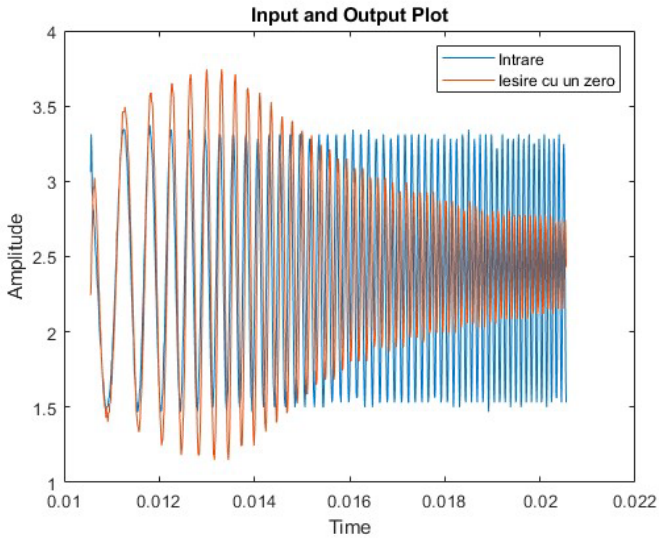


Fig. 2: Semnal inițial, intrare u si ieșire y2

Identificarea sistemelor

Prin metode neparametrice:

Semnalul de ieșire y1

In continuare am făcut identificarea sistemului pe ieșirea y1 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

```
% Identificarea sistemului prin metoda de rezonanta
umax = 208; %punct de pe intrare cu amplitudine maxima pozitiva
umin = 190; %punct de pe intrare cu amplitudine maxima negativa
ymax = 213; %punct de pe ieșirea y1 cu amplitudine maxima pozitiva
ymin = 195; %punct de pe ieșirea y1 cu amplitudine maxima negativa
```

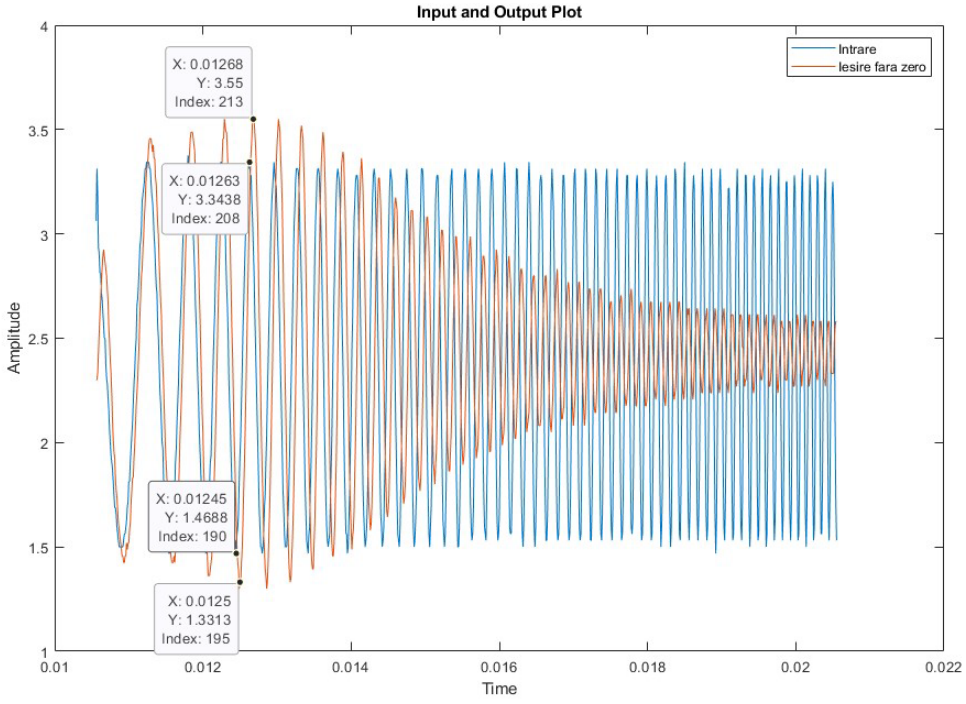


Fig. 3: Alegerea punctelor pe perioada de rezonanta

Pe baza punctelor selectate am determinat:

*Intervalul de timp:	$T_r = \left(t(y_{max}) - t(y_{min})\right) \cdot 2$ $\boldsymbol{T_r = 3.6 \cdot 10^{-4}}$
*Pulsația de rezonanta:	$\omega_r = 2 \cdot \frac{\pi}{T_r}$ $\boldsymbol{\omega_r = 1.7453 \cdot 10^{4}}$
*Timp de eșantionare:	$dt = t(y_{\max}) - t(u_{\max})$ $\boldsymbol{dt = 5 \cdot 10^{-5}}$
*Modul rezonanta:	$M_r = \frac{y_1(y_{\min}) - y_1(y_{\max})}{u(u_{\min}) - u(u_{\max})}$ $\boldsymbol{M_r = 1.1833}$
*Factor de amortizare:	$\zeta = \sqrt{\frac{M_r - \sqrt{M_r^2 - 1}}{2 \cdot M_r}}$ $\boldsymbol{\zeta = 0.4824}$
*Pulsația naturala:	$\omega_n = \frac{\omega_r}{\sqrt{1 - 2 \cdot \zeta^2}}$ $\boldsymbol{\omega_n = 2.3869 \cdot 10^{4}}$
*Factor de proporționalitate:	$K = \frac{\bar{y}}{\bar{u}}$ $\boldsymbol{K = 1.0079}$

Cu ajutorul acestui cod din **Matlab**:

```
%% Compunerea Parametrilor

%Interval de timp
Tr = (t(ymax) - t(ymin)) * 2;
%Pulsatie de rezonanta
wr = 2 * pi / Tr;
%Timp de esantionare
dt = t(ymax) - t(umax);
%Modul rezonanta raportul
Mr = (y1(ymin) - y1(ymax)) / (u(umin) - u(umax));
%Factor de amortizare
tita = sqrt((Mr - sqrt(Mr^2 - 1)) / (2 * Mr));
%Pulsatia naturala
wn = wr / sqrt(1 - 2 * tita^2);
%Factor de proportionalitate
K = mean(y1)/mean(u);
```

Pentru a determina funcția de transfer de ordin 2 am folosit variabilele determinate:

$$\boldsymbol{H(s) = \frac{k \cdot \omega_n}{s^2 + 2\zeta\omega_n S + \omega_n^2}} = 1.0079 \cdot \frac{23869 \cdot 10^4}{s^2 + 2 \cdot 0.4824 \cdot 2.3869 \cdot 10^4 \cdot s + (2.3869 \cdot 10^4)^2}$$

$$\boldsymbol{H(s) = \frac{5.742 \cdot 10^8}{s^2 + 2.303 \cdot 10^4 s + 5.697 \cdot 10^8}}$$

Cod Matlab:

```
Hs = tf(K * wn^2, [1, 2 * tita * wn, wn^2]);
```

Pentru simularea sistemului ne vom folosi de spațiul stărilor:

Cu parametri:	Cu valori:	Cod Matlab:
$A = \begin{pmatrix} 0 & 1 \\ -\omega_n^2 & -2\xi\omega_n \end{pmatrix}$	$A = \begin{pmatrix} 0 & 1 \\ -5.6975 \cdot 10^8 & -2.3027 \cdot 10^4 \end{pmatrix}$	`A = [0 1;-wn^2 -2*tita*wn];`
$B = \begin{pmatrix} 0 \\ K \cdot \omega_n^2 \end{pmatrix}$	$B = \begin{pmatrix} 0 \\ 5.7423 \cdot 10^8 \end{pmatrix}$	`B = [0;K*wn^2];`
$C = (1 \quad 0)$	$C = (1 \quad 0)$	`C = [1 0];`
$D = (0)$	$D = (0)$	`D = 0;`

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:

```
% Convertire in spatiul starilor
sys = ss(A, B, C, D);
% Generare semnal simulat
ysim1 = lsim(sys, u, t,[y1(1),(y1(2)-y1(1))/(t(2)-t(1))]);
```

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:

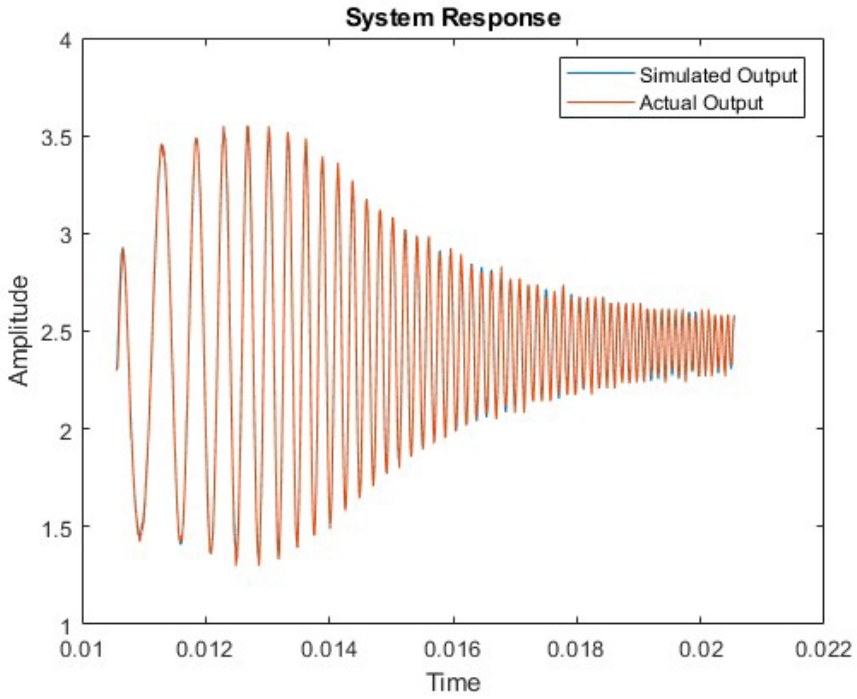


Fig. 4: Validare semnal simulat

Cod Matlab:

```
% Plot cu semnalul de iesire y1 si cu semnalul simulat
figure
plot(t, ysim1, t, y1)
xlabel('Time')
ylabel('Amplitude')
legend('Simulated Output', 'Actual Output')
title('System Response')
hold on
```

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

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

```
% Calcularea eroare patratica normalizata
empn = norm(y1 - ysim1) / norm(y1 - mean(y1)) * 100;
```

$$\varepsilon_{MPN} = 4.6194\ \%$$

Prin metode parametrice:

Semnalul de ieșire y1

In continuare am făcut identificarea sistemului pe ieșirea y1 fără zero prin 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.**

```
%Grupez intr-un singur obiect
Ta =t(2)-t(1);% timp de achizitie
d_id1 = iddata(y1,u,Ta)
%Verificare ordin
n4sid(d_id1,1:10)
```

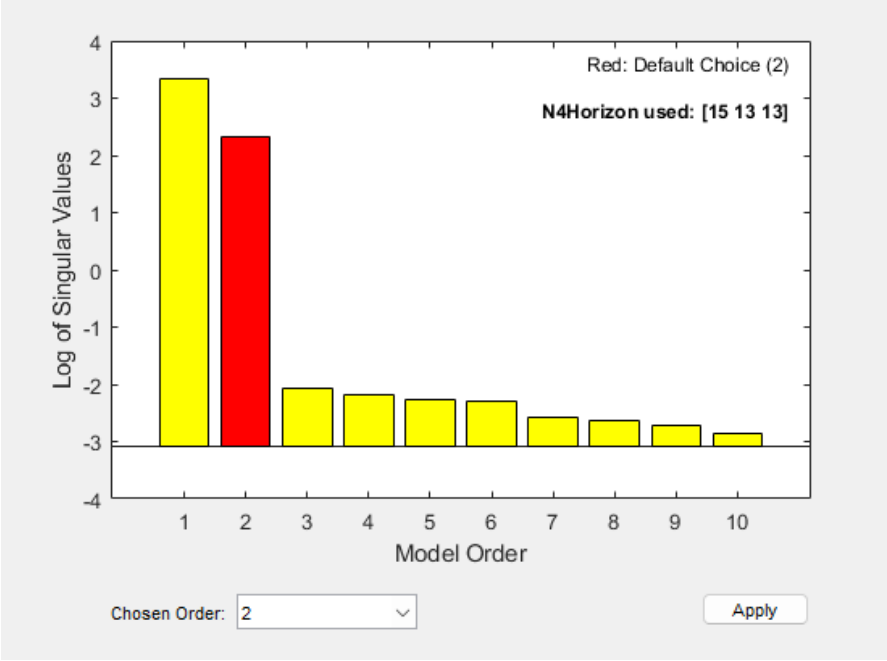


Fig. 5: Determinare ordin sistem prin **N4SID**

MCMMPE – **ARMAX**(Metoda celor mai mici pătrate extinsa)

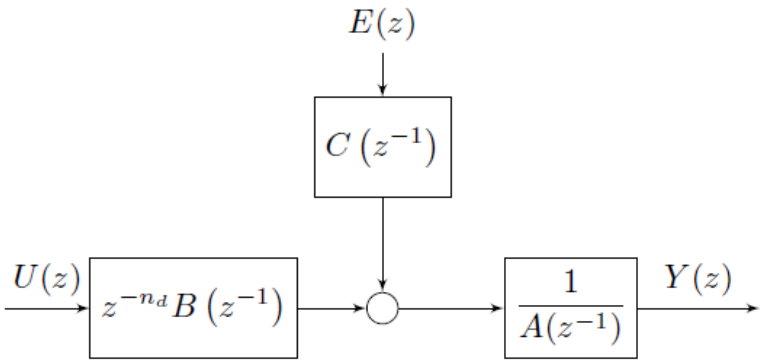


Fig. 6: Structura corespunzătoare metodei **ARMAX**

Modelul discret de tip proces + perturbație corespunzător metodei **ARMAX** este:

$$A(q^{-1})^{y}(q) = q^{-n_d}B(q^{-1})U(q) + C(q^{-1})E(q)$$

Parametrii de structură al sistemului sunt:

nA = ordinul polinomului A (numărul de poli),

nB = ordinul polinomului B (numărul de zerouri),

nC = ordinul polinomului C,

nd = numărul tacților de întârziere.

Pentru a face genera polinoamele A,B,C vom folosi funcția **armax** din Matlab, iar pentru a vedea daca sistemul trece testul de autocorelatie, vom folosi comanda **resid**, iar pentru a valida semnalul simulat vom folosi comanda **compare**.

Cod matlab:

```
%% 1:Metoda validata prin autocorelatie
M1_armax = armax(d_id1,[2,1,2,1])
resid(d_id1,M1_armax), shg %pentru validare
autocorelatie/intercorelatie
figure
compare(d_id1,M1_armax), shg %verificare suprapunere semnal simulat
```

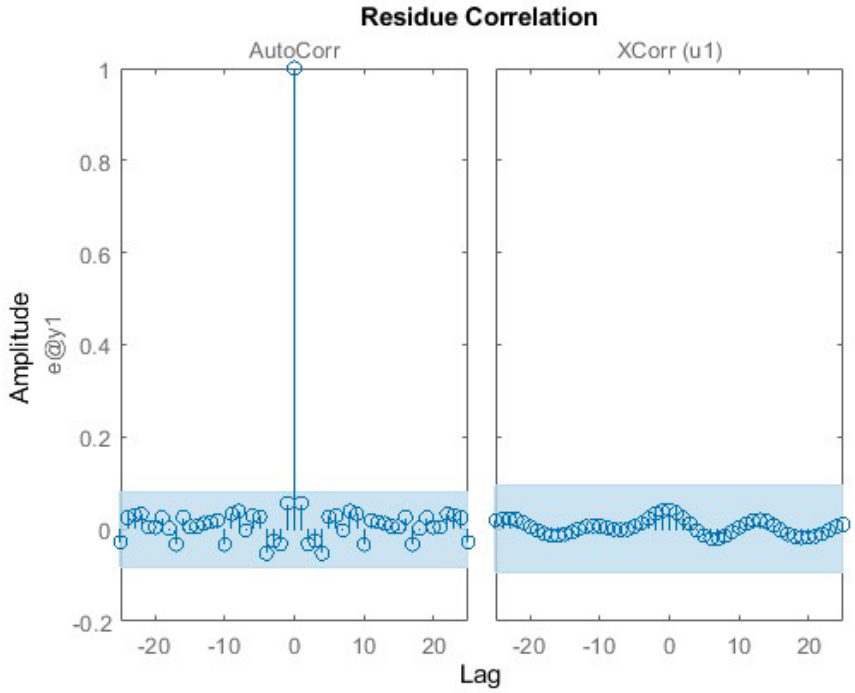


Fig. 6: Testare validare autocorelație **ARMAX**

Observam ca in figura de mai sus modelul este validat prin trecerea testului de autocorelație deoarece se încadrează in banda de încredere reprezentata cu albastru deschis. Iar in figura de mai jos putem observa ca gradul de urmărire este de **95.88%.**

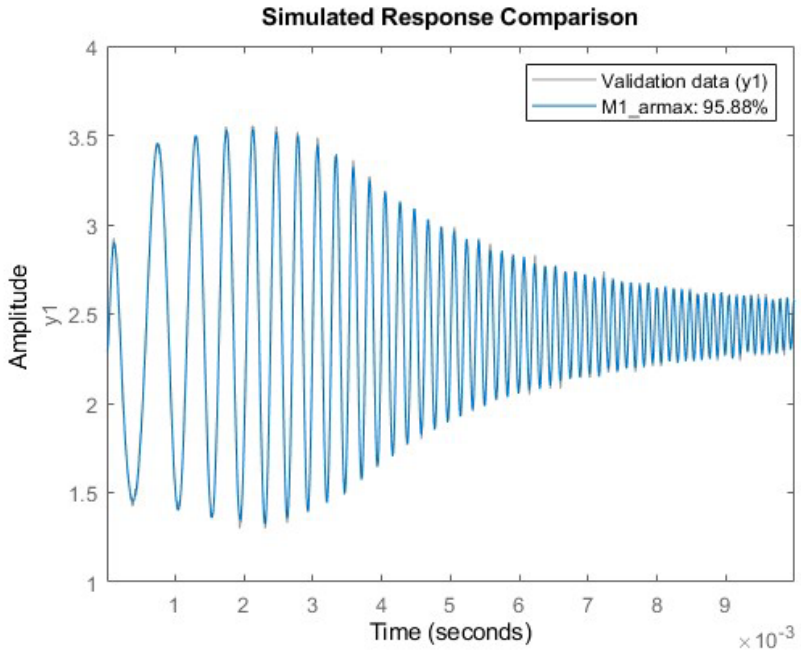


Fig. 7: Validare metoda **ARMAX**

Pentru a determina funcția de transfer in discret am folosit comanda **tf** din Matlab:

$$H_{zarx1} = \frac{0.05037z^{-1}}{1 - 1.749z^{-1} + 0.7985z^{-2}}$$

Polinoamele sistemului:

$A(z) = 1 - 1.7486z^{-1} + 0.7985z^{-2}$
$B(z) = 0.05037z^{-1}$
$C(z) = 1 - 1.6z^{-1} + 0.6762z^{-2}$

Alți parametri:
NoiseVariance: 3.8887*10^-4
Ts: 1*10^-5

Pentru a determina funcția de transfer a sistemului din discret in continu am folosit comanda **d2c** ,prin metoda **zoh**(zero order hold):

$$H_{sarx1} = \frac{2732s + 5.652 \cdot 10^8}{s^2 + 2.25 \cdot 10^4 + 5.606 \cdot 10^8}$$

Cod Matlab:

```
Hz_arx1 = tf(M1_armax.B,M1_armax.A,Ta,'variable','z^-1') %functia de
transfer in discret
Hs_arx1 = d2c(Hz_arx1,'zoh') %functia de transfer in continu
```

OE - Metoda erorii de ieșire

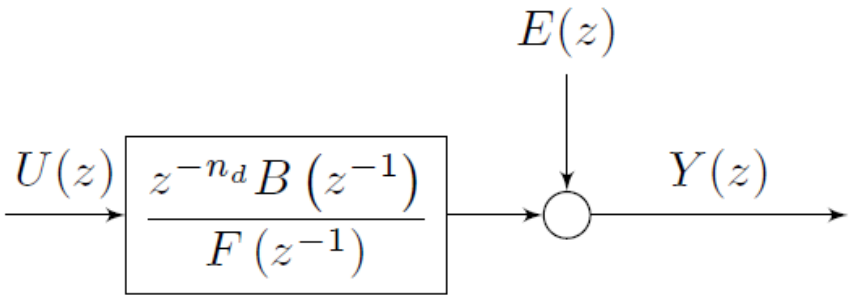


Fig. 6: Structura corespunzătoare metodei **OE**

Modelul discret de tip proces + perturbație corespunzător metodei **OE** este:

$$Y(q) = \frac{q^{-n_d}B(q^{-1})}{F(q^{-1})}U(q) + E(q)$$

Parametrii de structură al sistemului sunt:

nF = ordinul polinomului F (numărul de poli),

nB = ordinul polinomului B (numărul de zerouri),

nd = numărul tacților de întârziere.

Pentru a face genera polinoamele B,F vom folosi funcția **OE** din Matlab, iar pentru a vedea daca sistemul trece testul de intercorelație, vom folosi comanda **resid**, iar pentru a valida semnalul simulat vom folosi comanda **compare**.

Cod matlab:

```
%% 2:Metoda validata prin intercorelatie
M1_oe = oe(d_id1,[1,2,1])
resid(d_id1,M1_oe), shg %validare autocorelatie/intercorelatie
figure
compare(d_id1,M1_oe), shg %verificare suprapunere semnal simulat
```

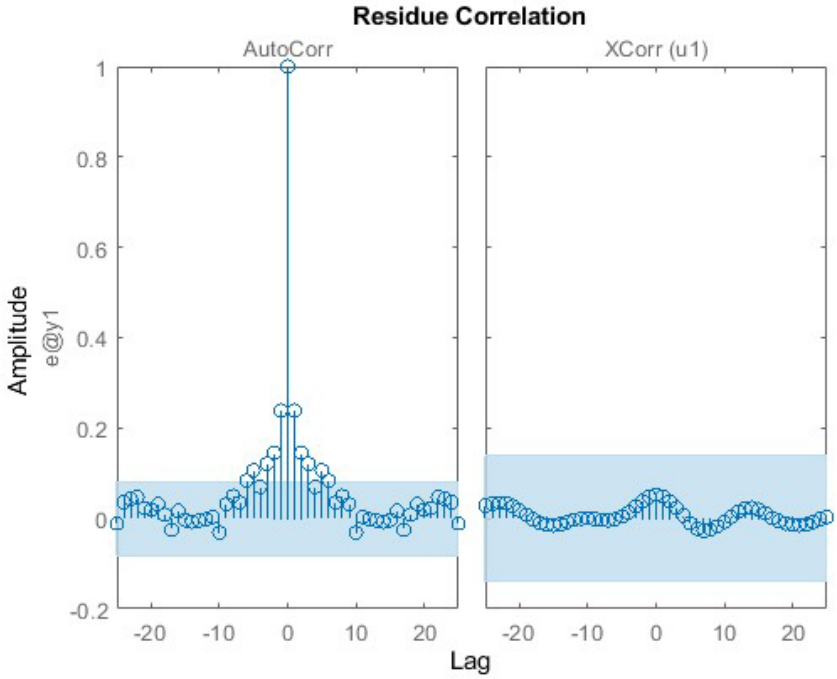


Fig. 8: Testare validare Intercorelație **OE**

Observam ca in figura de mai sus modelul este validat prin trecerea testului de intercorelație deoarece se încadrează in banda de încredere reprezentata cu albastru deschis. Iar in figura de mai jos putem observa ca gradul de urmărire este de **95.88%.**

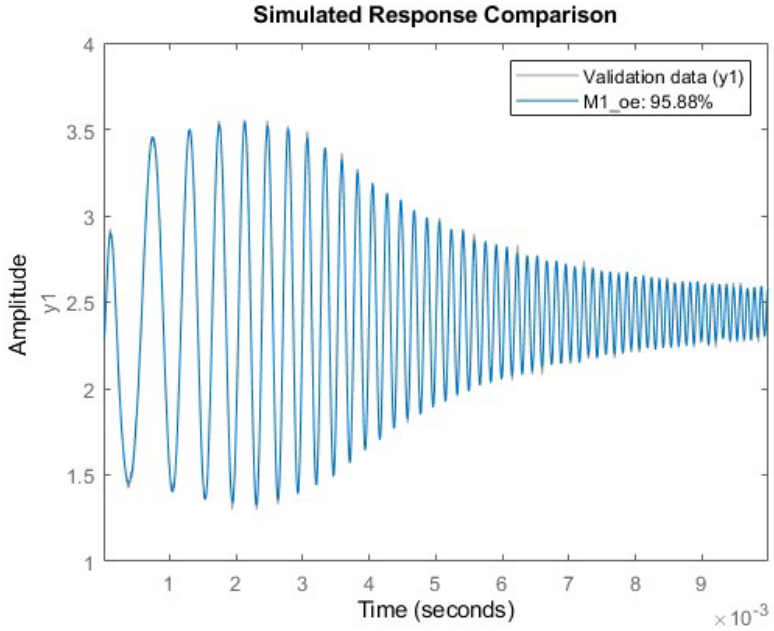


Fig. 9: Validare metoda **OE**

Pentru a determina funcția de transfer in discret am folosit comanda **tf** din Matlab:

$$H_{zoe1} = \frac{0.05046z^{-1}}{1 - 1.749z^{-1} + 0.7988z^{-2}}$$

Polinoamele sistemului:

$B(z) = 0.05046z^{-1}$
$F(z) = 1 - 1.7487z^{-1} + 0.7988z^{-2}$

Alți parametri:
NoiseVariance: 4.1670*10^-4
Ts: 1*10^-5

Pentru a determina funcția de transfer a sistemului din discret in continu am folosit comanda **d2c** ,prin metoda **zoh**(zero order hold):

$$H_{soe1} = \frac{2732s + 5.66 \cdot 10^8}{s^2 + 2.247 \cdot 10^4 + 5.614 \cdot 10^8}$$

Cod Matlab:

```
Hz_oe1 = tf(M1_oe.B,M1_oe.F,Ta,'variable','z^-1')%functia de
transfer in discret
Hs_oe1 = d2c(Hz_oe1,'zoh')%functia de transfer in continu
```

Semnalul de ieșire y2

In continuare am făcut identificarea sistemului pe ieșirea y2 cu un zero prin 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.**

```
%Grupez intr-un singur obiect
Ta =t(2)-t(1);% timp de achizitie
d_id2 = iddata(y2,u,Ta)
%Verificare ordin
n4sid(d_id2,1:10)
```

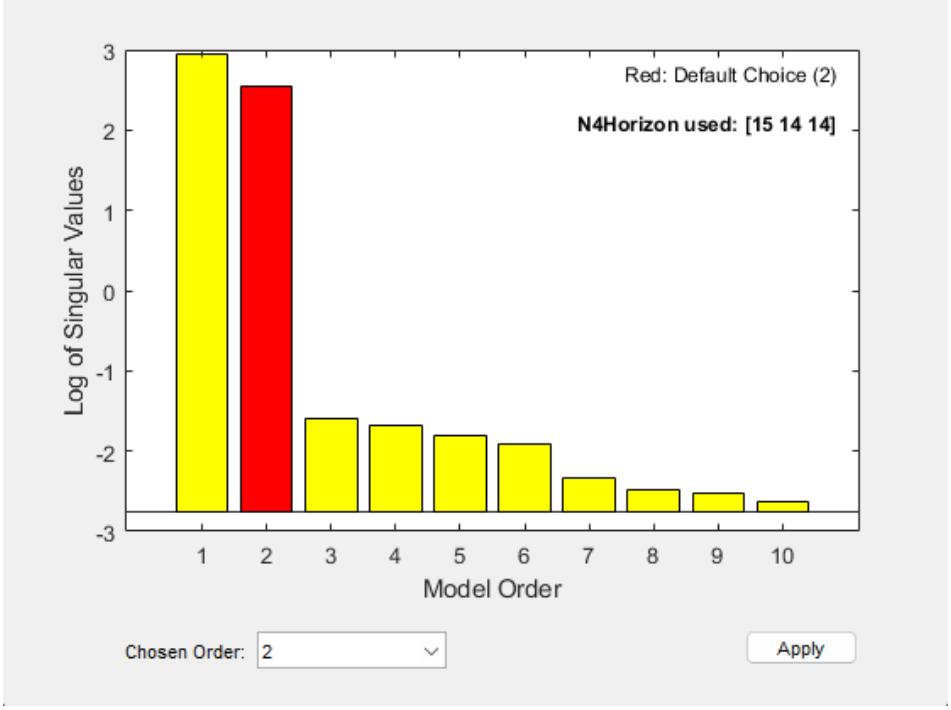


Fig. 5: Determinare ordin sistem prin **N4SID**

MCMMPE – **ARMAX**(Metoda celor mai mici pătrate extinsa)

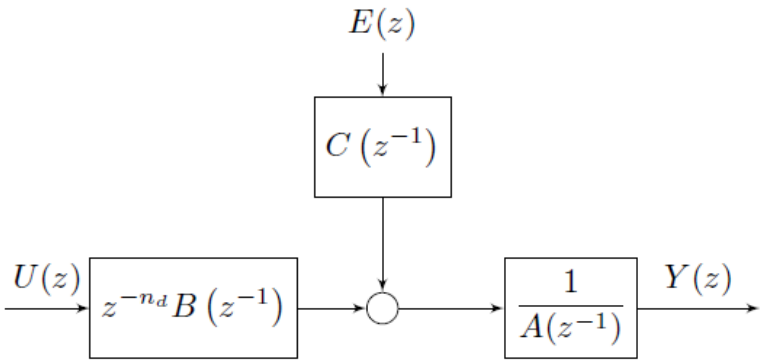


Fig. 6: Structura corespunzătoare metodei **ARMAX**

Modelul discret de tip proces + perturbație corespunzător metodei **ARMAX** este:

$$A(q^{-1})^{y}(q) = q^{-n_d}B(q^{-1})U(q) + C(q^{-1})E(q)$$

Parametrii de structură al sistemului sunt:

nA = ordinul polinomului A (numărul de poli),

nB = ordinul polinomului B (numărul de zerouri),

nC = ordinul polinomului C,

nd = numărul tacților de întârziere.

Pentru a face genera polinoamele A,B,C vom folosi funcția **armax** din Matlab, iar pentru a vedea daca sistemul trece testul de autocorelație, vom folosi comanda **resid**, iar pentru a valida semnalul simulat vom folosi comanda **compare**.

Cod matlab:

```
%% 1:Metoda validata prin autocorelatie
M2_armax = armax(d_id2,[2,2,2,1]) %intercorelatie intre iesire si
zgomot
resid(d_id2,M2_armax), shg %pentru validare
autocorelatie/intercorelatie
figure
compare(d_id2,M2_armax), shg
```

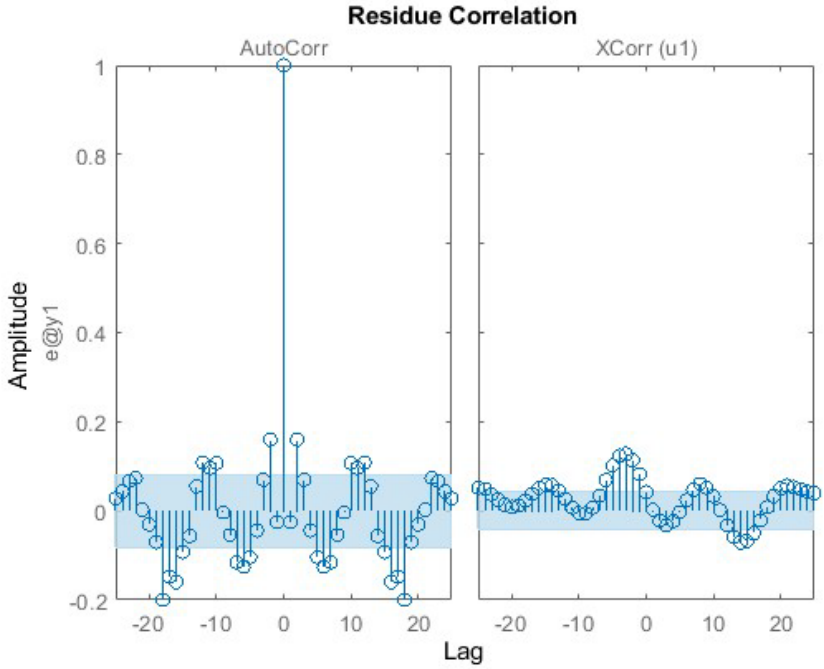


Fig. 6: Testare validare autocorelație **ARMAX**

Observam ca in figura de mai sus modelul este validat prin trecerea testului de autocorelație deoarece se încadrează in banda de încredere reprezentata cu albastru deschis. Iar in figura de mai jos putem observa ca gradul de urmărire este de **91.17%.**

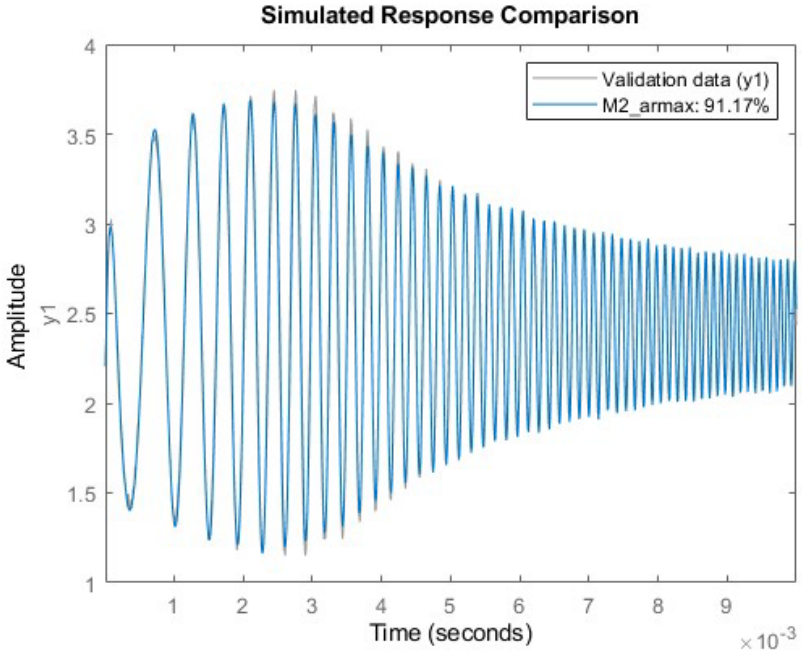


Fig. 7: Validare metoda **ARMAX**

Pentru a determina funcția de transfer in discret am folosit comanda **tf** din Matlab:

$$H_{zarx2} = \frac{0.2189z^{-1} - 0.1759z^{-2}}{1 - 1.744z^{-1} + 0.7867z^{-2}}$$

Polinoamele sistemului:

$A(z) = 1 - 1.7441z^{-1} + 0.7867z^{-2}$
$B(z) = 0.2189z^{-1} - 0.1759z^{-2}$
$C(z) = 1 - 0.9935z^{-1} + 0.07704z^{-2}$

Alți parametri:
NoiseVariance: 9.4234*10^-4
Ts: 1*10^-5

Pentru a determina funcția de transfer a sistemului din discret in continu am folosit comanda **d2c** ,prin metoda **zoh**(zero order hold):

$$H_{sarx2} = \frac{2.228 \cdot 10^4 s + 4.855 \cdot 10^8}{s^2 + 2.399 \cdot 10^4 + 4.809 \cdot 10^8}$$

Cod Matlab:

```
Hz_arx2 = tf(M2_armax.B,M2_armax.A,Ta,'variable','z^-1')%functia de
transfer in discret
Hs_arx2 = d2c(Hz_arx2,'zoh')%functia de transfer in continu
```

IV - Metoda variabilelor instrumentale

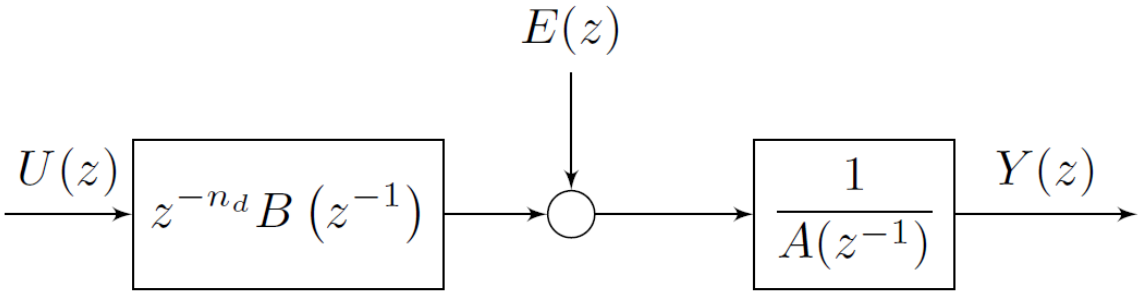


Fig. 6: Structura corespunzătoare metodei **IV**

Modelul discret de tip proces + perturbație corespunzător metodei **IV** este:

$$A(q^{-1})^{y}Y(q) = q^{-n_d}B(q^{-1})U(q) + E(q)$$

Parametrii de structură al sistemului sunt:

nA = ordinul polinomului A (numărul de poli),

nB = ordinul polinomului B (numărul de zerouri),

nd = numărul tacților de întârziere.

Pentru a face genera polinoamele A,B vom folosi funcția **IV** din Matlab, iar pentru a vedea daca sistemul trece testul de intercorelație, vom folosi comanda **resid**, iar pentru a valida semnalul simulat vom folosi comanda **compare**.

Cod matlab:

```
%% 2:Metoda validata prin intercorelatie
M2_iv4 = oe(d_id1,[2,2,1])
resid(d_id2,M2_iv4,10), shg %validare autocorelatie/intercorelatie
figure
compare(d_id2,M2_iv4), shg %verificare suprapunere semnal simulat
```

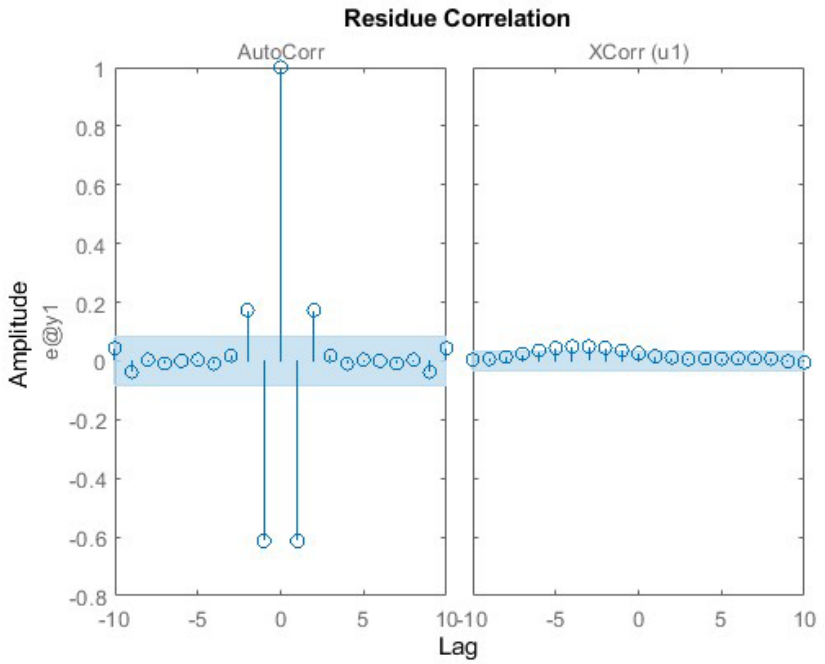


Fig. 8: Testare validare Intercorelație **IV4**

Observam ca in figura de mai sus modelul este validat prin trecerea testului de intercorelație deoarece se încadrează in banda de încredere reprezentata cu albastru deschis. Iar in figura de mai jos putem observa ca gradul de urmărire este de **82.75%.**

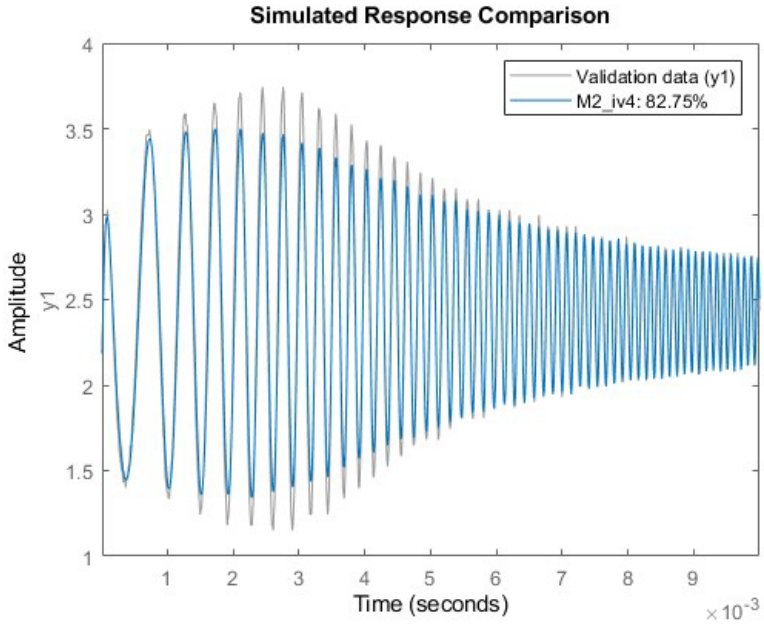


Fig. 9: Validare metoda **IV4**

Pentru a determina funcția de transfer in discret am folosit comanda **tf** din Matlab:

$$H_{ziv2} = \frac{0.196z^{-1} - 0.1485z^{-2}}{1 - 1.704z^{-1} + 0.7516z^{-2}}$$

Polinoamele sistemului:

$A(z) = 1 - 1.7043z^{-1} + 0.7516z^{-2}$
$B(z) = 0.196z^{-1} - 0.1485z^{-2}$

Alți parametri:
NoiseVariance: 0.0019
Ts: 1*10^-5

Pentru a determina funcția de transfer a sistemului din discret in continu am folosit comanda **d2c** ,prin metoda **zoh**(zero order hold):

$$H_{siv2} = \frac{1.985 \cdot 10^4 s + 5.66 \cdot 10^8}{s^2 + 2.855 \cdot 10^4 + 5.464 \cdot 10^8}$$

Cod Matlab:

```
Hz_iv2 = tf(M2_iv4.B,M2_iv4.A,Ta,'variable','z^-1')%functia de
transfer in discret
Hs_iv2 = d2c(Hz_iv2,'zoh')%functia de transfer in continu
```

COD MATLAB:

```
%% Incarcare Date

clc
clear
close all
load('Semnal.mat')
t = Barbul(:, 1); %timpul
u = Barbul(:, 2); %semnalul de intrare
y1 = Barbul(:, 3); %primul semnal de iesire
y2 = Barbul(:, 4); %al doilea semnal de iesire

%% Plot intrare/iesire

%Plot intrare/iesire y1
figure
plot(t, u, t, y1)
xlabel('Time')
ylabel('Amplitude')
legend('Intrare', 'Iesire fara zero' )
title('Input and Output Plot')

dcm = datacursormode(gcf);
set(dcm, 'UpdateFcn', @customCursorUpdateFcn);

%Plot intrare/iesire y2
figure
plot(t, u,t , y2)
xlabel('Time')
ylabel('Amplitude')
legend('Intrare', 'Iesire cu un zero')
title('Input and Output Plot')

dcm = datacursormode(gcf);
set(dcm, 'UpdateFcn', @customCursorUpdateFcn);

%% Identificarea sistemului pe iesirea y1 prin metoda de rezonanta

umax = 208; %punct de pe intrare cu amplitudine maxima pozitiva
umin = 190; %punct de pe intrare cu amplitudine maxima negativa
ymax = 213; %punct de pe iesirea y1 cu amplitudine maxima pozitiva
ymin = 195; %punct de pe iesirea y1 cu amplitudine maxima negativa

%% Compunerea Parametrilor

%Interval de timp
Tr = (t(ymax) - t(ymin)) * 2;

%Pulsatie de rezonanta
wr = 2 * pi / Tr;

%Timp de esantionare
dt = t(ymax) - t(umax);

%Modul rezonanta raportul
Mr = (y1(ymin) - y1(ymax)) / (u(umin) - u(umax));

%Factor de amortizare
tita = sqrt((Mr - sqrt(Mr^2 - 1)) / (2 * Mr));

%Pulsatia naturala
wn = wr / sqrt(1 - 2 * tita^2);

%Factor de proportionalitate
```

```
K = mean(y1)/mean(u);

%% Generare semnal simulat si afisare

% Crearea functiei de transfer
Hs = tf(K * wn^2, [1, 2 * tita * wn, wn^2]);

% Generare matrici pentru spatiul starilor
A = [0 1;-wn^2 -2*tita*wn];
B = [0;K*wn^2];
C = [1 0];
D = 0;

% Convertire in spatiul starilor
sys = ss(A, B, C, D);

% Generare semnal simulat
ysim1 = lsim(sys, u, t, [y1(1), (y1(2) - y1(1)) / (t(2) - t(1))]);

% Plot cu semnalul de iesire y1 si cu semnalul simulat
figure
plot(t, ysim1, t, y1)
xlabel('Time')
ylabel('Amplitude')
legend('Simulated Output', 'Actual Output')
title('System Response')
hold on

% Calcularea eroare patratica normalizata
empn = norm(y1 - ysim1) / norm(y1 - mean(y1)) * 100;

% Afisare in command valoare eroare patratica normalizata
disp(['Eroare patratica normalizata (empn): ' num2str(empn) '%']);

%% Fara Zero

%grupez intr-un singur obiect
Ta =t(2)-t(1);% timp de achizitie
d_id1 = iddata(y1,u,Ta)
%verificare ordin
n4sid(d_id1,1:10)
```

```
%% 1:Metoda validata prin autocorelatie

M1_armax = armax(d_id1,[2,1,2,1]) %intercorelatie intre iesire si zgomot
resid(d_id1,M1_armax), shg %validare autocorelatie/intercorelatie
figure
compare(d_id1,M1_armax), shg %verificare suprapunere semnal simulat

Hz_arx1 = tf(M1_armax.B,M1_armax.A,Ta,'variable','z^-1')%functia de transfer in discret
Hs_arx1 = d2c(Hz_arx1,'zoh')%functia de transfer in continu

%% 2:Metoda validata prin intercorelatie

M1_oe = oe(d_id1,[1,2,1])
resid(d_id1,M1_oe), shg %validare autocorelatie/intercorelatie
figure
compare(d_id1,M1_oe), shg %verificare suprapunere semnal simulat

Hz_oe1 = tf(M1_oe.B,M1_oe.F,Ta,'variable','z^-1')%functia de transfer in discret
Hs_oe1 = d2c(Hz_oe1,'zoh')%functia de transfer in continu

%% Cu un zero

%Grupez intr-un singur obiect
Ta =t(2)-t(1);% timp de achizitie
d_id2 = iddata(y2,u,Ta)
%verificare ordin
n4sid(d_id2,1:10)
%% 1:Metoda validata prin autocorelatie

M2_armax = armax(d_id2,[2,2,2,1]) %intercorelatie intre iesire si zgomot
resid(d_id2,M2_armax), shg  %validare autocorelatie/intercorelatie
figure
compare(d_id2,M2_armax), shg %verificare suprapunere semnal simulat

Hz_arx2 = tf(M2_armax.B,M2_armax.A,Ta,'variable','z^-1')%functia de transfer in discret
Hs_arx2 = d2c(Hz_arx2,'zoh')%functia de transfer in continu

%% 2:Metoda validata prin intercorelatie

M2_iv4 = iv4(d_id2,[2 2 1]); %intercorelatie intre iesire si zgomot
resid(d_id2,M2_iv4,10), shg %validare autocorelatie/intercorelatie
figure
compare(d_id2,M2_iv4), shg %verificare suprapunere semnal simulat

Hz_iv2 = tf(M2_iv4.B,M2_iv4.A,Ta,'variable','z^-1')%functia de transfer in discret
Hs_iv2 = d2c(Hz_iv2,'zoh')%functia de transfer in continu
```